PROCEEDINGS

The 11th HKBU-CSD Postgraduate Research Symposium

PG Day 2010

Department of Computer Science
Hong Kong Baptist University

March 15 & March 16, 2010

The 11th HKBU-CSD Postgraduate Research Symposium (PG Day) Program

March 15th 2010, Monday		
Time	**Sessions**	
09:00- 09:15	**On-site Registration (OEE 601-3)**	
09:15- 09:30	**Welcome: Prof. Jiming Liu, Head of Computer Science Department, HKBU**	
09:30- 12:00	**Session I (Chair: Mr. Chen Xiaowei, OEE 601-3)**	**Session II (Chair: Mr. Xia Shang, FSC 701B)**
	• ***A new BEMD method based on self-similar feature*** Mr. Pan Jian Jia • ***Face Recognition based on the New Phase Local Features*** Ms. Zhang Dan • ***Binary Discriminant Analysis for Face Template Protection*** Mr. Feng Yicheng • ***Very Low Resolution Face Recognition Problem*** Mr. Zou Weiwen • ***Display results of opinion mining with tagclouds*** Mr. Qi Luole	• ***Maintenance of Minimal Bisimulation of Cyclic Graphs*** Mr. Deng Jintian • ***L-BFGS and Delayed Dynamical Systems Approach for Unconstrained Optimization*** Ms. Xie Xiaohui • ***Speeding up K-Means Algorithm on GPU*** Mr. Li You • ***Design and Implementation of Multiple-precision Integer library for GPUs*** Mr. Zhao Kaiyong ***Hiding Emerging Pattern with Local Recoding Generalization*** Mr. Cheng Michael
12:00- 14:30	Noon Break	
14:30- 16:30	Distinguished Lecture (LT1) **(Chair: Prof. Jiming Liu, Head of Department of Computer Science, HKBU)**	
	Product Form Solutions for Stochastic Networks: Discovery or Invention? **Prof. Erol Gelenbe** Professor in the Dennis Gabor Chair, Imperial College London	
16:30- 19:00	**Session III (Chair: Mr. Li You, OEE 601-603)**	**Session IV (Chair: Mr. Cheng Michael, FSC 701B)**
	• ***Enhanced Location Estimation in Wireless LAN environment using Hybrid method*** Mr. Shum Chin Yiu • ***Concept Discovery in Youtube.com using Factorization Method*** Ms. Leung Kwan Wai • ***Measurements, Analysis and Modeling of Private Trackers*** Mr. Chen Xiaowei • ***Understanding the Cultural Influence on Tagging Pattern*** Mr. Tsoi Ho Keung • ***Vaccination Deployment Strategies in Protection against Influenza A (H1N1) Infection*** Mr. Xia Shang	• ***Autonomy-Oriented Mechanisms for Efficient Energy Distribution*** Mr. Shi Benyun • ***Selectivity Estimation of XPath for Cyclic Graphs*** Mr. Peng Yun • ***Patient Journey Optimization using a Multi-agent Approach*** Mr. Choi Chung Ho • ***Methods of Motion Segmentation in Compressed Domain*** Mr. Cheng Quan Jia • ***Secure proximity detection*** Mr. Li Hong Ping

The 11th HKBU-CSD Postgraduate Research Symposium (PG Day) Program

March 16th 2010, Tuesday	
Time	**Sessions**
09:00- 10:30	**Session V (Chair: Mr. Zou Weiwen, OEE 601-3)**
	• ***Survey of Content-base Music Information Retrieval*** Mr. Deng Jie • ***Efficient Waiting Time Management in Hospital from AOC Approach*** Mr. Tao Li • ***Characterizing Multiple Social Dynamics with Autonomy Oriented Computing*** Mr. Huang Lailei
10:30- 10:40	Tea Break
10:40- 12:10	**Session VI (Chair: Mr. Zou Weiwen, OEE 601-3)**
	• ***Estimate the Number of Relevant Images in Infinite Databases Using Two-Order Markov Chain*** Ms. Wang Xiaoling • ***A Local Region-Based Approach For Lip Contour Extraction Using Localized Active Contour Model*** Mr. Liu Xin • ***Automatic Segmentation of Color Lip Images Based on Morphological Filter*** Mr. Li Meng
12:10- 17:00	Break
17:00- 17:30	**Best Paper & Best Presentation Awards Announcement (OEE 601-3)**
Closing	

Table of Contents

Session I

Session II

Session III

Session IV

Session V

Session VI

A new BEMD method based on self-similar feature

JianJia Pan
Department of Computer Science, Hong Kong Baptist University
jjpan@comp.hkbu.edu.hk

Abstract

This paper presents a new method for texture analysis through Bidimensional Empirical Mode Decomposition (BEMD). Although there have been many filter based methods for texture analysis, problems of non-adaptively and redundancy are still hard to solve. The BEMD is a locally adaptive method and suitable for the analysis of nonlinear or nonstationary signals. The texture image can be decomposed to several IMFs (intrinsic mode functions) by BEMD, which present new characters of the images. But for the BEMD, the boundary interference is a main limit for its application. In this paper, we proposed a new BEMD method based on the self-similar extend method and the neighbor local extremes to reduce the boundary interference. This new method can get a lower orthogonality index (OI) of the IMFs, and the experiment result shown the new method also reduced the computation complex compared to other surface interpolation-based methods. And a denoising algorithm based on the new BEMD is also present in this paper.

1. Introduction

Empirical mode decomposition (EMD), developed by Huang [1], is a data driven processing algorithm, which applies no predetermined filter. The EMD is based on the local characteriristic scale of the data, which is able to perfectly analyze the nonlinear and nonstationary signals. The main processing of the EMD is to decompose the signal to its intrinsic mode functions (IMFs), and then those IMFs are analyzed by the Hilbert transform. This process is also known as Hilbert-Huang transforms (HHT) [1]. The EMD has been used to analyze the two-dimenational signals, for example, the images, which are known as bidimensional EMD (BEMD). BEMD has better quality than Fourier, wavelet and other decomposition algorithms in extracting intrinsic components of textures because of its data driven property [2, 4].

One topic of the BEMD questions is the boundary effect processing, which is to reduce the boundary effect to the intrinsic mode functions (IMFs) [3,4]. There are two ways. One way is to view the image as a long lengthened vector and then apply the one dimensional extended method to solve it [4, 6]. The other way is to extend the image by mirror extending, neural network training or AR model et. [11]. But in some cases, the marked change may also be occur in the image boundary, which make the boundary processing worse, or change the local extrema points. So the boundary processing is still a challenge problem for the BEMD.

Fractal geometry is a kind of new theory for studying non-linear complex systems. Fractal, characterized by self-similar structure, is proposed by Mandelbrot. The self-similar feature means that, irrespective of the complexity of the shape of an object, by looking deeply into its structure, an observer can be the same (or similar) shapes on contractible scales. Such objects exist widely in nature, such as coastlines, contour lines of mountains. With the rapid developments of fractal, fractal geometry has been applied broadly in image processing.

In this paper, we proposed a new BEMD method based on the self-similar extend method and the neighbor local extremes. This new method can reduce the orthogonality index (OI) of the IMFs, and the experiment result shown the new method also reduced the computation complex compared to other surface interpolation based methods. In section 4, based on the new BEMD, an image denoising algorithm is proposed.

2. Review of BEMD

2.1 1-D Empirical Mode Decomposition (EMD)

EMD is first proposed by Huang et al. [1] for the processing of non-stationary functions. The tool decomposes signals into components called Intrinsic Mode Functions (IMFs) satisfying the following two conditions:

(a).The numbers of extrema and zero-crossings must either equal or differ at most by one;

(b).At any point, the mean value of the envelope defined by the local maxima and the envelope by the local minima is zero.

Huang [1] have also proposed an algorithm called 'sifting' to extract IMFs from the original signal *f*(*t*) as follows:

$$f(t) = \sum_{i=1}^{N} I_i(t) + r_N(t) \qquad (1)$$

Where $I_i(t)$, i=1,…,N are IMFs and $r_N(t)$ is the residue.

Although the discussions about EMD lack concrete theoretical foundation until now, there have been some facts demonstrated by experiments. The most attractive one among the facts is that EMD acts as a dyadic filter band for 1-D fractional Gaussian noise and the quality of the obtained Intrinsic Mode Functions is related to the Hurst exponent [5].

2.2 The bidimensional EMD (BEMD)

The bidimensional EMD (BEMD) process is conceptually the same as the one dimension EMD, except that the curve fitting of the maxima and minima envelope now becomes a surface fitting exercise and the identification of the local extrema is performed in space to take into account for the connectivity of the points.

The main process of the BEMD can be described as:

(a).Locate the maximum and minimum points in the image *I(k)*;

(b).Interpolation the surface between the all maxima (resp. munima) to build the envelope *Xmax(k)* and *Xmin(k)*;

(c).Compute the mean envelope function

Xm(k) =(Xmax(k) +Xmin(k))/2; (2)

(d).Update the *I(k)=I(k-1)-Xm(k)* ;

(e).Check the stopping criterion

$$SD = \frac{1}{N}\sum_{k=0}^{K} \frac{(I_{i,j-1}(k) - I_{i,j}(k))^2}{I_{i,j-1}^2(k)} \qquad (3)$$

if SD is larger than a threshold *ε*, repeat the steps (a)-(e) with *I(k)* as the input, other wise, *I(k)* is an IMF *d(k)*;

(f).Update the residual *I(k)=I(k-1)-d(k)*;

(g).Input the *I(k)* to steps(a)-(e) until it can not be decomposed, and the last residual *I(k)=r(n)*.

After the BEMD, the decomposition of the image can be written as following form:

$$I(n) = \sum_{k=1}^{K} d_k(n) + r(n) \qquad (4)$$

The *d(k)* is the IMFs (intrinsic mode functions) of the images, and *r(n)* is the residual function.

3. The new BEMD details

With the intension of some difficult in implement BEMD, we used some methods to improve the BEMD. The local extrema are detected based on its neighbor and the extended parts are rebuilt based on self-similarity.

3.1 Local extrema detection

Detection the local extrema means finding the local maxima and minima points from given images. In the normal BEMD methods [1,4], the mathematical morphology method is used to local the extrema, but we find the extrema points will be reduced fast. It means that, after two or three surface interpolations, the image will be too smooth to local any significative extrema points. Neighbor location method [7] is used to detect the extrema in our method.

Definition 1: f[i,j] is a maximum (or. minimum) if it is larger (or. lower) than the value of *f* at the nearest neighbors of *[i,j]*.

Let *X* be an *M*×*N* 2D matrix represented by

$$X = \begin{bmatrix} x_{11} x_{12} \cdots x_{1N} \\ x_{21} x_{22} \cdots x_{2N} \\ \vdots \; \vdots \; \cdots \; \vdots \\ x_{M1} x_{M2} \cdots x_{MN} \end{bmatrix} \qquad (5)$$

x_{mn} is the element of *X* located in the *m*th row and *n*th column.

Let the window size for local extrema determination be *(2w+1)×(2w+1)*, Then,

$$x_{mn} = \begin{cases} Local\ Maximum & if\ x_{mn} > x_{ij} \\ Local\ Minimum & if\ x_{mn} < x_{ij} \end{cases} \qquad (6)$$

Where

$$x_{ij} = \{x \mid (m-w):i:(m+w), (n-w):j:(n+w)\}$$
$$i \neq m,\ j \neq n \qquad (7)$$

From the experimental, we find *3×3* window results in an optimum extrema map for a given image. The larger windows are also used in some conditions to reduce the computation, but as the mathematical morphology method, the extrema points will be reduced fast.

3.2 Surface interpolation method

Another difficulty in the BEMD comes from generating a smooth fitting surface to the identified maxima and minima. There are several interpolation methods for BEMD. Nunes [2, 8] used the radial basis function (RBF) for surface interpretation. Linderhed [9] used the spline for surface interpretation to develop two-dimensional EMD data. Damerval [10] used a third way based on Delaunay triangulation to obtain an upper surface and a lower surface.

Delaunay triangulation can effectively reduce the interpolation computation. Our interpolation method is based on a Delaunay triangulation. The subset Dmax (resp. Dmin) of D is corresponds to maxima (resp. minima) for f and then on RBF interpolation on triangles as explained in [8]. Delaunay triangulation is well adapted to the interpolation of scattered data for the set Dmax (resp. Dmin).

3.3 Self-similar for BEMD Boundary Processing

A self-similar object is exactly or approximately similar to a part of itself, which means the whole has the same shape as one or more of the parts. Many objects in the real world are statistically self-similar: parts of them show the same statistical properties at many scales. Self-similarity is a typical property of fractals.

A compact topological space X is self-similar if there exists a finite set S indexing a set of non-subjective homeomorphisms $\{f_s\}_{s\in S}$ for which

$$X = \bigcup_{s\in S} f_s(X) \tag{8}$$

If $X \subset Y$, we call X self-similar if it is the only non-empty subset of Y such that the equation above holds for $\{f_s\}_{s\in S}$. We call

$\Gamma = (X, S, \{f_s\}_{s\in S})$ as *self-similar structure*.

The self-similar feature means that, irrespective of the complexity of the shape of an object, by looking deeply into its structure, an observer can be the same (or similar) shapes on contractible scales. In the boundary process, we use this self-similar property to build the extend boundary. The basic idea is: the extend part can find a self-similar part in the original image.

The concrete algorithm is as follows:

Assume the size of original image I is $N\times N$. The size of the extend block is $M\times M$. After extending, the extended image is $(N+2M)\times(N+2M)$ with middle $N\times N$ block the original image. The original image I is divided to $M\times M$ size blocks. For each extended block $part_e$, its three neighbor blocks in the original image are defined as its neighbor blocks $part_n$. And then in the image I, find the blocks which are the most similar to the $part_n$. The similar judgment criterion is based on the MAD (Mean Absolute Difference) for representing the distances different between boundary blocks and the matched blocks. At last, the block with most similar neighbor blocks is used as the extend block.

After the self-similar based extension boundary processing, the boundary interference of the BEMD will be reduced, and the IMF components is more significant.

4. Image denoising based on the new BEMD

The EMD has found a vast number of diverse applications such as biomedical [13], watermarking [14], and audio processing [15]. A more generalized task in which EMD can prove useful is image denoiseing. [12]

The EMD used for denoising tool were developed both by Flandrin[17] and Wu[16], using the statistical analysis model. The energy of the IMFs of a noise-only signal with certain characteristics is known, then in actual cases of signals comprising both information and noise following the specific characteristics, a significant discrepancy between the energy of a noise-only IMF and the corresponding noisy-signal IMF indicates the presence of useful information.[12]

In this part, we will show an image denoisig method based on the new BEMD. First, the noised image is decomposed by the BEMD, and then, the IMFs are denoised by the denoising factor by different paramters, at last, the denoised components are composed.

This denoising factor is less than 1 and will decrease when the absolute value of the coefficients increases. This algorithm considers energy distribution property of the decomposition of the image globally, and obtains a better denoising result. The realization process is as follows:

$$F(x,y) = \begin{cases} w(x,y) & |w(x,y) \geq 2\sigma| \\ 0 & |w(x,y) \leq |aver|| \\ w(x,y)\times k & else \end{cases} \tag{5}$$

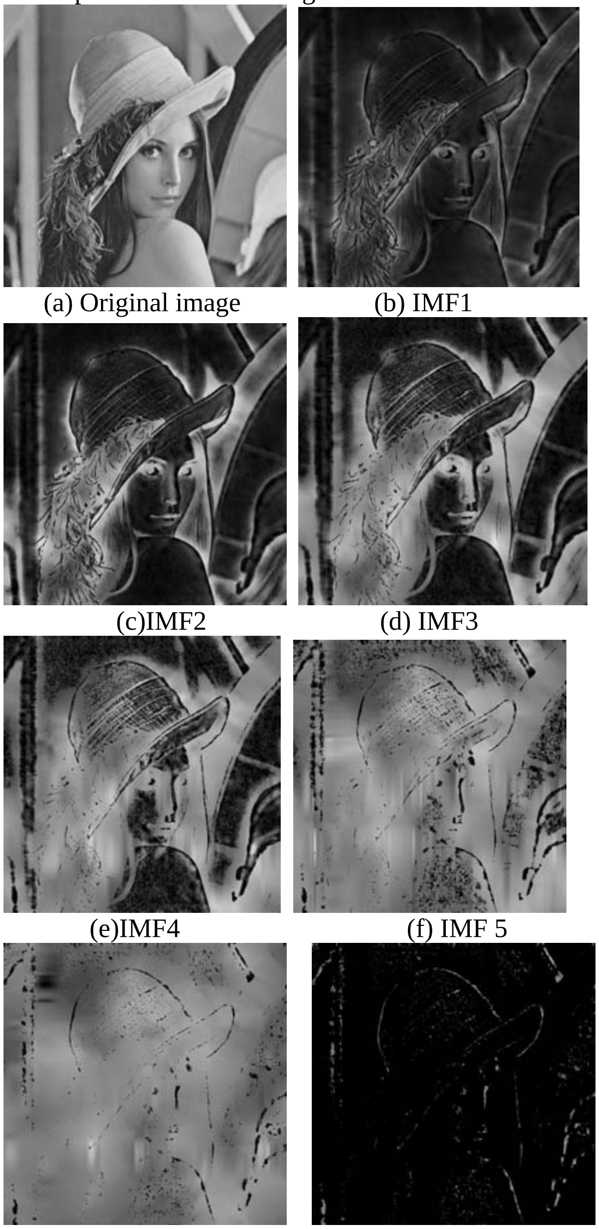
Where *w(x,y)* denotes the coefficient, *F(x,y)* denotes the coefficient gained after denoising; *σ*, *aver* indicate the variance and the mean value from different decomposition levels and different directions [18].

5. Experimental results

5.1 the new BEMD decomposition result

In this section, we present the experimental results of the image to decompose by using the self-similar extension and the neighbor local extremes.

Image can be regarded as nonlinear signal in two-dimensional. The BEMD can decompose the signals adaptively so it is suitable for analyzing the image. The proposed BEMD create a filter bank when applied to the texture images. Figure 1 shows a decomposition result of image.



(a) Original image (b) IMF1

(c)IMF2 (d) IMF3

(e)IMF4 (f) IMF 5

(g)IMF 6 (h) Residue

Figure1 Original image and its decomposition IMFs

The orthogonaltiy index (OI), has been proposed for IMFs in [1], the extended formula for two dimensional is defined as follows:

$$OI = \sum_{x=1}^{M}\sum_{y=1}^{N}\left(\sum_{i=1}^{K+1}\sum_{j=1}^{K+1}\frac{C_i(x,y)C_j(x,y)}{\sum C^2(x,y)}\right) \qquad (9)$$

A low value of the OI indicated a good decomposition in terms of local orthogonality among the IMFs [4,7]. The texture image's OI by our proposed method and some other methods are shown in the Table 1.

Methods	OI	Consuming time (s)
Nunes[8]	0.0033	68.52
Linderhed [9]	0.0029	60.51
Our method	0.0025	46.59

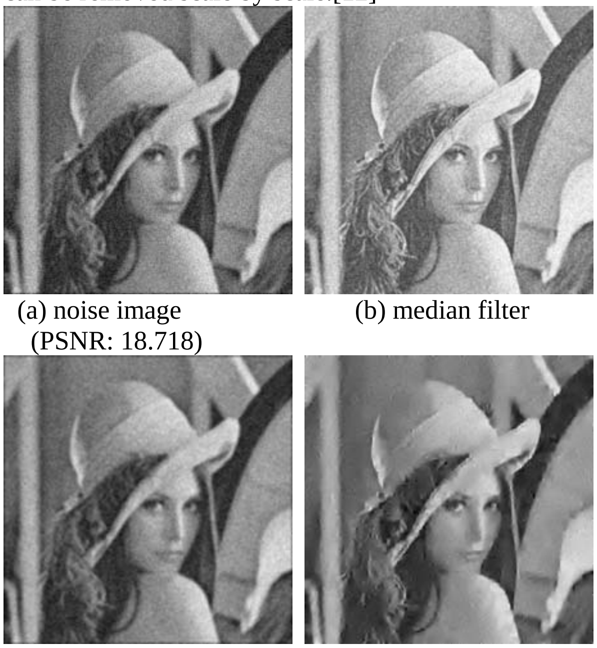
Table 1. Compared result of Orhogonaltiy index (OI) and the consuming time

As the Table 1 shown, our method can reduce the OI, which improve the BEMD method, and the consuming time is also reduced.

5.2 The new-BEMD based image denoising

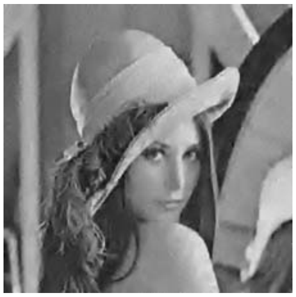
In this part, results of the denoising method discussed in section 4 are present.

The traditional denoising algorithms, such as median filter and Wiener filter, can reduce the Gaussian noise and salt & pepper noise. But the PSNR of the denoise result is still not idea; the noise is still in the image. BEMD method decomposed the image to many IMFs, which are basically the image portion contained in each adaptive subband. By use of the factor, the noise can be removed scale by scale.[12]



(a) noise image (PSNR: 18.718) (b) median filter

(c) Wiener filter (d) Nunes[8]+ factor



(e) Our approach

Figure 2. noise image and compared denoised image

The result shown that based on the BEMD and denoise factor, the noise can be reduced more effetely than the traditional methods. And the new BEMD's denoise result is also better than the Nunes' BEMD.

Output PSNR Input PSNR	Median filter	Wiener filter	Nunes[8]+ factor	BEMD + factor
18.718	24.323	24.59	26.287	29.31
16.146	22.947	22.61	24.599	27.307
14.508	21.66	21.352	24.402	26.503

Table 2 denoising result with output PSNR

6. Conclusion and future work

Aimining for the boudary effects of BEMD, a new BEMD method is proposed, which is based on the self-similar feature and the neighbor local extremes. This new method can reduce the orthogonality index (OI) of the IMFs, and the experiment result shown the new method also reduced the computation complex compared to other surface interpolation based BEMD methods.

Based on the new BEMD method, we proposed a image denoising method, compare with the normal denoising algorithm, the BEMD-based method can achieve a good result. In the future, we will focus on developing a better BEMD method to reduce the OI and more research about the BEMD application in image processing and pattern recognition.

References

[1] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, et. "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis". Proceedings of Royal Society. Lond, (A) vol.454, pp:903-1005.1998

[2] J.C.Nunes, Y. Bouaoune, E. Delechelle, O.Niang, Ph. Bunel. "Image analysis by bidimensional empirical mode decomposition". Image and Vision Computing Vol.21(12) , pp:1019-1026, November 2003

[3] C.Damerval, S.M eignen, and V.Perrier. "A fast algorithm for bidimensional EMD", IEEE signal processing letters, vol.12,no.10, pp 701-704, 2005

[4] N.E.Huang, M.L.C.Wu, S.R.Long, "A confidence limit for the EMD and Hilerbet spectral analysis", Proceeding of the royal society A, vol 459., pp 2317-345, 2003

[5]P. Flandrin, G. Rilling and P. Goncalves. "Empirical mode decompositionas a filter bank". IEEE Signal Processing Letters, vol.11(2),pp:112-114, 2004.

[6]Bhagavatula,R., MariosSavvides, and M. "Acoustics. Analyzing Facial Images using Empirical Mode Decomposition for Illumination Artifact Removal and Improved Face Recognition." IEEE International Conference on Speech and Signal Processing, 2007. Vol. 1, pp. 505-508. April 2007

[7] Sharif M. A. Bhuiyan, Reza R. Adhami, and Jesmin F. Khan. "Fast and Adaptive Bidimensional Empirical Mode Decomposition Using Order-Statistics Filter Based Envelope Estimation". EURASIP Journal on Advances in Signal Processing, vol. 2008, Article ID 728356, 18 pages, 2008.

[8] Nunes J. C., Guyot S., and Deechelle E. "Texture analysis based on local analysis of the Bidimensional Empirical Mode Decomposition". Machine Vision and Applications vol.16(3), pp. 177-188, 2005.

[9] Linderhed, A., "Variable sampling of the empirical mode decomposition of two-dimensional signals," International Journal of Wavelets, Multiresolution and Information Processing., vol.3, 435–452, 2005

[10] Damerval, C., S. Meignen, and V. Perrier, "A fast algorithm for bidimensional EMD", IEEE Signal Processing Letter, vol.12(10), 701–704, October, 2005.

[11] Zhao Na, "A Periodic Extension Approach for HHT Empirical Mode Decomposition", Computer Simulation, vol.25(12) 2008

[12]Yannis Kopsinis,and Stephen McLaughlin, "Development of EMD-Based Denoising Methods Inspired by Wavelet Thresholding" IEEE Transactions on Signal Processing, vol. 57, no. 4, April 2009 1351-1362

[13]L. Hadjileontiadis,"Empirical mode decomposition and fractal dimension filter," IEEE Eng. Med. Biol. Mag., pp. 30–39, Jan. 2007.

[14] B. Ning, S. Qiyu, Y. Zhihua, H. Daren, and H. Jiwu, "Robust image watermarking based on multiband wavelets and empirical mode decomposition," IEEE Trans. Image Process., pp. 1956–1966, Aug. 2007.

[15] M. K. I. Molla and K. Hirose, "Single-mixture audio source separation by subspace decomposition of Hilbert spectrum," IEEE Trans. Audio, Speech Lang. Process., pp. 893–900, Aug. 2007.

[16] N. E. Huang and Z. Wu, "Statistical significance test of intrinsic mode functions," in Hilbert-Huang Transform and Its Applications, N. E. Huang and S. Shen, Eds., 1st ed. Singapore: World Scientific, 2005.

[17]P. Flandrin, G. Rilling, and P. Gonçalvès, “EMD equivalent filter banks, from interpetation to applications,” in Hilbert-Huang Transform and Its Applications, N. E. Huang and S. Shen, Eds., 1st ed. Singapore: World Scientific, 2005.

[18]Lanyan Xue, Jianjia Pan, “Edge Detection Combining Wavelet Transform and Canny Operator Based On Fusion Rules” ,Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, 12-15 July 2009 pp:324-328

Face Recognition based on the New Phase Local Features

Dan Zhang

Abstract

The Hilbert-Huang transform (HHT) is a novel signal processing method which can efficiently handle nonstationary and nonlinear signals. It firstly decomposes signals into a series of Intrinsic Mode Functions (IMFs) adaptively by the Empirical Mode Decomposition (EMD), then applies the Hilbert transform on the IMFs afterward. Based on the analytical signals obtained, the local analysis of the IMFs are conducted. This paper contains two main works. First, we proposed a new two-dimensional EMD (2DEMD) method, which is faster, better-performed than the current 2DEMD methods. Second, the Riesz transform are utilized on the 2DIMFs to get the 2D analytical signals. The local features (amplitude, phase orientation, phase angle, etc) are evaluated and used in face recognition.

1 Introduction

Texture [1] is ubiquitous and provides powerful characteristics for many image analysis applications such as segmentation, edge detection, and image retrieval. Among various texture analysis methods, signal processing methods are promising, which include Gabor filters [2], wavelet transforms [3], Wigner distributions and so forth. They characterize textures through filter responses directly.

Hilbert-Huang transform is a new signal processing method proposed by Huang et al [8, 10, 9]. It contains two kernel parts: Empirical Mode Decomposition (EMD) and Hilbert transform. First, it decomposes signals into a series of Intrinsic Mode Functions (IMFs) adaptively by the Empirical Mode Decomposition (EMD), then applies the Hilbert transform on the IMFs afterward. Based on the analytical signals obtained, the local analysis of the IMFs are conducted. Though Fourier spectral analysis and wavelet transform have provided some general methods for analyzing signals and data, they are still weak at non-stationary and nonlinear data processing. However, due to the fully data-driven process, the HHT is more efficient in this situation. It provides an efficient way for the local analysis.

As the kernel part of HHT, EMD works as a filter bank. It has a wide application in signal analysis including ocean waves, rogue water waves, sound analysis, earthquake time records as well as image analysis [19, 18, 16, 21, 20, 22, 23]. The EMD has been extended to 2DEMD. However, it countered a lot of difficulties such as inaccuracy of surface interpolation, high computational complexity and so forth [27, 29, 25, 26, 28, 31]. Finding a powerful 2DEMD is still a challenge. In this paper, we implemented a modified 2DEMD and study the local properties of the 2DIMFs by Riesz transform [4, 5]. The image after its Riesz transform, we can get the 2D analytical signal. The estimation of the local features is crucial in image processing. Generally, structures such as lines and edges can be distinguised by the local phase, the local amplitude can be used for edge detection.

This paper is organized as follows: Section 2 presents an introduction to HHT. In Section 3, the details of the modified 2DEMD are shown firstly, then we reviewed the Riesz transform. The simulation results are demonstrated in Section 4. Finally, the conclusions are given.

2 Hilbert-Huang transform

Hilbert-Huang transform was proposed by N.E.Huang in 1998. It contains two parts in terms of Empirical Mode Decomposition (EMD) and Hilbert transform. The signals are decomposed into a series of Intrinsic Mode Functions (IMFs), then Hilbert transform are applied on these IMFs to get analytic signals. Since this method is local, data-driven, it is capable of handling nonlinear and non-stationary signals.

EMD captures information about local trends in the signal by measuring oscillations, which can be quantized by a local high frequency or a local low frequency, corresponding to finest detail and coarsest content. Here we briefly review the sifting process of EMD. Four main steps are contained, S1, S2, S3 and S4 are abbreviation for Step 1 to Step 4. Given a signal x(t),

S1. Identify all the local minima and maxima of the input signals $x(t)$;

S2. Interpolate between all minima and maxima to yield two corresponding envelopes $E_{max}(t)$ and $E_{min}(t)$.

Calculate the mean envelope $m(t) = (E_{max}(t) + E_{min}(t))/2$;

S3. Compute the residue $h(t) = x(t) - m(t)$. If it is less than the threshold predefined then it becomes the first IMF, go to Step 4. Otherwise, repeat Step 1 and Step 2 using the residue $h(t)$, until the latest residue meets the threshold and turns to be an IMF;

S4. Input the residue r(t) to the loop from Step 1 to Step 3 to get the next remained IMFs until it can not be decomposed further.

The analytical signal provides a way to compute the 1D signal's local amplitude and phase, which is obtained by the Hilbert transform on a real signal. The Hilbert transform $f_H(x)$ of a real 1D signal f is given by:

$$f_H(x) = f(x) * \frac{1}{\pi x},$$

where $*$ is convolution. $f_H(x)$ is the imaginary part of the signal. The analytical signal can be written as

$$f_A = f(x) + i f_H(x) = a(t)e^{i\theta(t)},$$

in which, $a(t)$ is the amplitude, $\theta(t)$ is the phase.

Applying Hilbert transform on each IMF can evaluate the local properties such as amplitude and phase.

3 Local Analysis of 2DIMFs

3.1 The improved 2DEMD

Here we propose an alternative algorithm for EMD. Instead of using the envelopes generated by splines we use a low pass filter to generate a "moving average" to replace the mean of the envelopes. The essence of the sifting algorithm remains.

The moving average is the most common filter in digital signal processing. It operates by averaging a number of points from the input signal to produce each point in the output signal, it is written:

$$y[i] = \frac{1}{M}\sum_{j=0}^{M-1} x[i+j],$$

where $x[]$ is the input signal, $y[]$ is the output signal, and M is the number of points used in the moving average. It is actually a convolution using a simple filter $[a_i]_{i=1}^{M}$, $a_i = \frac{1}{M}$, and $[A_{i,j}]_{i=1,j=1}^{M,N}$, $A_{i,j} = \frac{1}{M \times N}$ for the 2-dimensional case.

Detection of local extrema means finding the local maxima and minima points from the given data. No matter for 1D signal or 2D array, neighboring window method is employed to find local maxima and local minima points. The data point/pixel is considered as a local maximum (minimum) if its value is strictly higher (lower) than all of its neighbors.

We illustrated 1-dimensional case and 2-dimensional case separately.

- 1-dimensional case:
  For each extrema map, the distance between the two neighborhood local maxima (minima, extrema, zero-crossing) has been calculated called as adjacent maxima (minima, extrema, zero-crossing) distance vector Adj_max ($Adj\_min, Adj\_ext, Adj\_zer$). Four types of window size:
  - Window-size I: $\max(Adj\_max)$;
  - Window-size II: $\max(Adj\_min)$;
  - Window-size III: $\max(Adj\_zer)$;
  - Window-size IV: $\max(Adj\_ext)$.
- 2-dimensional case:
  The window size for average filters is determined based on the maxima and minima maps obtained from a source image. For each local maximum (minimum) point, the Euclidean distance to the nearest local maximum (minimum) point is calculated, denoted as adjacent maxima (minimum) distance array Adj_max (Adj_min).
  - Window-size I: $\max(Adj\_max)$;
  - Window-size II: $\max(Adj\_min)$;

3.2 Monogenic signal

The analytic signal is the basis for all kinds of approaches which makes use of the local phase. The combination of a 1D signal and its Hilbert transform is called the analytic signal. Similarly, the combination of a image and its Riesz transform, which is the generalization of Hilbert transform, is called the monogenic signal [4, 5].

The monogenic signal is often identified as a local quantitative or qualitative measure of an image. Different approaches to an nD analytic or complex signal have been proposed in the past:

- Total Hilbert Transform, The Hilbert transform is performed with respect to both axes:

  $$H_T(\overrightarrow{v}) = j sign(v_1) sign(v_2)$$

  This transform is not a valid generalization of the Hilbert transform since it does not perform a phase shift of $\pi/2$.It can't meet orthogonality.

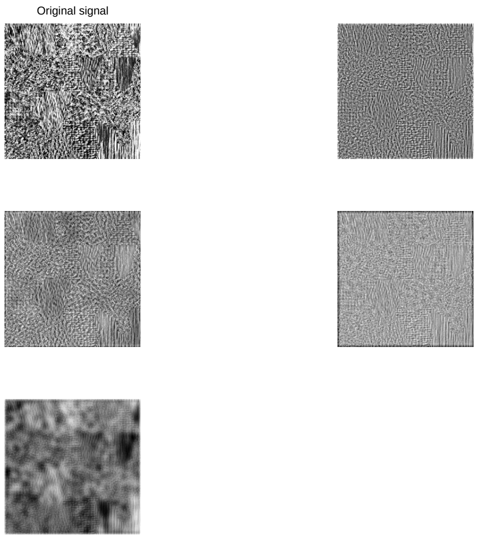


Figure 1. The 2DIMFs obtained by the improved 2DEMD.

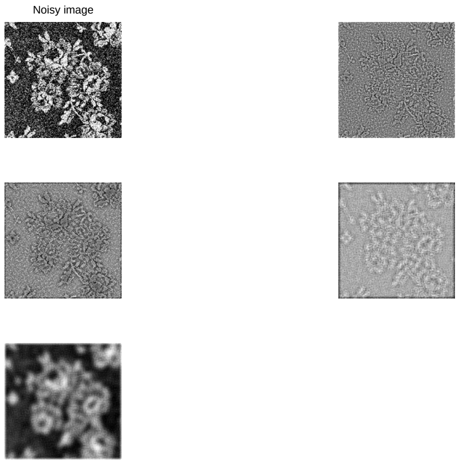


Figure 2. The 2DIMFs obtained by the improved 2DEMD on the noisy image.

- Partial Hilbert Transform, The Hilbert transform is performed with respect to a half-space that is chosen by introducing a preference direction:

  $$H_T(\overrightarrow{v}) = jsign(\overrightarrow{v}, \overrightarrow{d}),$$

  where $\overrightarrow{v} = (v_1, v_2)$ is one 2D vector, $\overrightarrow{d}$ is one preference direction. This transform missed the isotropy.

- Total Complex Signal,

- Hypercomplex Signal.

The Riesz transform [4, 5] is a multidimensional generalization of the Hilbert transform. The expression of Riesz transformed signal in the frequency domain:

$$F_R(\overrightarrow{v}) = \frac{i\overrightarrow{v}}{\overrightarrow{v}}F(\overrightarrow{v}) = H_2(\overrightarrow{v})F(\overrightarrow{v}),$$

where the transfer function H_2 of the Riesz transform is the generalization of the Hilbert transform, the corresponding space representation of Riesz transform is:

$$f_R(\overrightarrow{x}) = -\frac{\overrightarrow{x}}{2\pi|\overrightarrow{x}|^3} * f(\overrightarrow{x}) = h_2(\overrightarrow{x}) * f(\overrightarrow{x}).$$

The Riesz transformed signal and the original signal constitutes the 2D analytical signal, this is the monogenic signal.

$$f_M(\overrightarrow{x}) = f(\overrightarrow{x}) - (i,j)f_R(\overrightarrow{x}).$$

From this formulation, we see the 2D analytical signal is 3D vector and we can get the local features of the monogenic signal.

- Phase: Phase as we all know the polar representation of the complex $z = x + iy$ is $(r, \varphi) = (\sqrt{z\overline{z}}, \arg(z))$. Where $\overline{z}$ is the conjugate of z, $\arg(z)$ is the phase of the complex: $\arg(z) = a\tan 2(y,x) = sign(y)a\tan(|y|/x)$, $sign(y)$ represents the direction of rotation. The phase of the 2D analytical signal is:

  $$a\tan 3(y,x) = \frac{\overrightarrow{x}_D}{|\overrightarrow{x}_D|}a\tan(\frac{|\overrightarrow{x}_D|}{< (0,0,1)^T, \overrightarrow{x} >}),$$

  where $\overrightarrow{x}_D = (0,0,1)^T \times \overrightarrow{x}$ yields the direction of the rotation vector. The phase of the monogenic signal is:

  $$\varphi(\overrightarrow{x}) = a\tan 3(f_M(\overrightarrow{x})) = \arg(f_M(\overrightarrow{x})).$$

- Amplitude: The local amplitude of $f_M(\overrightarrow{x})$ is:

  $$|f_M(\overrightarrow{x})| = \sqrt{f_M(\overrightarrow{x})\overline{f_M(\overrightarrow{x})}} = \sqrt{f^2(\overrightarrow{x}) + |f_R(\overrightarrow{x})|^2},$$

  given the local phase $\varphi(\overrightarrow{x})$ and the local amplitude $|f_M(\overrightarrow{x}|$ of a monogenic signal, it can be reconstructed by

  $$f_M(\overrightarrow{x} = |f_M(\overrightarrow{x}|\exp((-j,i,0)\varphi(x)).$$

2DEMD permits extracting multiscale components. The monogenic signal of each IMF permits to compute local amplitude, local phase and the local direction. We have shown this feature through experiment results for both natural textures and synthetic textures.

4 Experimental Results

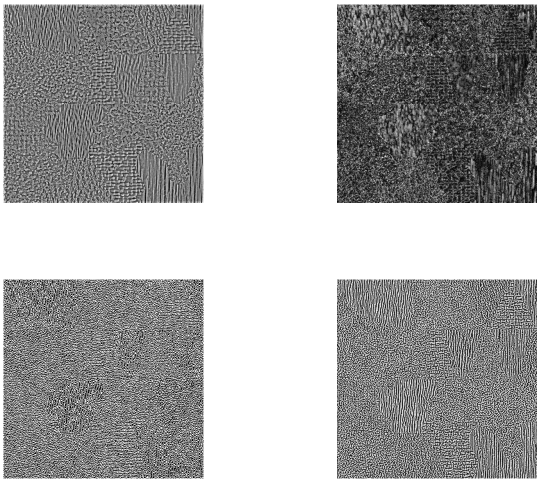


Figure 3. left-up: 1st IMF, right-up: amplitude, left-down: phase orientation, right-down: phase angle.

In all our numerical experiments we determine the window size in each decomposition with Window-size I. Unless otherwise specified we use $\alpha = 0.5$ for our stopping criterion.

We show the local features amplitude, phase orientation, phase angle extracted by SMV of the 1st IMF.

By having access to these representations of scenes or objects, we can concentrate on only one or several modes (one individual or several spatial frequency components) rather than the image entirety. The improved 2DEMD and Riesz local analysis offer a new and more promising way to analyze texture images.

5 Conclusions

This paper contains two main works. First, we proposed a new two-dimensional EMD (2DEMD) method, which is faster, better-performed than the current 2DEMD methods. Second, the Riesz transform are utilized on the 2DIMFs to

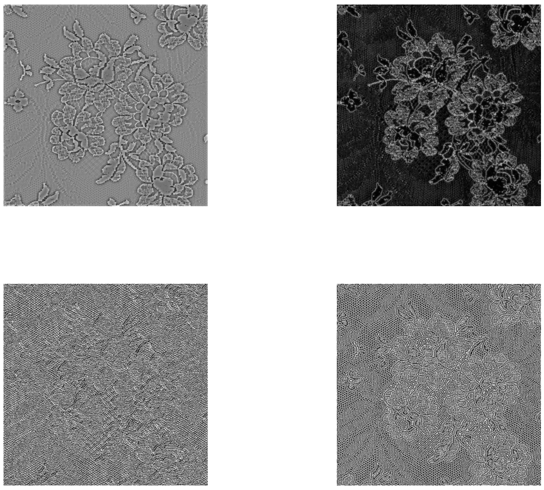


Figure 4. left-up: 1st IMF, right-up: amplitude, left-down: phase orientation, right-down: phase angle.

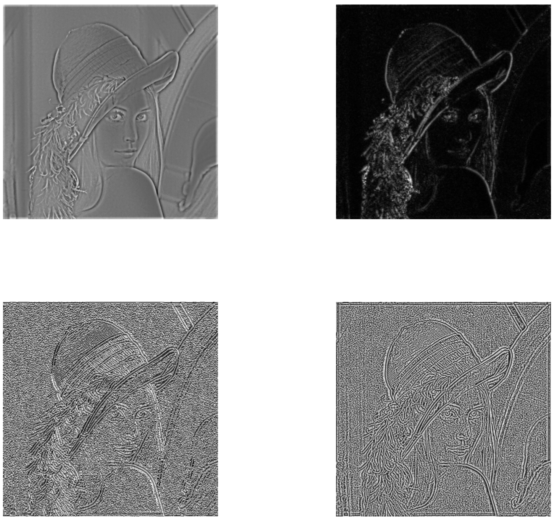


Figure 5. left-up: 1st IMF, right-up: amplitude, left-down: phase orientation, right-down: phase angle.

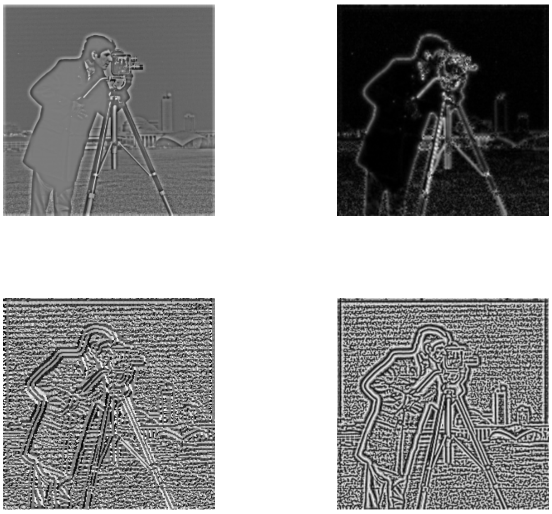


Figure 6. left-up: 1st IMF, right-up: amplitude, left-down: phase orientation, right-down: phase angle.

get the 2D analytical signals. The local features (amplitude, phase orientation, phase angle, etc) are evaluated. The performances are demonstrated on both texture images and natural images.

References

[1] Materka A and Strzelecki M, Texture analysis method: a review. *COST B11 report, Technical University of Lodz.*

[2] A. K. Jain and Farrokhnia F, Unsupervised texture segementation using Gabor filters. *Pattern Recognition, 24 (12), pp. 1167-1186, 1991.*

[3] Mallat S, Wavelets for a vision. *Proceedings of the IEEE, Vol. 84, Iss. 4, pp. 604-614, 1996.*

[4] Felsberg M and Sommer. G. The monogenic signal. *IEEE Trans Signal Process 49, 2001.*

[5] Felsberg M and Sommer. G. The Monogenic Scale-Space: A Unifying Approach to Phase-Based Image Processing in Scale-Space. *2004.*

[6] Felsberg M and Sommer. G. Structure multivector for local analysis of images. *Workshop on theoretical foundations of computer vision, SpringerVerlag, pp 93-104, 2001.*

[7] Felsberg M and Sommer. G. Low Level Image Processing with the Structure Multivector. *2002.*

[8] N. E. Huang, Z. Shen, S. R. Long, et al.. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A, vol. 454, no. 1971, pp. 903C995, 1998.*

[9] HILBERT-HUANG TRANSFORM AND ITS APPLICATIONS. *Book in Interdisciplinary Mathematical Sciences, Vol. 5, edited by N. E. Huang, and Samuel S P Shen, 2005.*

[10] N. E. Huang, M. L. C. Wu, S. R. Long, et al.. A confidence limit for the empirical mode decomposition and Hilbert spectral analysis. *Proceedings of the Royal Society A, vol. 459, no. 2037, pp. 2317C2345, 2003.*

[11] S. R. Long. Applications of HHT in image analysis. in Hilbert-Huang Transform and Its Applications, N. E. Huang and S. S. P. Shen, Eds., World Scientific, River Edge, NJ, USA, 2005.

[12] Harishwaran Hariharan, Andrei Gribok, Besma Abidi, and Mongi Abidi. Multi-modal Face Image Fusion using Empirical Mode Decomposition. *The Biometrics Consortium Conference, Crystal City, VA, September 2005.*

[13] H. Hariharan, A. Koschan, B. Abidi, A. Gribok, and M.A. Abidi. Fusion of visible and infrared images using empirical mode decomposition to improve face recognition. *IEEE International Conference on Image Processing ICIP2006, Atlanta, GA, pp. 2049-2052, October 2006.*

[14] Bhagavatula, R., Marios Savvides, and M. Acoustics. Analyzing Facial Images using Empirical Mode Decomposition for Illumination Artifact Removal and Improved Face Recognition. *IEEE International Conference on Speech and Signal Processing, 2007 (ICASSP 2007). Vol. 1, Issue , 15-20 April 2007 pp. I 505-508.*

[15] J. C. Nunes, Y. Bouaoune, E. Delechelle, O. Niang, and Ph. Bunel. Image analysis by bidimensional empirical mode decomposition. *Image and Vision Computing Volume 21, Issue 12, Pages 1019-1026, November 2003.*

[16] Nunes J. C., Guyot S., and Deléchelle E. Texture analysis based on local analysis of the Bidimensional Empirical Mode Decomposition. In *Machine Vision and Applications 16, 3, pp. 0932-8092, 2005.*

[17] A. Linderhed. 2-D empirical mode decompositions in the spirit of image compression. *in Wavelet and Independent Component Analysis Applications IX, vol. 4738 of Proceedings of SPIE, pp. 1C8, Orlando, Fla, USA, April 2002.*

[18] A. Linderhed. Compression by image empirical mode decomposition. *IEEE International Conference on Image Processing, 2005 (ICIP 2005), Vol. 1, pp. I 553-6, 2005.*

[19] H. Hariharan, A. Gribok, M. Abidi, and A. Koschan. Image Fusion and Enhancement via Empirical Mode Decomposition. *Journal of Pattern Recognition Research, Vol. 1, No. 1, pp. 16-32, January 2006.*

[20] Sinclair, S. and Pegram, G. G. S. Empirical Mode Decomposition in 2-D space and time: a tool for space-time rainfall analysis and nowcasting. *Hydrol. Earth Syst. Sci. Discuss., 2, 289-318, 2005.*

[21] Jian Wan, Longtao Ren, and Chunhui Zhao. Image Feature Extraction Based on the Two-Dimensional Empirical Mode Decomposition. *2008 Congress on Image and Signal Processing, Vol. 1, pp. 627-631, 2008.*

[22] Jalil Taghia, Mohammad Ali Doostari and Jalal Taghia. An Image Watermarking Method Based on Bidimensional Empirical Mode Decomposition. *2008 Congress on Image and Signal Processing, Vol. 5, pp. 674-678, 2008.*

[23] Fauchereau, N., Sinclair, S., and Pegram, G. 2-D Empirical Mode Decomposition on the sphere, application to the spatial scales of surface temperature variations. *Hydrol. Earth Syst. Sci. Discuss., 5, 405-435, 2008.*

[24] C. Damerval, S. Meignen, and V. Perrier. A fast algorithm for bidimensional EMD. *IEEE Signal Processing Letters, vol. 12, no. 10, pp. 701C704, 2005.*

[25] Sharif M. A. Bhuiyan, Reza R. Adhami, and Jesmin F. Khan. Fast and Adaptive Bidimensional Empirical Mode Decomposition Using Order-Statistics Filter Based Envelope Estimation. *EURASIP Journal on Advances in Signal Processing, vol. 2008, Article ID 728356, 18 pages, 2008.*

[26] Sharif M. A. Bhuiyan, Reza R. Adhami, and Jesmin F. Khan. A novel approach of fast and adaptive bidimensional empirical mode decomposition. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 (ICASSP 2008), pp. 1313-1316, 2008.*

[27] Sherman Riemenschneider, Bao Liu, Yuesheng Xu and Norden E. Huang. B-spline based empirical mode decomposition. *Hilbert-Huang Transform and Its Applications, Book chapter 2, 2005.*

[28] Louis Yu Lu. Fast intrinsic mode decomposition of time series data with sawtooth transform. *Technical report, Nov 2007.*

[29] Yong-Ping Huang, Xue-Yao Li and Ru-Bo Zhang. A research on local mean in empirical mode decomposition. *Proceedings of the 7th international conference on Computational Science, Part III: ICCS 2007, Lecture Notes In Computer Science, Vol. 4489, pp. 125-128.*

[30] Luan Lin, Yang Wang, and Haomin Zhou. A new approach to empirical mode decomposition. *Preprint.*

[31] Lixin Shen. Local mean and empirical mode decomposition. *Report on The Second International Conference on the Advances of Hilbert-Huang Transform and Its Applications, Guangzhou, Dec 2008. Preprint.*

Binary Discriminant Analysis for Face Template Protection

Yicheng Feng

Abstract

Biometric cryptosystem (BC) is a very secure approach for template protection because the stored template is encrypted. The key issues in BC approach include (i) limited capability in handling intra-class variations and (ii) binary input is required. To overcome these problems, this paper adopts the concept of discriminative analysis and develops a new binary discriminant analysis (BDA) method to convert a real-valued template to a binary template. Experimental results on CMU-PIE and FRGC face databases show that the proposed BDA method outperforms existing template binarization schemes.

1 Introduction

Biometrics is a reliable, robust and convenient way for person authentication [5, 3]. With the growing use of biometrics, there is a rising concern about the security and privacy of the stored biometric templates. Biometric cryptosystem [4] is a very secure approach for template protection because the output is encrypted, but suffers from two major limitations. First, this approach requires binary input while most of the templates are real-valued. Therefore, a template binarization step is required. Second, the capability for handling the intra-class variations is limited. In turn, the recognition accuracy may not be satisfied.

To fulfill the binary input requirement, some binarization schemes [8, 9, 10, 11, 12, 13, 14, 15] have been proposed in the last few years. We roughly categorize these schemes into two approaches, namely local and global. Local binarization methods consider each component x_r of the input real-valued template $\mathbf{x} = (x_1, x_2 \ldots x_l)$ (feature vector) independently. For each x_r, a function f_r is applied to extract a bit b_r (or maybe several bits) from x_r ($b_r = f_r(x_r)$). The advantages of local approach are simple and low complexity but, it will distort the original data distribution. In turn, the system performance will be degraded. Global binarization methods consider template as a whole. A series of functions $f_1, f_2 \ldots f_n$ are constructed to extract bits $b_1, b_2 \ldots b_n$ from template $\mathbf{x}$. That is, $b_r = f_r(\mathbf{x})$. The advantage of this approach is that the binary templates could preserve the original real-valued template data distribution and discriminability, Therefore, the recognition performance of global methods, normally, outperform local methods. However, existing methods are either designed only for authentication system [10] or fingerprint biometric [11].

To overcome the limitations on existing binarization algorithms, this paper proposes a new binary discriminant analysis method to convert a real-valued template to binary template while the discriminability is maximized. The proposed method follows the global template approach and can be used for both authentication and identification. The idea of the proposed method is inspired by the linear discriminant function (LDF) for classification process. When applying in our context, each linear discriminant function divides the biometric template space into two subspaces and each subspace is then represented by a bit (either ”1” or ”0”) as illustrated in Figure 1(a). Therefore, when more linear discriminant functions are used, the space will be divided into a number of subspaces. Each subspace can then be represented by a binary bit string. In turn, all templates within a subspace will have the same binary representation as illustrated in Figure 2. So the problem is how to determine the ”optimal” set of linear discriminative functions. Details are discussed in next section.

2 Binary Discriminant Analysis

The rationale of our binary discriminant analysis algorithm is illustrated in Figure 2. We would like to find the binarization function f from the training data with class label information. In BC approach, the bio-cryptographic algorithms, such as fuzzy commitment scheme [6] and fuzzy vault scheme [7], could only handle a small image variations. Feng $etal.$ [10] have suggested that, ideally, all real-valued templates in the same class should be mapped to the same point in binary space. In practice, it is hard to achieve and therefore, we hope that all real-valued templates should be mapped to the same point in binary space as much as possible. However, it is very hard, if not impossible, to solve the multi-class optimization problem in binary space. Instead, we propose a new idea to tentatively fix the training class data center, called ideal centroids, and BCH code [1] is employed. In doing so, the ”optimal” binary function can

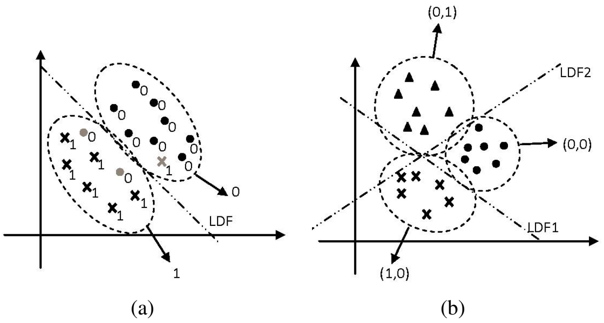


Figure 1. Linear discriminant functions for binarization. (a) Single LDF for binarization. The bits assigned to each point are the ideal labels. And the LDF labels the two subspaces to 0 and 1 respectively. (b) Multi LDF for binarization.

be determined using iterative gradient decent method.

In this section, we will first discuss how to make use of linear discriminant function for the binarization of real-valued template. After that, we will report our method in finding the ”optimal” discriminant functions. Finally, we will provide the complete algorithm for the enrollment and query stages.

2.1 Linear Discriminant Function for Binarization

Assume there are c classes $\Omega_1, \Omega_2 \ldots \Omega_c$ in a database. Each class Ω_i contains p training samples $\mathbf{x}_{ij}$ ($i = 1, 2 \ldots c$,$j = 1, 2 \ldots p$). We want to find a binarization function $f(\cdot)$, such that the between-class variance V_B is maximized and within-class variance V_W is minimized in binary space.

While the straightforward method is to find f using a multi-class optimization method, it does not work because the output is in binary space. Instead, this paper proposes to employ the linear discriminant function, which is a popular way in classification. As Figure 1(a) shows, the linear discriminant function draws a hyperplane to divide the data space into two subspaces and therefore separates the two classes. To find the optimal linear discriminant function, each sample will be labeled with a label “0” or “1” according to its class number.

For multi-class problem, obviously one discriminant function is not sufficient thus multiple discriminant functions are applied (as Figure 1(b) illustrates). Assume n discriminant functions

$$g_s(\mathbf{x}_{ij}) = (\mathbf{w}_s^{\mathbf{T}} \mathbf{x}_{ij}) + t_s \quad (s = 1, 2 \ldots n)$$

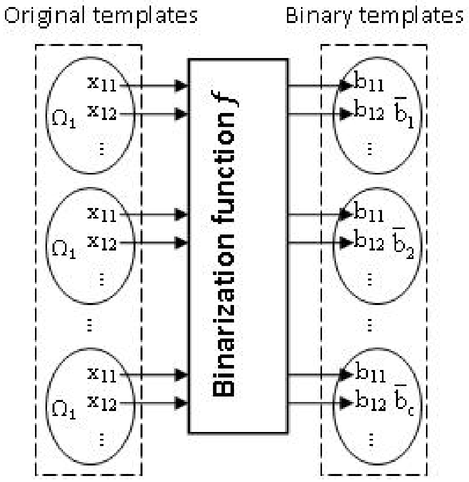


Figure 2. Rationale of the proposed algorithm.

are constructed. These functions are rewritten as the generalized form and the binarization can be written as follows:

$$b_s(\mathbf{x}_{ij}) = \begin{cases} 1 & : \quad \mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij} > 0 \\ 0 & : \quad \mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij} \leq 0 \end{cases} \qquad (s = 1, 2 \ldots n) \quad (1)$$

where $\mathbf{u}_s^{\mathbf{T}} = [\mathbf{w}_s^{\mathbf{T}} : t_s]$ and $\mathbf{y}_{ij}^{\mathbf{T}} = [\mathbf{x}_{ij}^{\mathbf{T}} : 1]$. And

$$\mathbf{b}_{ij} = (b_1(\mathbf{x}_{ij}), b_2(\mathbf{x}_{ij}) \ldots b_n(\mathbf{x}_{ij})) \quad (2)$$

Denote $\mathbf{U}$ as the matrix with columns $\mathbf{u}_s$. Next we need to construct the class centroids $\bar{\mathbf{b}}_i$ ($i = 1, 2 \ldots c$), and then find binarization function f with minimized V_W.

2.2 Gradient Descent Algorithm for Multi-class Binarization

To enhance the discriminability of the extracted binary templates, the reference binary templates should have large distances between each other to gain large V_B, and this paper adopts $[n, k, d]$ BCH codes. Such codes have length n, dimension k and minimum distance d between each other. Then any codewords from the BCH codes can be chosen as class centroid $\bar{\mathbf{b}}_i$ and they have a minimum distance d. With large d, the separation between each class will be large, i.e. large between-class variance V_B.

With the class centroid, we employ the gradient descent algorithm ([16]) which has been used to find the optimal linear discriminant function for two-class problem. Here, we will extend to multiple classes.

In traditional gradient descent algorithm, the perceptron criterion function is constructed as follows

$$J(\mathbf{u}) = \sum_{\mathbf{y}_{ij} \in \Phi} |\mathbf{u}^{\mathbf{T}} \mathbf{y}_{ij}| \quad (3)$$

where Φ is the misclassified set of $\mathbf{y}_{ij}$. While it is proved that the gradient descent algorithm will converge and $J(\mathbf{u})$

will reach 0 if the two classes are well separated, in many practical applications, this may not happen. The iteration will turn over again and again when $J(\mathbf{u})$ is small. The value $J(\mathbf{u})$ and corresponding $\mathbf{u}$ are then selected as the optimal value. In multi-class problem, the criteria function is constructed as follows.

$$J(\mathbf{U}) = \sum_{\mathbf{y}_{ij}\in\Phi} \sum_{\mathbf{u}_s \in \Psi \mathbf{y}_{ij}} |\mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij}| \tag{4}$$

where Φ indicates the misclassified set templates $\mathbf{y}_{ij}$ and $\Psi(\mathbf{y}_{ij})$ indicates the set of $\mathbf{u}_s$ which convert $\mathbf{y}_{ij}$ to bit which is different from its class centroid. Denote q as the number of iterations. Then

$$\mathbf{U}(q+1) = \mathbf{U}(q) - \eta(q) \frac{\partial J(\mathbf{U}(q))}{\partial \mathbf{U}} \tag{5}$$

From Equation (4), we have

$$\frac{\partial J(\mathbf{U})}{\partial \mathbf{U}} = \sum_{\mathbf{y}_{ij}\in\Phi} \frac{\partial \sum_{\mathbf{u}_s \in \Psi(\mathbf{y}_{ij})} |\mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij}|}{\partial \mathbf{U}}. \tag{6}$$

Since

$$\frac{\partial |\mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij}|}{\partial \mathbf{u}_s} = sign(\mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij}) \mathbf{y}_{ij} \tag{7}$$

where $sign(\cdot)$ denotes the sign of the argument, therefore,

$$\frac{\partial \sum_{\mathbf{u}_s \in \Psi(\mathbf{y}_{ij})} |\mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij}|}{\partial \mathbf{U}} = \sum_{\mathbf{u}_s \in \Psi(\mathbf{y}_{ij})} \mathbf{Y}_s. \tag{8}$$

where $\mathbf{Y}_s = [0 : 0 : 0 \ldots 0 : sign(\mathbf{u}_s^{\mathbf{T}} \mathbf{y}_{ij}) \mathbf{y}_{ij} : 0 \ldots 0]$ is the matrix with same dimension as $\mathbf{U}$, all columns except the s^{th} column are zero. Substitute Equation (8) into (6) and substitute (6) into (5), we have

$$\mathbf{U}(q+1) = \mathbf{U}(q) - \eta(q) \sum_{\mathbf{y}_{ij}\in\Phi} \sum_{\mathbf{u}_s \in \Psi(\mathbf{y}_{ij})} \mathbf{Y}_s(q) \tag{9}$$

And $\eta(q)$ is determined experimentally. Therefore, in each iteration, Equation (9) is used to update $\mathbf{U}$ and to find the optimal solution of our binarization process.

2.3 Procedure of the proposed BDA algorithm

This section summarizes the proposed binary discriminant analysis algorithm.

Enrollment:

1. Choose suitable $[n, k, d]$ BCH codes. Randomly choose c codewords from the BCH codes as reference bit templates $\bar{\mathbf{b}}_i$.
2. Randomly initialize $\mathbf{U}(1)$ and convert the original templates $\mathbf{x}_{ij}$ into bit template $\mathbf{b}_{ij}$ with Equation (1) and (2). Set iteration number $q = 1$.

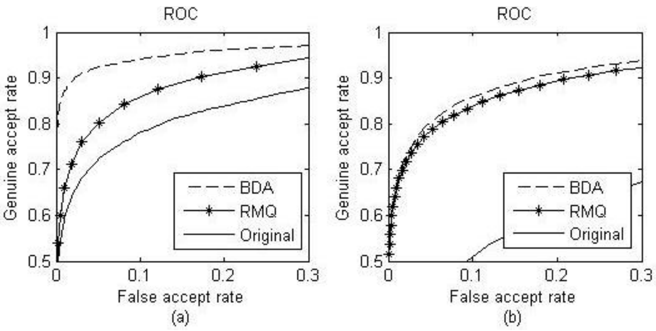


Figure 3. ROC curves for (a) CMU PIE database, (b) FRGC database.

3. While $q < q_{end}$,
   (a) Compute $J(\mathbf{U}(q))$ with Equation (4).
   (b) Do identification with $\mathbf{b}_{ij}$ as query and $\bar{\mathbf{b}}_i$ as reference. Store the misclassified binary templates to set Φ.
   (c) Update $\mathbf{U}(q)$ to $\mathbf{U}(q+1)$ with Equation (9).
   (d) Update $\mathbf{b}_{ij}$ with $\mathbf{U}(q+1)$ and $q = q + 1$.

The total iteration time q_{end} is determined experimentally. We choose the $\mathbf{U}(q)$ with minimum $J(\mathbf{U}(q))$ in the last few steps as the optimal U.

Query:

1. A query template $\mathbf{x}'_{i'j'}$ is presented to the system.
2. Generate a binary template $\mathbf{b}'_{i'j'}$ from $\mathbf{x}'_{i'j'}$ with Equation (1) and (2).
3. Perform identification: compare $\mathbf{b}'_{i'j'}$ with each reference binary template $\bar{\mathbf{b}}_i$, and classify the query to class Ω_{i_0} if $\bar{\mathbf{b}}_{i_0}$ is closet to $\mathbf{b}'_{i'j'}$.

3 Experimental Results

Two popular and public domain databases, namely CMU PIE [17] and FRGC [18], are employed in our experiments. The parameters are shown in Table 1, where c is the number of classes in the database, m denotes the number of images per class and p is the number of images per class used for training.

In all experiments, face region is manually extracted and aligned. Fisherface [2] is used to extract the facial feature vector which is considered as the original face template. We choose codewords from the $[511, 76, 171]$ BCH codes and determine the reference binary templates, which has sufficient between-class variance ($d = 171$) and sufficient security level ($k = 76$). Random Multi-space Quantization

(RMQ) algorithm [9] is used for comparison. The extracted RMQ binary templates have the same length as the original templates. We test our proposed algorithm in both identification and authentication systems.

Table 1. The experiment settings

Database	c	m	p
CMU PIE	68	105	10
FRGC	350	40	5

The experimental results are shown in Figure 3, with symbol “Original” denotes original system without protection, “RMQ” denotes the Random Multi-sacle Quantization algorithm and “BDA” denotes the proposed Binary Discriminant Analysis algorithm. The experimental results show that the binary templates extracted by our proposed BDA algorithm have better discriminability than the original templates and the binary templates extracted from the RMQ algorithm. The genuine accept rate (GAR) with fixed false accept rate FAR=0.1 in ROC curves and rank 1 accuracy of these methods are also reported in Table 2 and 3.

Table 2. GAR (in %) for the BDA algorithm, RMQ algorithm and the original template

GAR(%)	Original	RMQ	BDA
CMU PIE	59.26	66.18	87.32
FRGC	26.28	65.15	67.02

Table 3. Rank 1 accuracy (in %) for the BDA algorithm and RMQ algorithm

Accuracy (%)	RMQ	BDA
CMU PIE	66.93	83.25
FRGC	54.50	57.71

4 Conclusions

In this paper, we have proposed a new binary discriminant analysis (BDA) scheme to convert the original face templates into binary templates. A new multiple class gradient descent algorithm is proposed for optimizing the objective function in binary space. Experimental results show that the proposed BDA algorithm handles the discriminability of the extracted binary templates well and suitable for the biometric cryptosystem approach.

References

[1] J. L. Massey, “Shift-Register Synthesis and BCH Decoding,” *IEEE Trans. Inform. Theory,*, vol. 15, no. 1, pp. 122-127, January 1969.

[2] P N Belhumeur, J P Hespanha and D J Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection”, *IEEE Trans. on PAMI,* 19(7), pp. 711-720, 1997.

[3] N Ratha, J Connell and R Bolle, “Enhancing security and privacy in biometric-based authentication systems,” *IBM Systems Journal,* Vol. 40. No. 3, pp. 614 - 634, 2001.

[4] A K Jain, K Nandakumar and A Nagar, “Biometric template security,” *EURASIP Journal on Advances in Signal Processing*, Vol. 8, 2008.

[5] U Uludag, S Pankanti, S Prabhakar and A K Jain, “Biometric cryptosystems: issues and challenges,” *Proceedings of the IEEE,* vol. 92, no. 6, pp. 948-960, 2004.

[6] A Juels, M Wattenberg, “A fuzzy commitment scheme”, *Proceedings of the Sixth ACM Conf. on Comp. and Comm. Security,* pp. 28-36, 1999.

[7] A Juels and M Sudan. “A Fuzzy Vault Scheme”, *IEEE International Symposium on Information Theory,* 2002.

[8] F Monrose, M Reiter, Q Li and S Wetzel, “Cryptographic Key Generation from Voice,” *Proc. IEEE Symp. Security and Privacy*, pp.202-213, May 2001.

[9] A Teoh, A Goh and D. Ngo, “Random Multispace Quantization as an Analytic Mechanism for BioHashing of Biometric and Random Identity Inputs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1892-1901, Dec. 2006.

[10] Y C Feng, P C Yuen and A K Jain, “A Hybrid Approach for Generating Secure and Discriminating Face Template,” *IEEE Transactions on Information Forensics and Security*, in press, 2010.

[11] E C Chang and S Roy, “Robust extraction of secret bits from minutiae,” *in Proceedings of 2nd International Conference on Biometrics,* pp. 750C759, 2007.

[12] A Nagar, K Nandakumar and A K Jain, “A Hybrid Biometric Cryptosystem for Securing Fingerprint Minutiae Templates,” *Pattern Recognition Letters,* 2009.

[13] A Nagar, K Nandakumar and A K Jain, “Securing fingerprint template: Fuzzy vault with minutiae descriptors,” *International Conference on Pattern Recognition,* pp. 1-4, 2008.

[14] T A M Kevenaar, G J Schrijen, M Veen, A H M Akkermans, “Face recognition with renewable and privacy preserving binary templates,” *IEEE Workshop on Automatic Identification Advanced Technologies,* pp. 21-26, 2005.

[15] J P Linnartz, P Tuyls, “New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates,” *Audio and Video-Based Biometric Person Authentication,* 2003.

[16] R Duda, P Hart and D Stork, “Pattern classification,” 2001.

[17] http://vasc.ri.cmu.edu/idb/html/face/index.html

[18] http://www.frvt.org/FRGC/

Very Low Resolution Face Recognition Problem

Wilman ZOU
wwzou@comp.hkbu.edu.hk

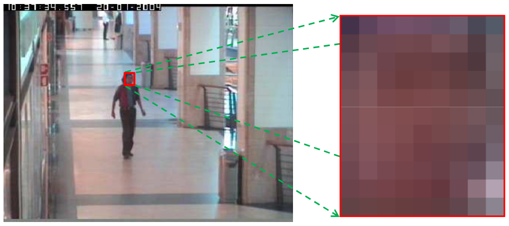
Pong C. Yuen
pcyuen@comp.hkbu.edu.hk

Abstract

This paper addresses the very low resolution (VLR) problem in face recognition in which the resolution of face image to be recognized is lower than 10x10. The VLR problem happens in many surveillance camera-based applications and existing face recognition algorithms are not able to give satisfactory performance on VLR face image. While face super-resolution (SR) methods can be employed to reconstruct a higher resolution image, the existing face SR methods do not perform well on such a low resolution. Unlike the existing SR methods working on VLR input image space, this paper proposes a new data constraint which performs error measurement in high resolution (HR) image space. As HR image contains detailed and discriminative information, the reconstructed HR image gives both better visual quality and recognition performance. Moreover, with the use of new data constraint, discriminative constraint can be easily integrated in the optimization process. CMU-PIE and FRGC face databases are selected for experiments. Experimental results show that the proposed method outperforms existing methods.

1. Introduction

With the growing installation of surveillance cameras in public areas, ranging from a small-scale stand-alone camera applications in banks and supermarkets to large-scale multiple networked-close-circuit television (CCTV) in law enforcement applications in public streets, there is an increasing demand of face recognition technology for surveillance cameras. Wide-angle cameras are normally used and installed in a way that viewing area is maximized. In turn, face region in the scene is normally very small. When the person is not close to the camera, the face region will be less than a hundred of pixels (i.e., smaller than 10x10 pixels) as shown in Figure 1. Recognition of such a very low-resolution face image called very low-resolution (VLR) face recognition problem. While face recognition research has been studying for more than three decades and many promising practical face recognition systems have been developed, it is assumed the face region is large enough and contains sufficient information for recognition [20]. Empirical studies [11] showed that minimum face image resolution between 32x32 and 64x64 is required for existing algorithms. The recognition performance of the existing methods on VLR face image will degrade dramatically. This is because the VLR face image contains very limited information and many image details have been lost in the downsampling process as shown in Figure 1(b). It can be seen that even human is hard to recognize the face image.


(a) surveillance video

(b) face region

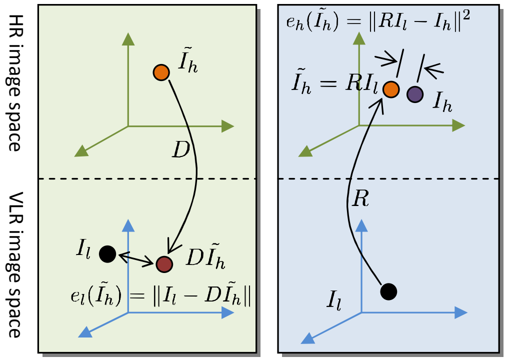
Figure 1. A typical frame from a surveillance video (from CAVIAR database)

Super-resolution (SR) is a method to construct a high-resolution (HR) image from its low-resolution (LR) image [3, 16, 13]. Theoretically, applying SR technique on VLR face image, the reconstructed HR image can be used for face recognition. However, existing face SR algorithms do not give satisfactory results on VLR face image.

Most, if not all, of the existing face SR algorithms are learning-based (please refer to section 2 for details) and formulated as a constraint optimization problem. Generally speaking, two types of constraints namely, data constraint and algorithm specific constraint, are considered. To the best of our knowledge, all existing algorithms employed the same data constraint, which is defined as

$$\|D\tilde{I}_h - I_l\|^2 \qquad (1)$$

where $\tilde{I}_h$ and I_l represent the reconstructed HR face image and the input VLR face image , D is the downsampling matrix. The objective of data constraint is to find a HR image such that the downsampled reconstructed HR image is very close to the input VLR face image. It can be seen that the data constraint measures the error in input low dimen-



(a) existing methods (b) proposed method

Figure 2. Data constraint in existing methods vs. the proposed data constraint

sional image space as illustrated in Figure 2(a). In VLR face recognition problem, the VLR image dimension is too small to reflect the error in high dimensional space. The contribution of the data constraint is not significant and the algorithm specific constraint will dominate the optimization process (please refer section 3 for details). In turn, the reconstructed HR image may not be similar with the input image.

To overcome the VLR problem, this paper proposes a new formulation of the data constraint. Unlike existing methods performing the error measurement in VLR image space, this paper proposes to perform the measure in HR image space as illustrated in Figure 2(b). To achieve this, this paper develops a new learning procedure to determine the relationship R between the VLR space and HR image space. It is noted that the VLR and HR image pairs (I_l, I_h) are available in training stage. Moreover, with the use of the new data constraint, discriminative term can also be easily integrated into the optimization process. In turn, the reconstructed HR image using the proposed method not only has high visual quality, but also has high discriminability for recognition.

The rest of this paper is organized as follows. Section 2 gives a brief review on the existing face super-resolution algorithms. Section 3 discusses the limitation on current data constraint used in existing methods. The proposed method and experimental results are reported in Section 4 and Section 5, respectively. Finally, Section 6 draws the conclusion.

2. Related Work

With the prior information that the image to be super-resolved is a human face, most, if not all, existing face SR algorithms (face hallucination [1]) are learning based. Let I_h and I_l be the HR image and input LR image, respectively. The objective of SR is to determine a high resolution image $\tilde{I}_h$, given the input VLR image and a downsampling matrix D, by solving the following equation,

$$D\tilde{I}_h = I_l \qquad (2)$$

Since D is an N-to-one mapping, solving Eq.2 is ill-posed. To overcome the problem, many learning-based face SR algorithms [1, 2, 4, 5, 6, 7, 9, 10, 17, 19, 12] have been developed. We categorize the algorithms into two approaches namely, example based and maximum a posterior (MAP) based.

In MAP-based approach, different constraints, which are induced from the conditional probability $P(I_h|I_l)$, are used to alleviate the ill-posed problem. Eq.(2) is converted to maximize the conditional probability $P(I_h|I_l)$ and in turn, maximize $P(I_h|I_l) * P(I_h)/P(I_l)$. To do this, the key step is to model $P(I_h)$ so that it can restrict the reconstructed HR image belonging to the HR face image space. Some researchers model $P(I_h)$ by non-parameter density estimation. Baker *et al.* [1] considered the distance between the LR input image patch and the most similar training image patch to estimate $P(I_h)$. Liu *et al.* [9] employed non-parameter Markov network to model the HR residual images which is useful to recover the HR image with good visual quality. Subspace method is also employed to restrict the reconstructed HR image locating inside the face subspace, such as PCA subspace [9] and KPCA subspace [2]. They estimated $P(I_h)$ by minimizing the reconstructed mean squared error.

In example-based approach, the HR image is reconstructed by restricting in the face subspace spanning by the HR examples. Denote $I_l = a_1 I_l^1 + a_2 I_l^2 + \cdots + a_n I_l^n$, where a_i and I_l^i are the weight (coefficient) and LR image example, respectively, i =$1 \cdots n$. The reconstructed HR image I_h can be obtained by replacing the LR examples (I_l^i) by the corresponding HR examples (I_h^i), i.e., I_h= $a_1 I_h^1 + a_2 I_h^2 + \cdots + a_n I_h^n$. This approach assumes that the structure in LR subspace is the same as that in HR subspace. Based on this idea, Wang *et al.* [18] proposed an algorithm using Eigen-transformation while Zhang *et al.* [19] performed on DCT domain. Liu *et al.* [10] and Jia *et al.* [6] conducted in patch tensor space while Park *et al.* [12] conducted on face texture space and face 3D shape space. The assumption, which the structure in LR subspace is the same as that in HR subspace, may not be valid due to the face variance. In turn, the performance may not be satisfactory on face image with variations.

Moreover, a few methods were developed from recognition perspective. Gunturk *et al.* [4] applied MAP-based SR method to reconstruct the eigenface coefficients for face recognition. Hennings-Yeomans *et al.* [5] proposed to perform SR and feature extraction from LR image simultaneously. Wang *et al.* [17] employed example-based approach to reconstructed HR images for face recognition, while Li *et al.* [7] reconstructed the image features, instead of HR images, for face recognition.

3. Limitation on Current Data Constraint

From the review in Section 2, existing super-resolution methods can be generalized and formulated as 2-constraint optimization problem,

$$\tilde{I_h} = \arg\min_{I_h} \varphi_D(I_h) + \varphi_S(I_h) \tag{3}$$

where $\varphi_D(.)$ is the data constraint and $\varphi_S(.)$ is the algorithm specific constraint. Both MAP-based and example-based approaches use the same data constraint

$$\varphi_D(I_h) = \|D\tilde{I_h} - I_l\|^2 \tag{4}$$

MAP-based approach employs $\|D\tilde{I_h} - I_l\|^2$ which is induced from the condition probability $P(I_l|I_h)$. Example-based methods use the data constraint implicitly. It minimizes the difference defined as follows

$$\{\tilde{a_i}\} = \arg\min_{\{a_i\}} \|I_l - \sum a_i I_l^i\|^2 \tag{5}$$

to get the weights for HR examples to reconstruct the HR image. Considering $DI_h^i = I_l^i$, this difference is the data constraint in Eq.(1).

Let the solution space (set) of equation e is $\mathbf{U}(e)$. It can be shown that the solution of Eq.(3), $\tilde{I_h}$, locates in the intersection of the two constraints' solution space, as follows:

$$\{\tilde{I_h}\} = \mathbf{U}(\varphi_D \leq c_1) \cap \mathbf{U}(\varphi_S \leq c_2) \tag{6}$$

where c_1 and c_2 are two positive error terms (constants) to control the dimension (size) of the solution space. The smaller the term is, the smaller the dimension of the solution space.

Under the VLR face recognition problem, even though c_1 is set to 0, the solution space $\mathbf{U}(\varphi_D = 0)$ is still very large. This can be explained by considering the following example. Suppose the resolution of the input image is 8 x 8 and the target HR image with resolution of 64 x 64. From linear algebra, the dimension of the solution space of $\mathbf{U}(\varphi_D = 0)$ is not less than 4032 (= 64x64 - 8x8), while the dimension of the target HR image space is 4096. In another word, $\mathbf{U}(\varphi_D = 0)$ occupies 98.44% of that in HR image space. It means that data constraint cannot effectively restrict the target HR image solution space. Therefore, in determining the high resolution image, the algorithm specific constraint $\mathbf{U}(\varphi_s \leq c_2)$ will dominate during the optimization process. In turn, there is a possibility that reconstructed HR image may have serious artifacts and/or not look like the original person.

4. Proposed Relationship Learning based Super-resolution

To overcome the limitation on current data constraint, this paper proposes a new formulation of the data constraint which measures the error in HR image space. As illustrated in Figure 2(b), the basic idea is to determine the relationship, R, between the VLR image space and HR image space, which is represented in the form of matrix. Unlike the current data constraint shown in Figure 2(a), the proposed new data constraint minimizes the error in HR image space. Since HR image space contains more useful and detailed image information, the reconstructed image will have a higher visual quality and contains more discriminative information. The detailed method in determining the relationship R will be given in Section 4.1. Another advantage of the proposed method is that other constraint(s) can be easily integrated into the new data constraint while determining R. To demonstrate this superior property, this paper adapts a simple discriminative constraint and integrates into the new data constraint. Details will be given in Section 4.2.

4.1. Relationship Learning (RL)

Under the VLR problem, the input VLR image space contains very little useful information, so current data constraint does not estimate the reconstruction error well. A reasonable method is to estimate such error in HR image space. But the challenge is that given a VLR query image, the corresponding HR image is not available. Instead of learning the HR image directly, we propose a new SR framework to learn the relationship between HR image space and LR image space. After determining the relationship R, HR image can be reconstructed by R.

Given a set of training HR and VLR image pairs ($\{I_h^i, I_l^i\}_{i=1}^N$), and let R be the relationship between the HR image space and the VLR image space. The HR image can be reconstructed from its VLR image and R, by $\tilde{I_h} = RI_l$. So the reconstructed error in HR image space, $e_h(\tilde{I_h})$, is given by,

$$e_h(\tilde{I_h}) = \|I_h - RI_l\|^2 \tag{7}$$

R can then be determined in training stage by minimizing this error as follows

$$R = \arg\min_{R'} \sum_{i=1}^{N} \|I_h^i - R' I_l^i\|^2 \tag{8}$$

It can be shown that R is unique if the number of training image pairs N is larger than the data dimension of VLR image space d_L. $N > d_L$ is true in many VLR face recognition applications.

In query stage, given a query VLR image I_l, the corresponding HR image is recovered as follows,

$$\tilde{I_h} = RI_l \tag{9}$$

In this paper, this method is called relationship learning (RL) super-resolution.

4.1.1 Error Analysis

Considering that a HR image I_h consists of two components namely, the low frequent image component l and high frequent image details h; and they satisfy $Dh = 0$ and $Dl = l$. The HR image can then be represented as

$$I_h = l + h \tag{10}$$

The reconstruction error can also be separated into two parts namely, low frequent image component error Δl and high frequent details error Δh. So the reconstructed HR $\tilde{I_h}$ is given by,

$$\tilde{I_h} = I_h + \Delta l + \Delta h \tag{11}$$

For the current data constraint used in existing methods, the error is formulated as

$$\begin{aligned} e_l(\tilde{I_h}) &= \|D\tilde{I_h} - I_l\|^2 = \|D(I_h + \Delta l + \Delta h) - I_l\|^2 \\ &= \|\Delta l\|^2 \end{aligned} \tag{12}$$

so $e_l(\tilde{I_h})$ reflects the error introduced by low frequent image component only. For our proposed method, we have

$$\begin{aligned} e_h(\tilde{I_h}) &= \|\tilde{I_h} - I_h\|^2 = \|I_h + \Delta l + \Delta h - I_h\|^2 \\ &= \|\Delta l + \Delta h\|^2 \end{aligned} \tag{13}$$

That means our proposed method can estimate the reconstruction error from both low frequent image component l and the high frequent details h.

In turn, both Δh and Δl will be minimized when $e_h(\tilde{I_h})$ is minimized, while minimizing $e_l(\tilde{I_h})$ only leads Δl is minimized.

From the discussion in Section 3, in VLR face recognition problem, it can be shown that the reconstructed HR image $\tilde{I_h}$ easily satisfies

$$\|\Delta l\|^2 = 0 \tag{14}$$

due to the large solution space of $\mathbf{U}(\|D\tilde{I_h} - I_l\|^2 = 0)$. This implies that the major reconstructed error is caused by $\|\Delta h\|^2$. To get better HR image quality, error caused by both Δl and Δh should be minimized. Our proposed data constraint can properly estimate the reconstructed error. In turn, the proposed SR algorithm could make use of more useful information in the HR image space, and recover the high frequent details of face image better.

4.2. Discriminative Constraint

It can be seen that R restricts the reconstructed HR images locating in an optimal subspace for minimizing the reconstruction error $e_h(\tilde{I_h})$. This inspires us to find an optimal subspace induced by R with other additional constraint(s). In order to further boost the discriminability of the reconstructed HR image, discriminative constraint is added to the relationship learning based SR in determining the ”optimal” R. A discriminative super-resolution (DSR) algorithm is proposed.

A natural step is to make use of the class information of the training data. From recognition perspective, we expect the reconstructed HR images should be clustered with the images from the same class, and far away from the images from other classes. Therefore, based on maximum margin criterion (MMC) [8], we design a discriminative constraint as follows:

$$\begin{aligned} d(R) = &\frac{1}{N_1} \sum_{\Omega(I_h^i)=\Omega(I_l^j)} \|I_h^i - RI_l^j\|^2 \\ &- \frac{1}{N_2} \sum_{\Omega(I_h^i)\neq\Omega(I_l^j)} \|I_h^i - RI_l^j\|^2 \end{aligned} \tag{15}$$

where N_1, N_2 are normalization constants, and $\Omega(u)$ is the class label of u. Integrating Eq.(15) with Eq.(8), the new discriminative super resolution formula can be written as :

$$\dot{R} = \arg\min_{R'} \frac{1}{N} \sum_{i=1}^{N} \|I_h^i - R' I_l^i\|^2 + d(R') \tag{16}$$

And the HR image can be reconstructed by $\dot{R}$.

This subspace induced by $\dot{R}$ is an optimal subspace for recognition with respect to MMC. That means the HR images reconstructed by $\dot{R}$ locates in a subspace where they can be linear separable. Therefore, the HR image reconstructed by $\dot{R}$ will contain more discriminability and better for recognition purpose.

5. Experiments and Analysis

Two experimental results are reported in this section. In the first experiment, we would like to evaluate the reconstructed HR image quality using the proposed method. Both objective measurement in terms of mean squared error (MSE) and subjective human visual quality are used. Comparison with existing face SR methods are also reported. The second experiment is to evaluate the reconstructed HR image discriminability using the proposed method. Three popular face recognition methods namely eigenface, kernel PCA and SVM, are selected as recognition engine. Recognition results on HR images reconstructed using existing SR methods are also reported.

Two public databases namely, CMU PIE [15] and FRGC 2.0 [14] are used for experiments. For CMU PIE database, a subset of 68 classes with 21 different illuminations is considered. For FRGC database, the subset of 311 classes with 10 images per class is used. In all experiments, the resolution of VLR image is 7x6 while the resolution of HR image is 56x48. The magnification is 8, which is challenging in SR. To determine the relationship matrix R and train the recognition engine, 13 and 8, HR and VLR image pairs from each person are randomly selected from CMU PIE and FRGC databases for training, respectively, while the rest are used for testing.

Three existing face SR methods namely Hallucination Face (HF) method [1], Eigentransfromation based Face SR (EF) method [18] and KPCA-based Face SR (KF) method [2] are selected for comparison. The results of BiCubic interpolation (BC) are also given for benchmarking.

5.1. Experiment 1: Image Quality

Figure 3 shows some of the reconstructed images using proposed method and existing methods. Figures 3(a) and (g) show the input 7x6 query image and original 56x48 HR image. Figures 3 (b) - (e) display the results using bicubic interpolation method, HF [1] method, EF [18] method and KF [2] method. It can be seen that both bicubic interpolation (BC) method and KF method give a relatively blur image and high frequency details cannot be recovered. Both HF method and EF method could recover some high frequent details. However, HF method generates some artifacts which degrade the human visual quality. The visual quality of reconstructed HR images from EF method are good. However, when comparing with the original HR image, the reconstructed HR image does not look like the original HR image. Figure 3(f) shows the results using our proposed method. It can be seen that the proposed method gives a good visual quality image which also look like the original one.

To further evaluate the SR algorithms in terms of the visual quality, the zoom-in results are given in Figure 4. For each image shown in Figure 4 (a), the first row shows the zoom-in result while the second row shows the downsampled VLR images. It can be seen that both HF method and EF method introduce server artifacts. At the same time, if we look at the VLR images, the artifacts can not be reflected at VLR images. This supports the analysis in Section 4.1.1. The results using our proposed method are shown in Figure 4(e). It can be seen that the results are good.

The mean squared error (MSE) of the proposed method and the existing methods on two databases are also recorded and shown in Table 1. The error reported in the table is the average of the all testing images. It can be seen that for both databases, the proposed method gives the smallest MSE.

5.2. Experiment 2: Image Discriminability

In this experiment, we would like to evaluate the performance of the discriminative super resolution (DSR) algorithm in terms of the recognition result. Recognition experiments are performed on (i) input VLR query images, (ii) original HR images, reconstructed HR images from (iii) HF [1], (iv) EF [18], (v) KF [2] and (vi) the proposed DSR method. Three face recognition engines namely eigenface, KPCA and SVM, are used for experiments on CMU PIE and FRGC databases. The results are recorded in Table 2. Experimental results show that:

- There is a significant drop of recognition accuracy (as high as 30%) for VLR image, comparing with the original HR image, for all recognition engines on both databases.
- The proposed method outperforms existing SR methods. It implies that the reconstructed HR image using the proposed method has high discriminability for recognition purpose.

The CMC curves are also plotted from Figures 5 - 7 for CMU PIE database and Figure 8-10 for FRGC database.

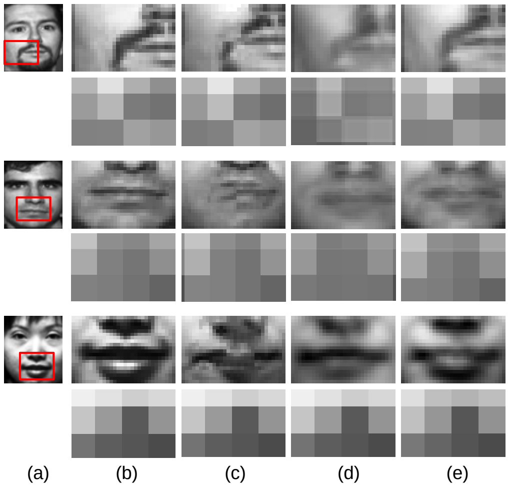


Figure 4. The zoom-in results of different SR algorithms. (a) and (b) are original HR images, (c) Hallucination Face method (HF)[1], (d) Eigentransfromation based Face SR method (EF) [18], (e) proposed RL method.

6. Conclusion

The very low resolution face recognition problem is defined and discussed in this paper. To solve the problem, a new super-resolution method has been developed. A new data constraint for super-resolution has been designed and reported. The proposed new data constraint offers at least two advantages. First, the error is measured in high resolution image space so that better high resolution image quality image can be obtained. Second, with the use of new data constraint, discriminative constraint can be easily integrated. Moreover, based on the new data constraint, a new learning based super-resolution approach which learns the relationship between low resolution image space and high resolution image space, is also proposed.

Database	BC	HF [1]	EF [18]	KF [2]	Proposed RL
CMU PIE	424.4	475.6	291.9	1143.1	179.9
FRGC 2.0	1259.4	1838.8	1510.1	1707.6	870.5

Table 1. The MSE of different SR methods (Average of all testing images)

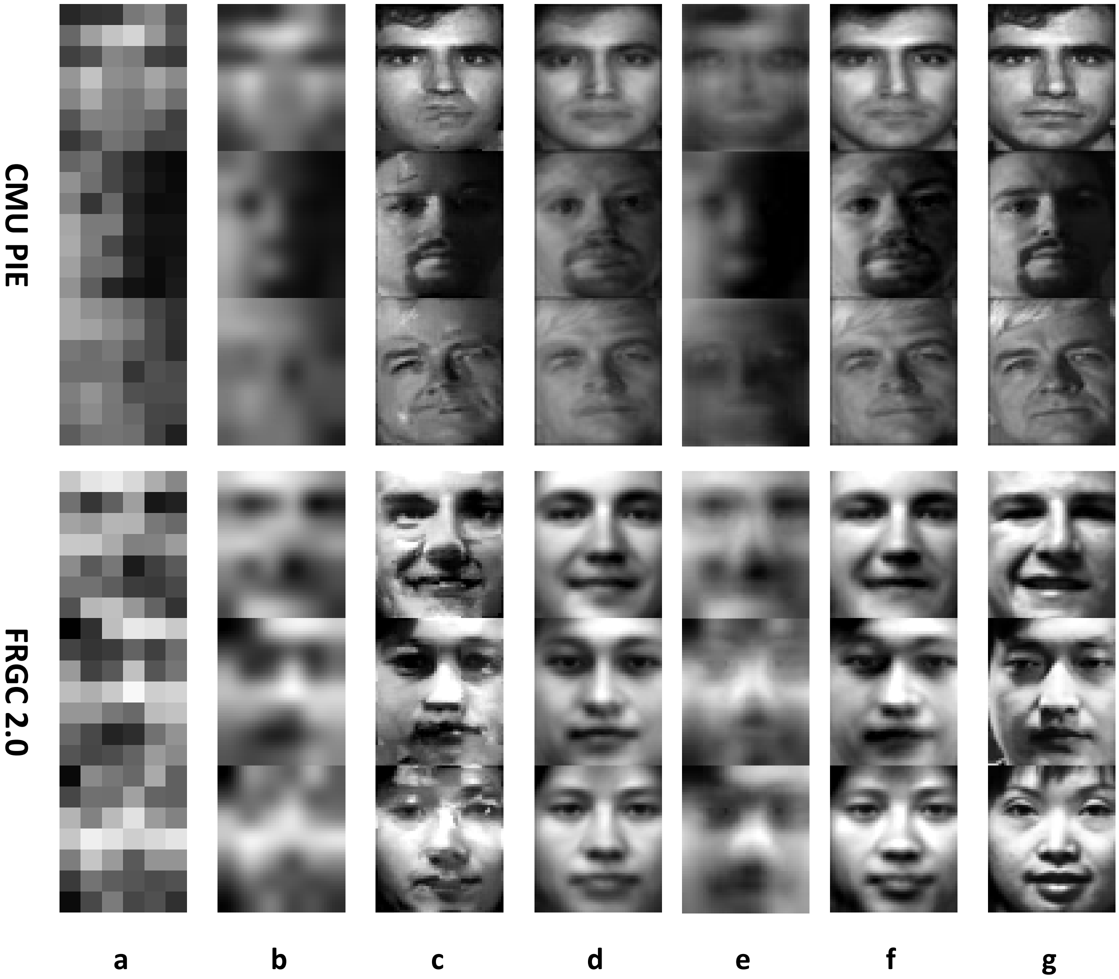


Figure 3. SR results: (a)input VLR images (7 x 6), (b) SR results by Bicubic interpolation,(c) by Hallucination Face method (HF)[1], (d) Eigentransfromation based Face SR method (EF) [18], (e) KPCA-based Face SR method (KF) [2], (f) Our proposed method (RL), (g) original HR images. The resolution of reconstructed HR images is 56 x 48.

CMU PIE and FRGC databases are selected for experiments and the results show that the proposed method not only gives better visual image quality, but also smaller mean squared error, comparing with existing face super-resolution methods. Three face recognition algorithms are also selected to evaluate the discriminability of the reconstructed high resolution images. Experimental results show that the reconstructed images using the proposed method outperforms those reconstructed from existing face super-resolution methods. This illustrates that the reconstructed high resolution has high discriminability as well.

References

[1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002. 2, 5, 6, 8

[2] A. Chakrabarti, A. N. Rajagopalan, and R. Chellappa. Super-resolution of face images using kernel pca-based prior. *IEEE Transactions on Multimedia*, 9(4):888–892, 2007. 2, 5, 6, 8

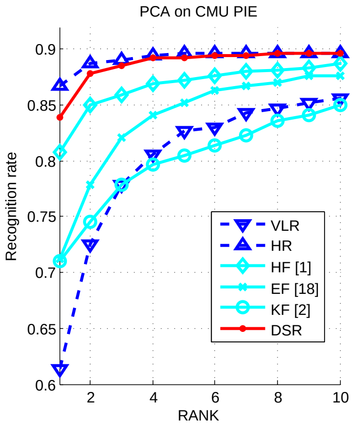


Figure 5. Eigenface on CMU PIE

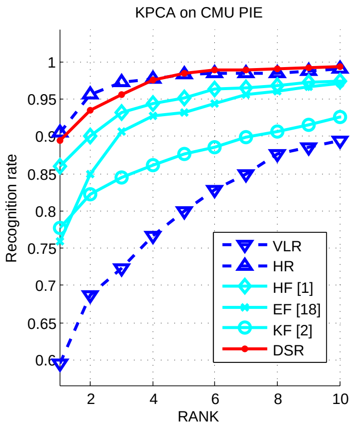


Figure 6. Kernel PCA on CMU PIE

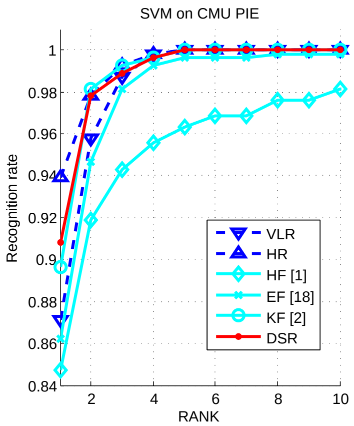


Figure 7. SVM on CMU PIE

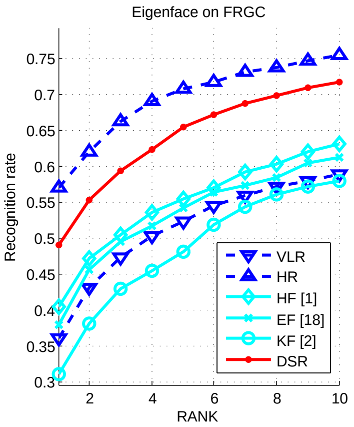


Figure 8. Eigenface on FRGC

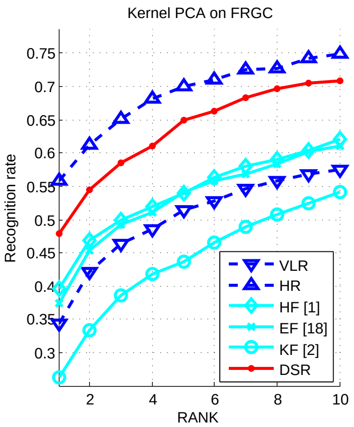


Figure 9. Kernel PCA on FRGC

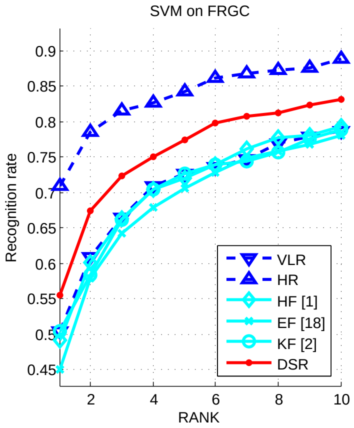


Figure 10. SVM on FRGC

[3] G. Cristobal, E. Gil, F. Sroubek, J. Flusser, C. Miravet, and F. B. Rodriguez. Superresolution imaging: a survey of current techniques. *Advanced Signal Processing Algorithms, Architectures, and Implementations XVIII*, 7074:70740C–70740C18, 2008. 1

[4] B. Gunturk, A. Batur, Y. Altunbasak, M. Hayes, and R. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE Transactions on Image Processing*, 12(5):597–606, 2003. 2

[5] P. H. Hennings-Yeomans, S. Baker, and B. Kumar. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[6] K. Jia and S. Gong. Generalized face super-resolution. *IEEE Transactions on Image Processing*, 17(6):873–886, 2008. 2

[7] B. Li, H. Chang, S. Shan, X. Chen, and W. Gao. Hallucinating facial images and features. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 1–4, 2008. 2

[8] H. Li, T. Jiang, and K. Zhang. Efficient and robust feature extraction by maximum margin criterion. *IEEE Transactions on Neural Networks*, 17(1):157–165, 2006. 4

[9] C. Liu, H. Y. Shum, and W. T. Freeman. Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75(1):115–134, 2007. 2

[10] W. Liu, D. Lin, and X. Tang. Hallucinating faces: Tensorpatch super-resolution and coupled residue compensation. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 478 – 484, 2005. 2

[11] Y. M. Lui, D. Bolme, B. A. Draper, J. R. Beveridge, G. Givens, and P. J. Phillips. A meta-analysis of face recognition covariates. In *Proceedings of International Conference on Biometrics: Theory, Applications and Systems(BTAS)*, 2009. 1

[12] J. S. Park and S. W. Lee. An example-based face hallucination method for single-frame, low-resolution facial images. *IEEE Transactions on Image Processing*, 17(10):1806–1816, 2008. 2

[13] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003. 1

Database	face recognition algorithm	very low resolution image	original high resolution image	HF [1]	EF [18]	KF [2]	Proposed DSR
CMU PIE	Eigenface	61.3	86.7	80.8	71.3	71.7	**83.9**
	Kernel PCA	59.5	90.4	86.0	75.9	77.8	**89.5**
	SVM	87.1	93.9	84.7	86.2	89.6	**90.8**
FRGC 2.0	Eigenface	36.0	57.1	40.5	37.9	31.1	**49.0**
	Kernel PCA	34.4	55.8	39.6	37.5	26.4	**47.9**
	SVM	50.4	70.9	49.2	45.0	50.4	**55.5**

Table 2. Rank 1 recognition rate (%).

[14] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 947–954, 2005. 4

[15] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003. 4

[16] J. van Ouwerkerk. Image super-resolution survey. *Image and Vision Computing*, 24(10):1039 – 1052, 2006. 1

[17] X. Wang and X. Tang. Face hallucination and recognition. *LNCS: Advances in Neural Networks*, pages 486–494, 2003. 2

[18] X. Wang and X. Tang. Hallucinating face by eigentransformation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(3):425–434, 2005. 2, 5, 6, 8

[19] W. Zhang and W. K. Cham. Learning-based face hallucination in dct domain. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[20] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458, 2003. 1

Display results of opinion mining with tagclouds

Luole Qi

Abstract

Nowadays, many big commercial websites such as Yahoo shopping, amazon.com often ask their customers to write and post their reviews and hands-on experiences on products they have purchased. Unfortunately, reading through all customer reviews is time-consuming, especially for popular products, the number of reviews can be up to hundreds or even thousands. This makes it difficult for a potential customer to read them to make an decision. The system designed in this work aims to mine customer reviews of a product and extract product features on which reviewers express their opinions and use an effective way called Tagclouds to present them. This summarization task is different from traditional text summarization because we are only interested in the specific features of the product that customers have opinions on and also whether the opinions are positive or negative. We do not summarize the reviews by selecting or rewriting a subset of the original sentences from the reviews to capture their main points as in the classic text summarization. In this paper, a number of techniques are presented to mine such features and we also give the method to generate tagclouds.

1. Introduction

With the rapid expansion of e-commerce, more and more products are sold on the Web, and more and more people are buying products on the Web. In order to enhance customer satisfaction and their shopping experiences, it has become a common practice for online merchants to enable their customers to write and post reviews or to express opinions on the products that they buy. With more and more common users becoming comfortable with the Internet, an increasing number of people are writing reviews. As a consequence, the number of reviews that a product receives grows rapidly. Some popular products can get hundreds of reviews at some large merchant sites. This makes it very hard for a potential customer to read them to help him or her to make a decision on whether to buy the product. This paper aims to design a system that is capable of extracting, product features and opinion expressions from product reviews and present the result to users in tagclouds form.

The objective in our system is to answer the following questions: given a particular product, 1) how to extract potential product features from the reviews? 2) how to group the synonymous product features together? and 3) How to design the font size and weight of tags in tagclouds.

The rest of this paper is organized as follows: section 2 discusses related work. Section 3 describes in detail the system framework and each system component. We describe in section 4 the design of font size and weight of features and opinion words in tagcloud and present the result. In the end, we give our conclusions and our future work in section 5.

2. Related Work

.

Opinion mining has been studied by many researchers in recent years. The research maybe divided into two main research categories, say, document level opinion mining and feature level opinion mining. In document level, Turney et. al [1] proposed a method to determinw document's polarity by calculating the average semantic orientation of extracted phrases. So was computed by using pointwise mutual information (PMI) to measure the dependence between extracted phrases and the reference words “excellent” and “poor” by using web search hit counts. Littman ea. al [2] further expanded Turney et. al's work by using cosine distance in latent semantic analysis (LSA) as the distance measure. Dave, Lawrence and Pennock [3] classified reviews on Amazon by calculating scores using normalized term frequency on uni-gram, bi-gram and tri-gram with different smoothing techniques. Das and Chen [6] studied document level sentiment polarity classification on financial documents. Pang, Lee and Vaithyanathan [4] used several machine learning approaches to classify movie reviews and in [5], they further studied another machine learning approach based on subjectivity detection and minimum cuts in graphs for sentiment classification of movie reviews. Our work is different from these as their goal is to determine the sentiment of documents while ours is to perform extraction and classification on features. Another difference is they were not focused on features being commented on.

In feature level opinion mining, Hu and Liu [7] proposed a statistical approach capturing high frequency feature words by using association rules. Infrequent feature words are captured by extracting known opinion words'adjacent noun phrases. A summary is generated by using high frequency feature words (the top ranked features) and ignoring infrequent features. Zhuang, Jing and Zhu [8] classified and summarized movie reviews by extracting high frequency feature keywords and high frequency opinion keywords. Feature-opinion pairs were identified by using a dependency grammar graph. However, it used a fixed list of keywords to recognize high frequency feature words, and thus the system capability is limited. Popescu and Etzioni [9] proposed a relaxation labeling approach to find the semantic orientation of words. However, their approach only extracted feature words with frequency greater than an experimentally set threshold value and ignored low frequency feature words. Ding, Liu and Yu [10] further improved Hu's system by adding some rules to handle different kinds of sentence structures. However all their work does not group synonymous features and do not present them to users in an effective way.

3. The Proposed Method

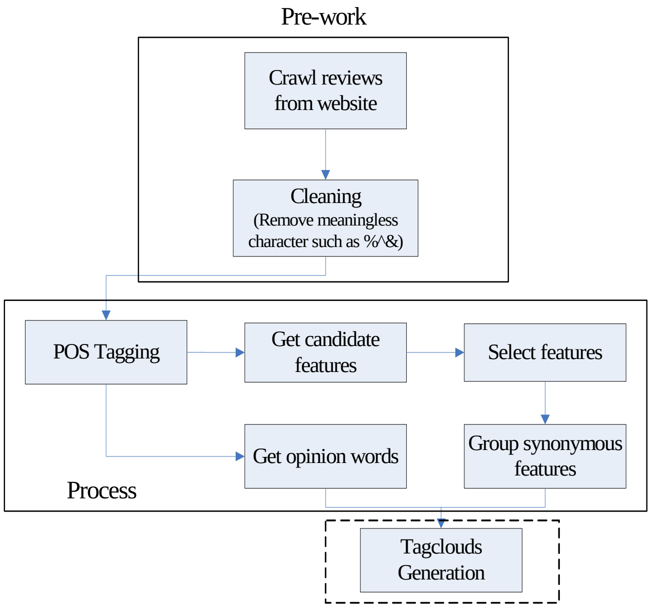

Pre-work
Crawl reviews from website
Cleaning (Remove meaningless character such as %^&)
POS Tagging
Get candidate features
Select features
Get opinion words
Group synonymous features
Process
Tagclouds Generation

Figure 1: A general schedule for the Symposium

Figure 1 gives an architectural overview for our opinion mining system. The system performs the mining in three main steps: features selection, opinion words extraction and tagclouds generation. The inputs to the system are all semi-structured reviews for all the reviews of each product. The output is the tagclouds webpage of the reviews. Given the inputs, the system first downloads (or crawls) all the reviews from Yahoo shopping website by using the API it provide to developers, and save them as XML format files in the local server. Some cleaning work such as deleting some meaningless characters will be done as well. The feature extraction function, which plays a crucial role in our work, first extracts noun words from cons and pros parts of the reviews as candidate product features. And then, we will group synonymous candidate product features together and use a typical word to represent them. After that the opinion words will be extracted by using the POS tags and the semantic polarity of the opinion words will be determined through a dictionary manually made by Pang, B. and Lee, L [11]. In the end, all the results will be displayed in a tagclouds format. In Figure 1, POS tagging is the part-of-speech tagging from natural language processing. Below, we discuss each of the functions in feature extraction in turn.

3.1. Semi-structure of review

All the reviews in our work are from Yahoo shopping website. We got these data using the Yahoo shopping API and save them as static XML file in the local server. The review format is semi-structured and it consists of several parts:

Table 1: Each part of one complete review

Review	Contains each individual review.
Title	The title of the review as entered by the reviewer.
Reviewer	The name or Yahoo! ID that the reviewer chose to call them self.
CreateTime	The UNIX time when the review was written
HelpfulRecommendations	The number of people who found this review helpful
TotalRecommendations	The total number of people who have read this review.
Ratings	Container for sub-ratings / detailed components of the user review.
Rating	Sub-rating for detailed component of the user reivew. Has an attribute ratingType, which specifies the name of the sub-rating component
OverallRatings	The rating given by the reviewer for the product, out of 5.
Pro	The pros (positive attributes) of the product as per the reviewer.
Con	The cons (negative attributes) of the product as per the reviewer.
Posting	The body of the review.

3.2. Part-of-speech Tagging

As many researchers observed, product features are usually nouns or noun phrases in review sentences. Thus the part-of-speech tagging is crucial. The task of POS tagging is the process of marking up the words in a text (corpus) as corresponding to a particular part-of-speech, such as noun and verb. We use the LBJPOS tool by natural language processing group of University of Illinois, Urbana-Champaign to produce the part-of-speech tag for each word in every review, the following shows the example of POS tags for one sentence from our reviews:

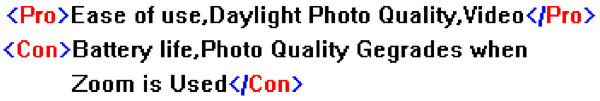
(PRP I) (VBD used) (NNP Olympus) (IN before) (, ,) (VBG comparing) (TO to) (NN canon) (IN in) (JJ general) (, ,) (PRP it) (VBD was) (DT a) (NN toy) (, ,) (NNP S3) (VBZ IS) (VBZ is) (RB not) (DT a) (JJ professional) (NN camera) (, ,) (CC but) (RB almost) (VBZ has) (NN everything) (PRP you) (VBP need) (, ,) (TO to) (PRP me) (, ,) (PRP it) (: ;) (VBZ s) (JJ professional) (, ,) (NN 6mb) (VBZ is) (JJ great) (, ,) (PRP I) (VB don) (: ;) (NN t) (VBP need) (DT a) (NN 10mb) (, ,) (VB zoom) (VBZ is) (JJ outstanding) (, ,) (NN night) (NNS snapshots) (VBP are) (RB really) (JJ good) (. .)

Each sentence is saved in different folders of the local server according to pros part, cons part and post part. Then some cleaning work is also performed which mainly focus on the misspelling checking.

3.3. Product feature extraction

3.3.1 Extraction from pros and cons

We firstly extract product features from pros part and cons part of each review. The reason why we put our focus on these two parts is that they are usually simple sentences or just some words segments sometimes. In addition, they are kind of like the summary of posting made by the users themselves, thus few non-feature noun words would be mentioned in them. In other words, most of the noun words in pros part or cons part tend to be product features based our observation. So, we find all the single noun and noun phase which consist of some consecutive noun words as the selected features for the product. Figure 2 shows the pro and con part of digital camera Nikon D90 of one user.



<Pro>Ease of use,Daylight Photo Quality,Video</Pro>
<Con>Battery life,Photo Quality Gegrades when Zoom is Used</Con>

Figure 2: A general schedule for the Symposium

However, due to the difficulty of natural language understanding, some types of sentences are hard to deal with, even they are simple sentence or just some words segments. Let us see a sentence from the pro part of one review for Nikon D90:

Picture quality is superb.

In this sentence, we could easily find the "Picture quality" is the feature of this product, and the adjective word "superb" describe how good the camera's picture quality is. The feature is mentioned in this sentence, so we know this is a product feature. This kind of feature is defined as explicit feature by many other researchers. But, in many sentences, no feature words would be mentioned let us see this example:

It's not good in low light.

The user is also talking about the picture quality of this camera in a dark environment, however, the product feature words "picture quality" does not appear in this sentence. This kind of feature is defined as implicit feature. In our work, we only focus on explicit features.

3.3.2 Extraction from posting

Different from pros and cons part, posting part usually contains some complicated sentences, which include many unrelated noun words. Thus, just extracting noun word or noun phrase would cause a lot of errors. For example, in this sentence that also is from one review of Nikon D90:

I bought the D90 over half a year ago and have been happy with my purchase.

In this sentence, D90 and purchase are both noun words, however, actually, neither of the two words is product feature. To solve this problem, we adopt a way to prune those un-feature words. Based on our observation, a product feature word is usually going with an adjective word. We define the step of any two consecutive words is 1, and then we find out all the noun words and noun phrases as candidate product features. For each candidate product feature, if there is an adjective word within 3 steps from the candidate, this word or phrase would be selected as product feature. Of course, if there is a colon, stop or any other punctuation within 3 steps, it can not be seen as a feature.

3.4. Grouping synonymous features

In this step, we are trying to find out all the synonymous features and represent them in a typical word to make our summary work more precise, because people usually use different words to describe the same feature, for example, "photo", "picture" and "image" all refers to the same feature in digital camera reviews. Thus, it is important to group features with similar meaning together. We use a simple method. The basic idea is to employ WordNet [12] to check if any synonym groups/sets exist among the features. For a given word, it may have more than one sense, i.e., different synonyms for different senses. However, we cannot use all the synonyms as they will result in many errors. For example, movie and picture are considered as synonyms in a sense, or in a synset (defined in WordNet). This is true when we talk about Hollywood movies. But, in the case of a digital camera review, it is not suitable to regard picture and movie in one synset, as picture is more related to photo while movie refers to video. To reduce the occurrence of such situations, we choose only the top two frequent senses of a word for finding its synonyms. That is, word A and word B will be regarded as synonyms only if there is a synset containing A and B that appear in the top two senses of both words.

3.5. Opinion Words Extraction

For those feature words extracted from pros and cons part, we already know their opinions expressed by users, either positive or negative. However, only knowing the sentiment of them can not satisfy many users, they usually want to see the detail descriptions, that is, how other user think about this product or what words they use to express their opinion. Thus we are trying to find out those words to present them to users in this step. For each feature in pros or cons part, we looking forward and back from the position it's in the context. All the adjective words within 3 step are it's opinion words, and also, if there is a colon, stop or any other punctuations within 3 steps which is more near the feature word, the adjective words can not be seen as an opinion word. For example,

The image is good, Nice high ISO.

In this sentence, "good" and "nice" are both adjective words within 3 steps from the position of product feature "image" in the context, but only "good" describe the "image", and "nice" does not.

We adopt the similar way for the features words which extracted from posting part, the only difference is that we need do determine the polarity of their opinion words. Here for each of them we decide their sentiment using a lexicon tagging more than 8000 adjective words with the key words positive or negative. (This lexicon is made by made by Pang, B. and Lee, L).

4. Tag clouds

After the steps described above, we find out the product features and their opinion words, that is, the opinion-feature pair. The traditional way to present them to users is simply listing them. However, this way can not catch the eyes of user at the first glance and it also can not express the information directly. To make users get the information in a straightforward way, we design Tag clouds to show the result. A tag cloud or word cloud (or weighted list in visual design) is a visual depiction of user-generated tags, or simply the word content of a site, typically used to describe the content of web sites. Tags are usually single words and are normally listed alphabetically, and the importance of a tag is shown with font size or color.[13]

4.1. Font size and weight of tags

In this step, we are trying to show the importance of tags by defining their font size and weight based on the times of one tag appears in a review and ratings given by the reviewer. Firstly, we will define some variables:

Table 2: The variables we defined

Variable names	Notation
HelpfulRecommendations	H_{ij}
TotalRecommendations	T_{ij}
Rating	R_{ijk}
OverallRatings	O_{ij}
Feature	F_{ijk}
Frequency	f_{ijk}
Importance	I_{ik}

In table 2, H_{ij} means the number of people who found the jth review of ith product helpful. T_{ij} means the total number of people who have read the jth review of ith product. R_{ijk} means sub-rating for the kth feature in jth review of ith product. O_{ij} means The rating given by the reviewer for the ith product in the jth review. F_{ijk} means the kth feature which appears in the jth review of the ith product. f_{ijk} means the appearing times of the kth feature in the jth review of the ith product. I_{ik} means the number of users who mention the kth feature of the ith product .

Thus, the impact of the rating value for the kth feature of the ith product (assume there are N_i reviews for the ith product):

$$\frac{R_{ijk}}{\frac{\sum_{j=1}^{N_i} R_{ijk}}{N_i}} = \frac{N_i R_{ijk}}{\sum_{j=1}^{N_i} R_{ijk}}$$

The impact of the overall rating value of any feature in the the jth review of the ith product:

$$\frac{O_{ij}}{\frac{\sum_{j=1}^{N_i} O_{ij}}{N_i}} = \frac{N_i O_{ij}}{\sum_{j=1}^{N_i} O_{ij}}$$

The impact of the number of users who mention the kth feature of the ith product:

$$1+\frac{I_{ik}}{N_i}$$

The impact of the frequency of the kth feature in the jth review of the ith product:

$$\ln(f_{ijk})$$

So the impact of the frequency of the kth feature in the ith product could be defined as (adding the impact of the HelpfulRecommendations and TotalRecommendations):

$$\prod_{j=1}^{N_i}[(\ln(f_{ijk})\cdot(1+\frac{H_{ij}}{T_{ij}})]$$

The font size of the kth feature of the ith product:

$F_{ik}(size)$ =

$$(1+\frac{I_{ik}}{N_i})\cdot\prod_{j=1}^{N_i}[(1+\frac{f_{ijk}}{\max(f_{ijk})})\cdot(1+\frac{H_{ij}}{T_{ij}})]$$

The weight of the kth feature of the ith product:

$F_{ik}(weight)$ =

$$(1+\frac{I_{ik}}{N_i})\cdot\prod_{j=1}^{N_i}\ [(\frac{N_i R_{ijk}}{\sum_{j=1}^{N_i}R_{ijk}})\cdot(1+\frac{H_{ij}}{T_{ij}})\cdot\frac{N_i O_{ij}}{\sum_{j=1}^{N_i}O_{ij}}]$$

All the font size and weight will be normalized from the smallest size to the biggest size.

The opinion words would employ the same rule, and the only difference is the whole scale of font size and weight.

4.2. Result

Figure 3 shows the result of opinion mining for the product Canon Powershot S3. You could see every feature word is following some opinion words. The font size of weight of all the words are based on the formula described above. The times of the feature word is also on the right down corner.

Through this picture we could see what features of this product mentioned by most users and the opinion words which describe them clearly. However, there is still some errors in the results, how to decrease them is what we should do in the next step. Also, in our future work, in order to prove our method is better than the traditional ways of present results, we would evaluate our system and conduct a comparison among similar systems by other researchers.

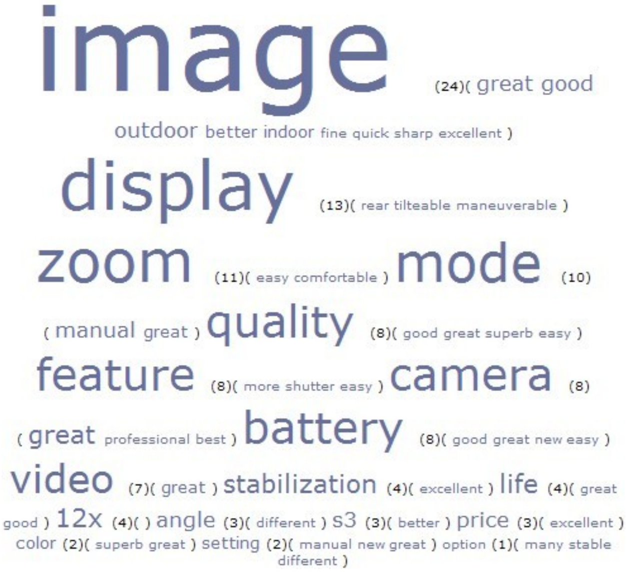


image (24)(great good
outdoor better indoor fine quick sharp excellent)
display (13)(rear tilteable maneuverable)
zoom (11)(easy comfortable) mode (10)
(manual great) quality (8)(good great superb easy)
feature (8)(more shutter easy) camera (8)
(great professional best) battery (8)(good great new easy)
video (7)(great) stabilization (4)(excellent) life (4)(great
good) 12x (4)( ) angle (3)(different) S3 (3)(better) price (3)(excellent)
color (2)(superb great) setting (2)(manual new great) option (1)(many stable
different)

Figure 3: the result of opinion mining

5. Conclusion

In this paper, we proposed a method to extract features and opinion words from a semi-structured review and design a tagcloud to display the results. The output of our system successfully generates features and opinion words tagclouds for products which is more straightforward than the traditional ways of listing result.

In our future work, we plan to improve and refine our system in several ways, such as extracting features words more precise, finding out the opinion words which are not expressed in adjective words and so on. We also will conduct to some work to evaluate our system and make comparisons with other researchers' systems. We believe that this problem will become increasingly important as more people are buying and expressing their opinions on the Web. Mining reviews is not only useful to sellers, buyers, but also crucial to product manufacturers.

References

[1] Turney, P. D. 2002. Thumbs up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), 417-424.

[2] Turney, P. D. and Littman, M. L. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Trans. On Information Systems, 21, 4 (2003), 315-346.

[3] Dave, K., Lawrence, S., and Pennock, D. M. 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic

Classification of Product Reviews. In Proceedings of the 12th international conference on World Wide Web (WWW'03), 519-528.

[4] Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP'02), 79-86.

[5] Pang, B. and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL'04), 271-278.

[6] Das, S. and Chen, M. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings of the 8th Asia Pacific Finance Association Annual Conference (APFA'01).

[7] Hu, M. and Liu, B. 2004. Mining and Summarizing Customer Reviews. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), 168-177.

[8] Zhuang, L., Jing, F., and Zhu, X. 2006. Movie Review Mining and Summarization. In Proceedings of the International Conference on Information and Knowledge Management (CIKM'06), 43-50.

[9] Popescu, A. and Etzioni, O. 2005. Extracting Product Features and Opinions from Reviews. In Proceeding of 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05), 339-346.

[10] Ding, X., Liu, B., and Yu, P. S. 2008. A Holistic Lexiconbased Approach to Opinion Mining. In Proceeding of the international conference on Web Search and Web Data Mining (WSDM'08), 231-239.

[11] http://www.cs.cornell.edu/home/llee/.

[12] Fellbaum, C. 1998. WordNet: an Electronic Lexical Database, MIT Press.

[13] http://en.wikipedia.org/wiki/Tag_cloud.

Incremental Maintenance of Minimal Bisimulation of Cyclic Graphs

Jintian Deng

ABSTRACT

Graph-structured databases have numerous recent applications including the Semantic Web, biological databases and XML, among many others. In this paper, we study the maintenance problem of a popular structural index, namely *bisimulation*, of a possibly cyclic data graph. To illustrate the design of our algorithm, first, we present some challenges of bisimulation minimization of cyclic graphs. Second, in the context of database applications, it is natural to compute minimal bisimulation with merging algorithms, as opposed to partition-refinement algorithms. We present a maintenance algorithm for a minimal bisimulation of a cyclic graph in the style of merging algorithm. Third, merging algorithms cannot determine the minimum bisimulation without examining all possible `SCC`s. We propose a feature-based optimization technique to prune the computation on non-bisimilar `SCC`s. The features are constructed and maintained more efficiently than bisimulation minimization. Finally, we present an experimental study that verifies the scalability of our algorithm and shows that our features-based optimization pruned 50% unnecessary bisimulation computation on average and when compared to previous work, our bisimulation can be 100% smaller, depending on the number of bisimilar `SCC`s in the data graph.

1. INTRODUCTION

Graph-structured databases have a wide range of recent applications, e.g., the Semantic Web, biological databases, `XML` and network topologies. To optimize the query evaluation in graph-structured databases, indexes have been proposed to summarize the paths of a data graph. In particular, many indexing techniques, e.g., [4,5,7,12,17,19,24], have been derived from the notion of *bisimulation* equivalence. In addition to indexing, bisimulation has been adopted as a notion of schemas for semi-structured data [3]. In recent works on selectivity estimation [14, 20, 21], bisimulation has also been used to construct synopses of graphs for path queries.

Two nodes in a data graph are bisimilar if they have the same set of incoming paths. To illustrate the application of bisimulation in graph-structured databases, we present a simplified sketch of a popular graph used in `XML` research, shown in Figure 1, namely `XMark`. `XMark` is a synthetic auction dataset: `open_auction` contains an author, a seller and a list of bidders, whose information is stored in `person`s; `person` in turn watches a few `open_auction`s. To model the bidding and watching relationships, `open_auction`s reference `person`s and vice versa. The references are encoded by `IDREF`s and represented by the dotted arrows in Figure 1.

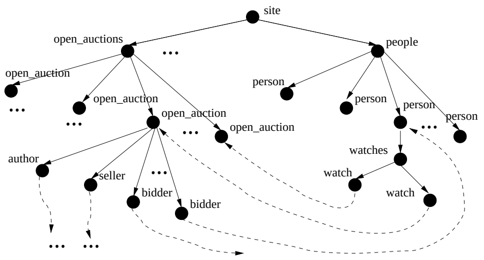


Figure 1: A simplified `XMark`

Without considering the references, `XMark` is a tree. It is evident that the bisimulation (see Appendix A) is smaller than the original data tree, which makes it a useful index for path queries. As an index structure, the bisimulation graph needs to be minimized.

In practice, data graphs are often cyclic (e.g., [1]) and subjected to updates. Therefore, when compared to other applications of bisimulation, its maintenance problem is much more important in database applications [13,22]. Our study on the maintenance problem of bisimulation of possibly cyclic graphs contributes to the current state-of-the-art in two aspects: (i) While there have been numerous applications on bisimulation, there has been relatively few work on its maintenance; (ii) Previous works [13,22] on maintenance of bisimulation of graphs mainly focus on directed *acyclic* graphs. There has not been explicit handling of cyclic structures.

There are two key challenges in maintaining minimal bisimulation of cyclic data graphs. Firstly, merging-based bisimulation algorithm as opposed to partition refinement is more natural for incrementally maintaining bisimulation. However, it is known [13] that merging-based algorithms fail to determine the minimum bisimulation of cyclic graphs. This is because the current merging step on a *strongly connected component* (`SCC`) causes subsequent merging steps to miss some bisimilar `SCC`s (to be detailed in Sections 3 and 4). Consequently, bisimulation minimization needs to examine many pairs of `SCC` as in the worst case there are exponentially many `SCC`s in a cyclic graph. Secondly, the nodes of `SCC`s must be considered *together* for checking bisimulation between `SCC`s of a cyclic graph. In particular, a node of an `SCC` can be bisimilar to a node of another `SCC` only if the two `SCC`s are bisimilar. However, merging-based algorithms compare one pair of nodes in each merging step.

In this paper, we carry out a comprehensive investigation on how to incrementally maintain minimal bisimulation of cyclic graphs. The first contribution is a study of some properties of bisimulation of cyclic graphs. These form the basis and terminologies of our discussions on the maintenance algorithm and influence the design of our algorithms. In particular, we study how a few nestings of cycles may affect bisimulation minimization. Our observation is that (i) determining the minimal bisimulation of cyclic graphs is tedious and subtle; and (ii) while it seems unlikely that there are many bisimilar cycles in a real-world data graph, it may not be reasonable to simply overlook them.

Second, we present a maintenance algorithm for minimal bisimulation of cyclic graphs. Similar to [13], our algorithm consists of a split and a merge phase. In the split phase, we split and mark the index nodes (i.e., the equivalence partitions) that are affected by an update. In the merge phase, we apply a (partial) bisimulation minimization algorithm on the marked index nodes. But different from [13], our algorithm has an explicit handling of bisimulation between SCCs. As such, our algorithm *always* produces smaller (if not the same) bisimulation graphs when compared to [13].

The third contribution is on our feature-based optimization for determining bisimulation between two SCCs. On the one hand, the computation of bisimulation between two SCCs can be costly. On the other hand, there may not be many bisimilar SCCs. Hence, we aim at deriving structural features from SCCs such that two SCCs are bisimilar only if they have the same features. Feature-based pruning has been a popular technique in determining subgraph isomorphism in graph-structured databases, among many others. We shall explore label-based, edge-based, path-based, tree-based and circuit-based features, by studying their pruning power, construction and maintenance efficiency.

2. RELATED WORK

Existing works on maintaining bsimulation can be categorized into two: *merging* and *partition-refinement* algorithms. There have been two previous merging algorithms [13,22] for incremental maintenance of bisimulation of cyclic graphs. The algorithm proposed in [13] contains a split and a merge phase. Upon an update on the data graph, the bisimulation graph is split to a correct but non-minimal bisimulation of the updated graph. Next, the bisimulation graph is minimized in the merge phase. For acyclic graphs, [13] produces the minimum bisimulation of the updated graph. If the graph is cyclic, [13] returns a minimal bisimulation only. Since [13] considers merging pairs of nodes iteratively, it does not minimize bisimulation between SCCs. Thus, to support cyclic graphs, the minimum bisimulation is occasionally re-computed from scratch. [22] can be considered as a follow-up of [13]. [22] proposes a split-merge-split algorithm with a rank flag for SCCs. The rank flag is originally proposed in [6] and adopted by [22]. The algorithm also returns a minimal bisimulation in response to an update of a cyclic graph. In comparison, our algorithm also contains the split and merge phases. A difference between our work and the previous works is that we provide efficient handling of SCCs and propose features to optimize bisimulation maintenance.

A recent partition-refinement algorithm [11] can be considered as a variant of Paige and Tarjan's algorithm [18] – a construction algorithm for the minimum bisimulation. The algorithm proposes its own split to handle edge changes. It has been extended to support maintenance of k-bisimulation. Their experiment shows that [11] produces a bisimulation that is always within 5% of the minimum bisimulation. It has been shown, through a later experiment, that [13] may produce even smaller bisimulations.

Bisimulation (relation) [16] has its root at symbolic model checking, state transition systems and concurrency theories. In a nutshell, two state transition systems are bisimilar if and only if they *behave* the same from an observer's point of view. Bisimulation minimization has been extensively studied through experiments in [8], in the context of modeling checking. A conclusion of [8] is that minimization may not be worthwhile for model checking as it may easily be more costly than checking invariance properties of systems. In comparison, when bisimulation is used as an index structure for query processing, bisimulation minimization and therefore its maintenance are far more important.

As discussed in Section 1, bisimulations have been recently used in numerous database applications. Indexes for path queries have been derived from bisimulation [5, 12, 17]. 1-index [17] adopts bisimulation as an index for regular paths. However, in practice, 1-index [17] can be large. A notion of local bisimulation, namely k-bisimulation, has been proposed to reduce index size. During query evaluation, local bisimulations may be combined to determine the complete path information. To balance query performance and index size, [5] proposes to adaptively adjust the k in k-bisimulation of subgraphs. [2, 4] consider bisimulation as a compressed instance of an XML repository for efficient query processing. In addition, bisimulation have been used as a summary structure for path query selectivity estimation, e.g., [20,21]. It is evident that a study on maintenance of bisimulation benefits all the above mentioned applications.

3. BACKGROUND

Next, we provide the background and the notations used.

Definition 3.1: A *graph-structured database* is a rooted directed labeled graph $G(V, E, r, \rho, \Sigma)$, where V is a set of nodes and E: $V \times V$ is a set of edges, $r \in V$ is a root node and $\rho : V \rightarrow \Sigma$ is a function that maps a vertex to a label, and Σ is a finite set of labels. □

For clarity, we may often denote a graph as $G(V,E)$ when r, ρ and Σ are irrelevant to our discussions.

Bisimulation. We recall the definition of bisimulation:

Definition 3.2: Given two graphs $G_1(V_1, E_1, r_1, \rho_1, \Sigma_1)$ and $G_2(V_2, E_2, r_2, \rho_2, \Sigma)$, an *upward bisimulation* $\sim$ is a binary relation between V_1 and V_2:

1. If (i) v_1 (resp. v_2) is the root of G_1 (resp. G_2) and (ii) $\rho(r_1) = \rho(r_2)$, then $v_1 \sim v_2$.

2. If (i) v_1 (resp. v_2) is not the root of G_1 (resp. G_2), (ii) $\rho(v_1) = \rho(v_2)$ and (iii) for each edge $(v_1', v_1) \in E_1$ (resp. $(v_2', v_2) \in E_2$), there is an edge $(v_2', v_2) \in E_2$ (resp. $(v_1', v_1) \in E_1$) such that $v_1' \sim v_2'$, then $v_1 \sim v_2$.

Two graphs G_1 and G_2 are *upward bisimilar* if an upward bisimulation can be established between G_1 and G_2. □

Definition 3.2 presents upward bisimulation in the sense that two nodes can be bisimilar only if their parents are bisimilar. This is a recursive definition – two nodes can be bisimilar only if their ancestors (up to the root) are bisimilar. The definition can be paraphrased in terms of paths. That is, two nodes are bisimilar if they have the same set of incoming paths. This definition is often convenient to simplify our discussions.

Proposition 3.1: *Two nodes are upward bisimilar if and only if the set of incoming paths of the two nodes are the same.* □

A set of bisimilar nodes is often referred to as an *equivalence partition* of nodes. Hence, a bisimulation of a graph is also often described as a set of equivalence classes.

We should remark that there have been other notions of bisimulation that have been applied in indexing/selectivity estimation but have not been the focus of this paper. The details of these definitions can be found in Appendix B. Our techniques can be extended to support them with some modifications. For presentation simplicity, we use bisimulation to refer to upward bisimulation, unless otherwise specified.

In this work, we consider the notion of bisimulation minimality presented in [13]. We paraphrase its definition below.

Definition 3.3: Given a bisimulation B of a graph G, B is *minimal* if for any two equivalence partitions I, $J \in B$, either (i) the nodes in I and J have different labels, or (ii) merging I and J results in some equivalence partition $K \in B$ unstable. □

Cyclic graphs. We present the definitions needed to discuss bisimulation of cyclic graphs. A *strongly connected component* (SCC) in a graph $G(V, E)$ is a subgraph $G'(V', E')$ whose nodes is a subset of nodes $V' \subseteq V$ where the nodes in V' can reach each other. The strongly connected components of a graph can be determined by classical algorithms, e.g., Gabow's algorithm, in $O(|V|+|E|)$.

Graph contraction has been a popular technique for processing cyclic data graphs. It is often convenient to use graph contraction in our discussions. Intuitively, given a cyclic graph, graph contraction reduces each strongly connected component into a supernode iteratively until an acyclic graph is obtained. Queries are often first processed on the reduced acyclic graph and then the supernodes (the strongly connected components). To present graph contraction, we recall the notion of exits and entries of SCCs.

Definition 3.4: A node n of an SCC $G'(V', E')$ of a graph $G(V, E)$ is an *exit* node if there exists an edge (n, n_1) where $n \in V'$ and $n_1 \notin V'$. Similarly, n is an *entry* node if there exists an edge (n_0, n) where $n_0 \notin V'$ and $n \in V'$. □

In general, an SCC can have multiple entry and exit nodes.

Definition 3.5: Given an SCC G' of a graph, a *one-step graph contraction* reduces G' into a supernode n and each incoming (resp. outgoing) edge to G' is modified to be an incoming (resp. outgoing) edge to n. □

The nestings of strongly connected components of a graph affect bisimulation minimization. To illustrate this, we define cycle height to describe the nestings of cycles.

Definition 3.6: The *cycle height h* (or simply *height*) of a graph is h if the height of its SCCs is at most h. The height of an SCC can be inductively defined as follows: A trivial SCC (a graph with a single node and no edge) has a height 0. An SCC s has a height $h + 1$ if the height of the SCCs in s is at most h. □

The cycle height of the nodes in G can be computed in $O(h \times (|E|+|V|))$, where h is the cycle height of the graph. It is straightforward that the cycle height of a node is not unique, which depends on the graph contraction steps.

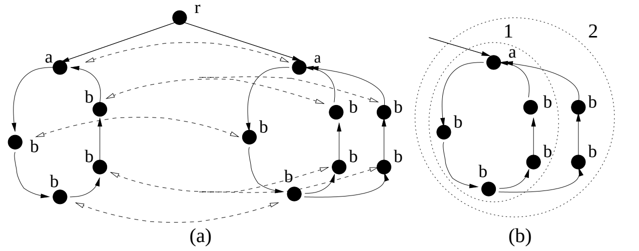


Figure 2: Two bisimilar SCCs with different cycle heights

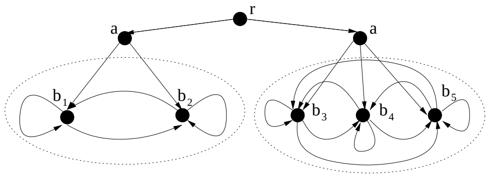


Figure 3: Two bisimilar SCCs with different numbers of entry nodes

4. BISIMULATION OF CYCLIC GRAPHS

In Subsection 4.1, we discuss bisimulation minimization in the presence of cyclic structures . In Subsection 4.2, we present a minimization algorithm for bisimulation of cyclic graphs, which is a major component of the maintenance algorithm presented in Section 5.

4.1 Properties of Bisimulation of Cyclic Graphs

We show a few properties of bisimulation of cyclic graphs. They shred some lights that it seems unlikely that merging algorithms could determine the minimum bisimulation between SCCs without examining many possible sub-SCCs.

Property 4.1: SCC*s with the same cycle height may not be bisimilar.* SCC*s with different cycle heights can be bisimilar.* □

The first part of this property is straightforward whereas the second part may require some elaborations. Consider a simple example shown in Figure 2(a). Consider the two SCCs in the figure. The cycle height of the SCC on the left is 1 while that on the right is 2 (Figure 2(b)). The dotted arrows in Figure 2(a) show a possible bisimulation between the two SCCs.

Property 4.2: *Two bisimilar* SCC*s with different numbers of entry nodes can be bisimilar.* □

A simple example is sketched in Figure 3. As discussed in Section 1, a possible bisimulation is $\{(b_1 \sim b_3), (b_2 \sim b_4), (b_2 \sim b_5)\}$. In another words, b_1 and b_3 have the same set of incoming paths and similarly, b_2, b_4 and b_5 have the same set of incoming paths.

Property 4.3: *Two bisimilar* SCC*s may have the different number of simple cycles. Two* SCC*s with the same number of simple cycles may not be bisimilar.* □

The first part of this property can be illustrated with Figure 2(a). The second part of the property can be illustrated with Figure 4. Both graphs in Figure 4 consist of 6 simple

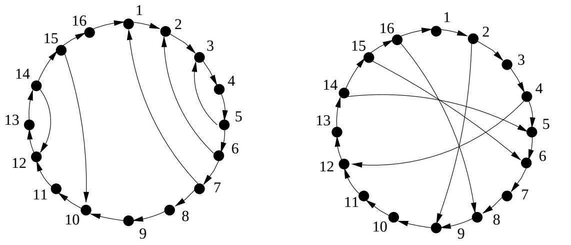


Figure 4: Cases of different overlapping cycles

cycles but are non-bisimilar. In addition, on LHS of Figure 4, the nodes of a cycle may be a subset of the nodes of other cycles. In comparison, on RHS of Figure 4, the cycles are overlapping but not contained in each other. When determining a bisimulation of a cycle, part of other cycles are involved. Hence, in the worst case, one may need to consider all sub-SCCs together, in the style of partition refinement [18].

Merging algorithms for bisimulation minimization are iterative in nature. Any merging algorithm could not return the minimum bisimulation since the current merging step of a simple cycle may affect other cycles. Therefore, to compute a minimal bisimulation of cyclic graphs, existing merging algorithms need to examine many possible SCCs.

4.2 Minimizing Bisimulation of Cyclic Graphs

Our algorithm `bisimilar_cyclic` for minimizing bisimulation of cyclic graphs is shown in Figure 5, which is a component of our maintenance algorithm. We assume the existence of a procedure `next_nodes_top_order`(G) of a node n which returns the next n's child in topological order in G.

The algorithm can be divided into two parts. First, Lines 01-06, if n_1 and n_2 are not both in some SCCs, we compute bisimulation between n_1 and n_2 in the sytle of a merging algorithm. In this case, we recursively invoke `bisimilar_cyclic`, for handling of cycles reachable from n_2.

Second, if both n_1 and n_2 are in some SCCs, Lines 07-21 check if S_1 *and* S_2, as opposed to simply n_1 and n_2, can be bisimilar. We prune non-bisimilar SCCs by using the feature-based optimization presented in Section 6, in Line 08. For presentation clarity, we assume that n_1 and n_2 are in two different SCCs. Then, we break the SCCs and check bisimulation recursively, in Lines 09-16. The main idea is illustrated with Figure 6. More specifically, we redirect the incoming edges of n_1 in n'_1 SCC (Lines 09-11) to an artificial node n'_1. Similarly, we redirect the incoming edges of n_2 to n'_2 (Lines 12-14). We clone the current bisimulation relation determined thus far (Line 15). Assuming that n_1 and n_2 are bisimilar, we check the possible bisimulation between the children of n_1 and n_2 by calling `bisimilar_cyclic` recursively (Lines 16-19). If we can construct a possible bisimulation between n'_1 and n'_2 (Line 20), then we declare that S_1 and S_2 are bisimilar.

The main difference between `bisimilar_cyclic` and [13] is that `bisimilar_cyclic` explicitly breaks a cycle when determining bisimulation between SCCs whereas [13] *does not* check bisimulation between SCCs *and* between SCCs' descendants. `bisimilar_cyclic` may be recursively called due to nested SCCs (Line 18). Without breaking a cycle, the feature-based optimization (Line 07) may always derive features of the "topmost" SCC. As verified by experiments (Figures 10(b) and 10(c)), the optimization will be essential for early backtracking in determining possible bisimulations between SCCs.

```
Procedure bisimilar_cyclic
Input: Nodes $n_1$ and $n_2$ where $\rho(n_1) = \rho(n_2)$, $n_1 \in G_1$
    and $n_2 \in G_2$; $B$, the current bisimulation relation
Output: An updated bisimulation relation $B'$
01 if $n_1$ and $n_2$ are not both in some SCC
02     if $\forall p_1 \in n_1$.parent $\exists p_2 \in n_2$.parent s.t. $p_1 \sim p_2$ then
03         add $(n_1, n_2)$ to $B$
04         for all $c_1$ in $n_1$.next_nodes_top_order$(G_1)$
05             for all $c_2$ in $n_2$.next_nodes_top_order$(G_2)$
06                 $B$ = bisimilar_cyclic$(c_1, c_2, B)$
07 else                 /* check bisimulation of the two SCCs */
08     assume $n_1$ and $n_2$ are in SCCs $S_1$ and $S_2$, respectively
       if feature_pruning$(S_1, S_2)$ return $B$      /* Sec. 6*/
09     clone $S_1$ to $S'_1$;   create an artificial node $n'_1$ for $n_1$
10     for all $(n, n_1) \in S'_1.E$
11         replace $(n, n_1)$ with $(n, n'_1) \in S'_1$
12     clone $S_1$ to $S'_1$;   create an artificial node $n'_2$ for $n_2$
13     for all $(n, n_2) \in S'_2.E$
14         replace $(n, n_2)$ with $(n, n'_2) \in S'_2$
15     clone $B$ to $B'$
16     add $(n_1, n_2)$ to $B'$                /* assume $n_1 \sim n_2$ */
17     for all $c_1$ in $n_1$.next_nodes_top_order$(S'_1)$
18         for all $c_2$ in $n_2$.next_nodes_top_order$(S'_2)$
19             $B'$ = bisimilar_cyclic$(c_1, c_2, B')$
20     if $(n'_1, n'_2)$ in $B'$ then $B = B \cup B'$      /* $S_1 \sim S_2$ */
21 return $B$
```

Figure 5: Bisimulation minimization of cyclic graphs

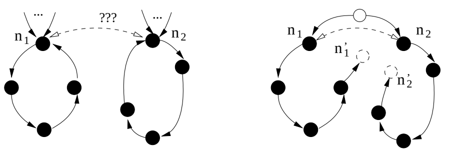


Figure 6: Breaking one cycle in an SCC

Analysis. For presentation clarity, `bisimilar_cyclic` did not incorporate with classical indexing techniques. `bisimilar_cyclic` runs in $O(|E|^2)$ due to the for loops at Lines 04-06 and Lines 17-19, assuming that `feature_pruning` can be performed more efficiently than `bisimilar_cyclic`. With classical indexing on nodes, the inner loop can be performed in $O(log(|V|))$ and the overall runtime is $O(|E|log(|V|))$.

5. MAINTENANCE OF BISIMULATION

We present an overall maintenance algorithm in this section. For simplicity, we present an edge insertion algorithm `insert` in Figure 7. Edge deletions are discussed at the end of this section. Our algorithm consists of a split phase and a merge phase, and with an explicit handling of SCCs. The merge phase is essentially bisimulation minimization, which has been detailed in `bisimilar_cyclic` in Section 4. In the following, we focus on the split phase.

The split phase. The split phase is presented in Lines 05-20. We maintain two variables to record two kinds of nodes that are needed to be split. More specifically, we use $\mathcal{S}$ to record the nodes of *SCCs* needed to be split and $\mathcal{Q}$ to record the *nodes* that are not in any SCCs but needed to be split.

```
Procedure insert
Input: an insertion of an edge $(n_1, n_2)$ a data graph $G$
    and its minimal bisimulation $B$
Output: An updated graph $G'$ and
    its updated minimal bisimulation $B'$
01 $G'$ = insert $(n_1, n_2)$ into $G$
02 if $n_2$ is new
   then create a new inode $I_{n_2}$; insert $I_{n_2}$ into $B$; mark $I_{n_2}$
   else if $I_{n_2}$ is not stable
03     $\mathcal{S} = \{(I_{n_2}, n_2) \mid n_2$ is in an SCC$\}$
04     $\mathcal{Q} = \{I_{n_2} \mid n_2$ is not in any SCC$\}$
05 while $\mathcal{Q} \neq \emptyset$ or $\mathcal{S} \neq \emptyset$
       /* split the relevant SCC */
06     if $\mathcal{S} \neq \emptyset$ then
07         pick a node $(I_n, n)$ from $\mathcal{S}$; remove $(I_n, n)$ from $\mathcal{S}$
08         while $I_n$ is not stable or a singleton
09           split $I_n$ into $I_1 = I_n$ - $\{n\}$ and $I_2 = \{n\}$
10           mark $I_1$ and $I_2$
11           $\mathcal{S} = \mathcal{S} \cup \{(I_{n_s}, n_s) \mid n_s$ is child of $n_i$, $n_i \in I_2$
                                           and $n_s$ in the SCC of $n\}$
12           $\mathcal{Q} = \mathcal{Q} \cup \{I_{n_q} \mid n_q$ is a child of $n_i$, $n_i \in I_2$
                                           and $n_q$ not in any SCCs$\}$
       /* split nodes not related to SCCs */
13     if $\mathcal{Q} \neq \emptyset$ then
14         pick a node $I_n \in \mathcal{Q}$; remove $I_n$ from $\mathcal{Q}$
15         if $I_n$ is not stable or a singleton
16             split $I_n$ into a stable set $\mathcal{I}$            // [13]
17             for each $I$ in $\mathcal{I}$
18                 mark $I$
19                 $\mathcal{S} = \mathcal{S} \cup \{(I_{n_s}, n_s) \mid n_s$ is a child of $n_i$, $n_i \in I$
                                               and $n_s$ in the SCC of $n\}$
20                 $\mathcal{Q} = \mathcal{Q} \cup \{ I_{n_q} \mid n_q \in$ child of $n_i$, $n_i \in I$
                                               and $n_q$ not in any SCCs$\}$
21 Gabow($G'$)          //update the SCC information in $G'$
/* merging the marked inodes */
22 $(G', B')$ = bisimilar_cyclic_marked($G$, $B$)
23 return $(G', B')$
```

Figure 7: Insertion for minimal bisimulation of cyclic graphs

Similar to previous work, we call an equivalence partition, which contains a set of bisimilar nodes, *inodes*, denoted with I. During the split phase we mark the affected inodes which will be examined in the merge phase.

Assume the insertion makes an inode of n_2 not stable. To initialize $\mathcal{S}$ (Line 03), we set $\mathcal{S}$ to the inode of n_2 and n_2, i.e., $\{(I_{n_2}, n_2)\}$, if n_2 is in some SCC. Otherwise, $\mathcal{S}$ is empty. Similarly, we initialize $\mathcal{Q}$ to I_{n_2} if n_2 is not in any SCC and empty otherwise (Line 04).

Next, we split the inodes recursively until $\mathcal{S}$ and $\mathcal{Q}$ are empty (Line 05).

(1) We process the nodes in $\mathcal{S}$ as follows (Lines 06-12): We select a node n from $\mathcal{S}$ and retrieve its inode I_n. We split n from I_n as the SCC of n is potentially non-bisimilar to the SCC of other nodes in I_n (Line 09). We mark the split inodes so that they will be checked in the merge phase (Line 10). In Lines 11-12, we insert the children of the split inode that are involved in some SCC(s) into $\mathcal{S}$ and the remaining children into $\mathcal{Q}$.

(2) The handling of $\mathcal{Q}$ is shown in Lines 13-20. We select an inode I_n from $\mathcal{Q}$ (Line 14). If I_n is not stable, we split I_n into a set of stable inodes $\mathcal{I}$, as in other works for maintaining bisimulation of acyclic graphs, e.g., [13] (Lines 15-16). We mark inodes in $\mathcal{I}$ in Line 18. In Lines 19-20, we update the affected nodes $\mathcal{S}$ and $\mathcal{Q}$, similar to Lines 11-12.

The split phase essentially traverses the bisimulation graph B and SCCs in the data graph to spilt and collect the inodes that are affected by the update. SCCs themselves may be affected by an update. In Line 21, we call Gabow's algorithm to update SCC information of a graph, which is needed in the merge phase.

The merge phase. The merge phase can be done by applying the minimization algorithm presented in Section 4.2 Figure 5. A simple optimization is that we do not apply merging on all inodes of the bisimulation graph but simply on the inodes that are marked in the split phase.

Example 5.1: We illustrate Algorithm insert with an example. A cyclic data graph is shown in Figure 8(a). For simplicity, we assume the node in the data graph has the same label. We show the node ids next to each node. Its minimal bisimulation is shown in Figure 8(b). We use {} to denote an inode. Assume that we insert an edge (20,17) into the data graph. Algorithm insert initially puts {12,17} into $\mathcal{Q}$ (Line 04). Then, in Line 16, node 17 is split from {12,17}. The split inodes are marked, with a "*" sign in the figure. The split phase proceeds recursively and finally produces the graph in Figure 8(c). We call Gabow'a algorithm to update the SCC information of the data graph. By calling bisimulation minimization, we obtain the bisimulation graph at Figure 8(d).

It should be remarked that while the previous work [13] produces the same split graph (Figure 8(c)). But, it returns Figure 8(c) as the final bisimulation graph. This is because it lacks the handling on SCCs as discussed in Section 4. Subsequently, any subgraph that is connected to the SCC with nodes 17, 18, 19 and 20, e.g., node 21, will not be merged, as the SCC is not merged. □

Analysis. The recursive procedure in Lines 05-20 traverses the graph $O(|E|)$. With optimization in [18], stablizing a set can be done in $O(log(|V|))$. Hence. the split phase runs in $O(|E|log(|V|)$. Gabow's algorithm in Line 21 runs in $O(|V| + |E|)$. The merge phase with optimization runs in $O(|E|log(|V|)$. Thus, the overall runtime of Algorithm insert is $O(|E|log(|V|)$.

Edge deletions. While our discussions focused on insertions, our technique can be generalized to support edge deletions with the following modifications. (i) In Line 01, we delete the edge from the data graph. (ii) If n_2 is connected after the deletion, we check the stability of I_{n_2} in Line 02, initialize $\mathcal{S}$ and $\mathcal{Q}$ and then invoke the split phase as before.

6. FEATURE-BASED OPTIMIZATION

As discussed in the previous section, determining if two SCCs are bisimilar can be computationally costly $O(|E|log(|V|))$. However, in practice, many SCCs may not be bisimilar. This motivates us to optimize bisimulation minimization of cyclic graphs by proposing features to prune computations on non-bisimilar SCCs.

In particular, we exploit the following property of a bisimulation between SCCs.

Proposition 6.1: *An* SCC *$G_1(V_1, E_1)$ is not bisimilar to another* SCC *$G_2(V_2, E_2)$ if and only if there is a node v in V_1 s.t. it is not bisimilar to any nodes in V_2.* □

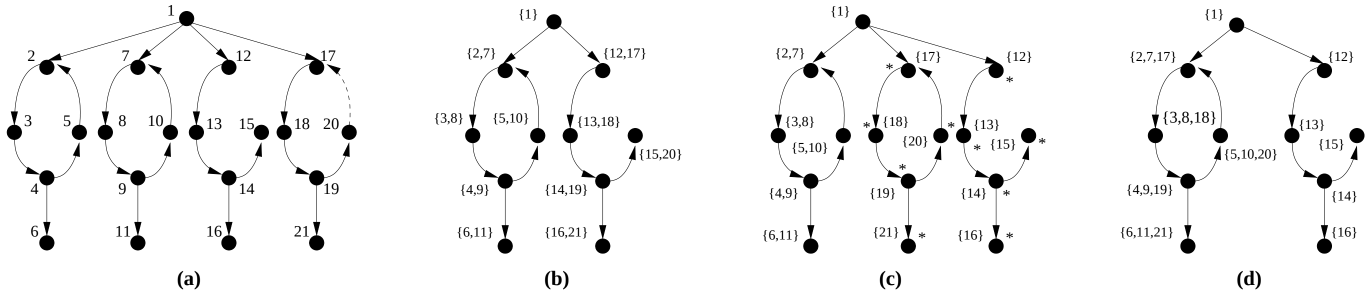


Figure 8: (a) A cyclic data graph; (b) the minimal bisimulation graph; (c) the split bisimulation graph; and (d) an updated minimal bisimulation graph

In this section, we adopt feature-based techniques for filtering computation on two non-bisimilar SCCs. The main idea is to derive a set of features from SCCs such that two SCCs can be bisimilar *only if* their features are the same or bisimilar. The features are ideally discriminative enough to reduce the computation on non-bisimilar graphs. In addition, while the maintenance of bisimulation is non-monotonic in nature, we design features that can be readily incrementally maintained. In the followings, we explore the details of label-based, edge-based, path-based, tree-based and circuit-based features.

Label-based or edge-based features. The label-based and edge-based features are straightforward and have many alternatives. For example, we may use all label and edge types that appeared in an SCC as an SCC feature. Obviously, two bisimilar graphs must contain the same type of labels and edges. In our experiments, we found that the incoming label or edge sets of an entry node are relatively concise and effective in distinguishing non-bisimilar SCCs. For example, in Figure 1, the incoming label set of the entry node open_auction is {open_auction, watch} and that of the entry node watches is {person, bidder}. The construction and maintenance of such labels can be efficiently supported by hashtables.

Path-based features. Regarding path-based features, one may be tempted to use all simple paths in an SCC. However, determining all simple paths of a cyclic graph is in PSPACE [15] and its maintenance is technically intriguing.

Proposition 6.2: *Two* SCC*s are bisimilar only if they have the same set of simple path(s) from their entry node(s).* □

There are other notions of paths that do not seem to be appropriate for our problem. For example, determining the longest paths of a cyclic graph is NP-complete.

In this work, we propose to use the set of incoming paths with a length at most k (or simply k-paths) as a feature of the entry nodes, where k is a user parameter. The value of k may be increased when maintenance of bisimulation spends substantial time on bisimulation computation. From Proposition 3.1, two bisimilar graphs must have the same set of k-paths. Contrarily, two graphs with different sets of k-paths must be non-bisimilar graphs. Hence, k-paths can be used as a feature. It is straightforward that k-paths can be efficiently constructed and maintained.

A remark is that k-paths may not consist of the node(s) that are not bisimilar to any nodes in any other SCC (Proposition 6.1). Another remark is that a node in an SCC may appear in a k-path set multiple times. Next, we propose a spanning tree as a feature of an SCC.

Feature of canonical spanning tree. First, we define the weight used in determining the canonical spanning tree. The *weight* of an edge (n_1, n_2) is *directly proportional* to the count of ($\rho(n_1)$, $\rho(n_2)$)-edges in the graph. We exploit a popular trick to perturb the weight of the edges such that each kind of edges has a unique weight.

Given the weight defined above, we can compute a minimum spanning tree, in the style of a greedy breath first traversal in $O(|V|+|E|)$. As the weight is defined to be directly proportional to the edge count, a minimum spanning contains more infrequent edge kinds of a graph. However, minimum spanning trees of a *directed* graph are often difficult to maintain. In comparison, maintenance of spanning trees of an undirected graph is much simpler, e.g., in amortized time $O(|V|^{1/3}log(|V|))$ [10]. Hence, we perform a couple of tricks on the data graph when constructing the spanning tree. First, we ignore the direction of the edges. Second, we adopt Prim's algorithm to construct the minimum spanning tree of the undirected graph. From the root of the minimum spanning tree, we derive the edge direction, which gives us the *canonical spanning tree.* (The edge direction is simply needed to check bisimulation between canonical spanning trees.) The direction of the edges in the canonical spanning tree may differ from that of the edges in the original graph.

Proposition 6.3: *Two* SCC*s are bisimilar only if their minimum canonical spanning trees returned by Prim's algorithm are bisimilar.* □

It should be remarked that SCCs are often nested. In the worst case, the total size of the spanning trees of all possible entry nodes of an SCC is $O((|V| + |E|)^2)$. In addition, computing bisimulation between large canonical spanning trees can be costly. Therefore, we introduce a termination condition to the Prim's algorithm – we do not expand the spanning tree further from a node n when there is an ancestor of n having the same label as n. The total size of the canonical spanning trees is then $O(|V| + |E|)$.

Example 6.1: We illustrate the construction of a canonical spanning discussed above with an example shown in Figure 9. Figure 9(a) shows a simplified SCC of open_auction from XMark with a scaling factor 0.1. The count of each edge type is shown on the edge. We perturb the weight to make each weight in the SCC unique. We ignore the direction of

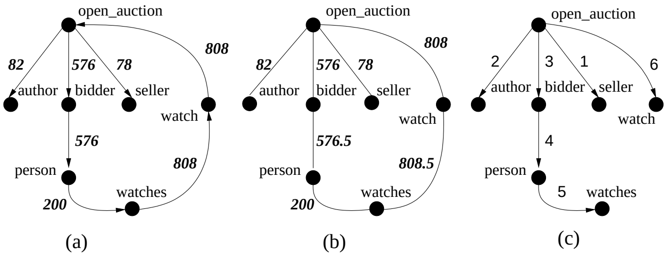


Figure 9: The construction of the canonical spanning tree from a simplified `open_auction`

the edges, shown in Figure 9(b). Then, it is straightforward to compute the spanning tree (shown in Figure 9(c), where the number on an edge shows the order of the edge is returned by Prim's algorithm). Finally, the direction of the edges are derived from the root of the tree `open_auction`. □

Circuit-based features. Finally, we discuss the feature of circuit bases, which contains much more structural information than spanning trees. It has been shown that the minimum circuit bases of directed graphs is unique [9]. Hence, one may be tempted to use circuit bases as a feature to prune computation on non-bisimilar graphs.

Proposition 6.4: *Two* `SCC`*s are bisimilar if their circuit bases are bisimilar.* □

However, determining the circuit bases is essentially $O(|V|^3)$. It is therefore more efficient to simply compute the bisimulation of two `SCC`s than using the feature of circuit bases.

6.1 Offline vs Online Feature Construction

Offline construction. The features of `SCC`s can be computed offline and used and maintained online (in Line 08 Figure 5). It has been known that determining all `SCC`s of a graph runs in exponential time. Therefore, we compute the features for *possible entry nodes* of `SCC`. To determine possible entry nodes, we apply Gabow's algorithm recursively. Gabow's algorithm returns the set of non-overlapping largest `SCC`s (therefore their entry nodes) of a graph. For each `SCC`, we remove an incoming edge of an entry node and apply Gabow's algorithm until no new entry nodes can be determined. The overall runtime of this method is evidently $O(|E|\times(|V|+|E|))$ and it returns all possible entry nodes. Finally, we compute the features for each possible entry node as discussed.

Online construction. Since the proposed features can be constructed efficiently, they may also be constructed during bisimulation computation, i.e., runtime. During runtime, we may incorporate the features with the partial bisimulation constructed so far for constructing features, for a higher pruning power. Specifically, since some nodes have been associated with an inode, we propose to use the id of inodes as opposed to the label alone to build features.

For example, consider the nested cycles on LHS of Figure 4. Assume all nodes have the same label. The cycles of (3,4,5) and (12,13,14) are obviously not bisimilar. However, the label/edge-based and path-based features cannot distinguish these two cycles. In runtime, when computing the bisimulation between (3,4,5) and (12,13,14), we have determined nodes 2 and 11 are not bisimilar. Hence, when we construct the label-based feature in runtime, node 3 has $\{A\{2\},...\}$ and node 12 has $\{A\{11\},...\}$. Thus, determining bisimulation between (3,4,5) and (12,13,14) is not needed.

7. EXPERIMENTAL EVALUATION

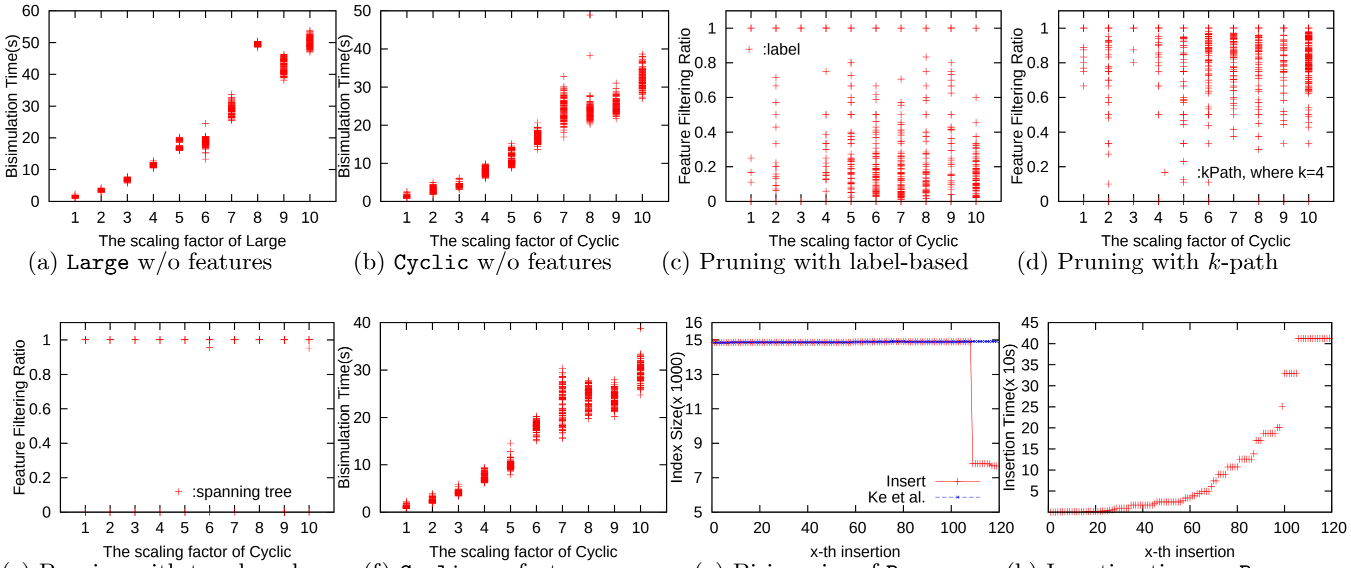
We present an experimental study on our algorithms. We modified the implementation of Ke *et al.* [13] to implement our algorithms. We used `XMark` [23] to obtain a group of graphs with one large `SCC`, denoted as `Large`. We decomposed `Large` to obtain another group of graphs with numerous `SCC`s, denoted as `Cyclic`. Details are in Appendix D. `Large` and `Cyclic` are needed to illustrate different aspects of our techniques.

Figures 10(a) and 10(b) show the performance of `bisimilar_cyclic` *without* feature-based optimization on `Large` and `Cyclic` with a scaling factor (s.f.) ranging from 0.01 to 0.1 (i.e., 1MB to 10MB). Since there is some randomness in the `SCC`s of `Large` and `Cyclic`, we ran 100 graphs for each s.f. Figures 10(a) and 10(b) show that the runtimes are roughly linear to s.f. At the same s.f. (hence same graph size), the runtime for `Large` is longer than that for `Cyclic`. The reason is that in `Cyclic`, there are many smaller random `SCC`s, which are often non-bisimilar, and `bisimilar_cyclic` can identify them relatively earlier. In comparison, `bisimilar_cyclic` in `Large` may spend more time in checking substructures in a large `SCC`.

Next, we verify the effectiveness of the features by using each feature on 100 `Cyclic` graphs for each s.f. The features were *computed in runtime* and k in the path-based feature is 4. We skipped the edge-based feature as its performance is similar to the label-based feature in `Cyclic`. The results are shown in Figures 10(c), 10(d) and 10(e). The y-axis is the percentage of non-bisimilar `SCC`s that were pruned by a feature. The label-based, path-based and canonical-tree feature pruned (on average) 14%, 62% and 73%, respectively. Figure 10(f) shows the runtime of `bisimilar_cyclic` with features. On average, it is 4% faster than that without features (Figure 10(b)). However, we remark that on average, 7.7% of the runtime is due to online feature construction.

Lastly, we present an experiment on Algorithm `insert`. We connect two `Large` graphs with a s.f. 0.01 and randomly remove 120 edges from the `SCC`s to form the base graph, denoted as `Base`. We insert the removed edges (randomly) one-by-one to `Base`. The result is shown in Figure 10(g). Figure 10(g) shows the size of the minimal bisimulation produced by `insert` and Ke *et al.* [13]. We did not show the result from Paige and Tarjan (the minimum) as `insert` always produces a bisimulation that is within 2% of the minimum. Initially, both `insert` and [13] are very close to the minimum. After some number of insertions, the two bisimilar `SCC`s in the original `Large` graph are recovered. We ran this experiment multiple times and find that this occurred randomly between 100th and 120th insertion. As shown in Figure 10(g), `insert` identifies the two bisimilar `SCC`s that lead to a bisimulation graph roughly 100% smaller than the one produced by [13]. We remark that the performance difference (in terms of bisimulation size) between `insert` and [13] depends on how many bisimilar `SCC`s are there in a graph.

The runtime of `insert` is shown in Figure 10(h). The runtime increases as we insert more edges into `Base`. After many insertions, `insert` runs slower because the two `SCC`s in `Base` become very similar. `bisimilar_cyclic` checks many nodes before it declares the `SCC`s are not bisimilar. The



(a) `Large` w/o features (b) `Cyclic` w/o features (c) Pruning with label-based (d) Pruning with k-path

(e) Pruning with tree-based (f) `Cyclic` w. features (g) Bisim. size of `Base` (h) Insertion time on `Base`

Figure 10: Scalability test of the minimization algorithm on `XMark`; the effectiveness of features; and the efficiency of the maintenance algorithm

runtime of [13] is close to 0s as it does not process `SCC`s.

8. CONCLUSIONS

In this paper, we studied the maintenance problem of minimal bisimulation of cyclic graph. To tackle the problem, we first presented a few properties about bisimulation on cyclic graphs. Second, we presented a bisimulation minimization algorithm that explicitly handles `SCC`s. Third, we presented a maintenance algorithm for minimal bisimulation of cyclic graphs. Fourth, we propose a feature-based optimization to avoid computation of non-bisimilar `SCC`s. We present an experiment to verify the scalability of our algorithms. In addition, our experiment shows that on average, the features can prune 50% unnecessary bisimulation computation. Our maintenance algorithm can return smaller bisimulation graphs (up to 100%) than previous work, depending the number of bisimilar `SCC`s in the data graph.

9. REFERENCES

[1] V. Batagelj and A. Mrvar. Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/.

[2] P. Buneman, B. Choi, W. Fan, R. Hutchison, R. Mann, and S. D. Viglas. Vectorizing and querying large XML repositories. In *ICDE*, 2005.

[3] P. Buneman, S. B. Davidson, M. F. Fernandez, and D. Suciu. Adding structure to unstructured data. In *ICDT*, 1997.

[4] P. Buneman, M. Grohe, and C. Koch. Path queries on compressed XML. In *VLDB*, 2003.

[5] Q. Chen, A. Lim, and K. W. Ong. D(k)-index: an adaptive structural summary for graph-structured data. In *SIGMOD*, 2003.

[6] A. Dovier, C. Piazza, and A. Policriti. An efficient algorithm for computing bisimulation equivalence. *Theor. Comput. Sci.*, 311(1-3):221–256, 2004.

[7] D. K. Fisher and S. Maneth. Structural selectivity estimation for XML documents. In *ICDE*, 2007.

[8] K. Fisler and M. Y. Vardi. Bisimulation minimization and symbolic model checking. *Form. Methods Syst. Des.*, 21(1):39–78, 2002.

[9] P. M. Gleiss, J. Leydold, and P. F. Stadler. Circuit bases of strongly connected digraphs. Working Papers 01-10-056, Santa Fe Institute, 2001.

[10] M. R. Henzinger and V. King. Maintaining minimum spanning trees in dynamic graphs. In *ICALP*, 1997.

[11] R. Kaushik, P. Bohannon, J. F. Naughton, and P. Shenoy. Updates for structure indexes. In *VLDB*, 2002.

[12] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. Exploiting local similarity for indexing paths in graph-structured data. In *ICDE*, 2002.

[13] Y. Ke, H. Hao, S. Ioana, and Y. Jun. Incremental maintenance of XML structural indexes. In *SIGMOD*, 2004.

[14] H. Li, M. L. Lee, W. Hsu, and G. Cong. An estimation system for XPath expressions. In *ICDE*, 2006.

[15] A. O. Mendelzon and P. T. Wood. Finding regular simple paths in graph databases. In *VLDB*, 1989.

[16] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[17] T. Milo and D. Suciu. Index structures for path expressions. In *ICDT*, 1999.

[18] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.

[19] N. Polyzotis and M. Garofalakis. XCluster synopses for structured XML content. In *ICDE*, 2006.

[20] N. Polyzotis and M. Garofalakis. XSketch synopses for XML data graphs. *ACM Trans. Database Syst.*, 31(3), 2006.

[21] N. Polyzotis, M. Garofalakis, and Y. Ioannidis. Approximate XML query answers. In *SIGMOD*, 2004.

[22] D. Saha. An incremental bisimulation algorithm. In *FSTTCS*, 2007.

[23] A. Schmidt, F. Waas, M. Kersten, M. J. Carey, I. Manolescu, and R. Busse. XMark: A benchmark for XML data management. In *VLDB*, 2002.

[24] J. Spiegel and N. Polyzotis. Graph-based synopses for relational selectivity estimation. In *SIGMOD*, 2006.

APPENDIX

A. BISIMULATION OF XMARK TREE

A sketch of the bisimulation (graph) of the `XMark` *tree* (shown in Figure 1 with dotted edges ignored) is shown in Figure 11, where bisimilar nodes are placed in an equivalence partition (enclosed by a rounded rectangle).

B. OTHER NOTIONS OF BISIMULATIONS

There has been a notion of *downward bisimulation*. Since its definition is very similar to Definition 3.2, we only highlight the difference between upward and downward bisimulations. Consider the second condition in Definition 3.2. Downward bisimulation, in contrast, states the following: if for any edge (v_1, v'_1) (resp. (v_2, v'_2)), there is an edge (v_2, v'_2) (resp. (v_1, v'_1)) such that $v'_1 \sim v'_2$ and $\rho(v'_1) = \rho(v'_2)$, then $v_1 \sim v_2$. This definition has been adopted for data graph compression [2,4], among others.

A local notion of bisimulation has been proposed, namely, *k-bisimulation*. Specifically, two nodes are k-bisimilar only if two nodes have the same set of incoming paths up to the length k. In other words, k-bisimulation considers bisimulation up to k steps only. While k-bisimulation may lose some path information, the size of k-bisimulation graph can often be small, in practice.

C. PROOF SKETCHES

Proof (sketch) of Proposition 6.3. The proof can be established by an induction on the run of Prim's algorithm on the graphs. The hypothesis is that the intermediate minimum spanning tree (or simply intermediate tree) of the size k returned by Prim's algorithm of G_1 is bisimilar to an intermediate tree of G_2 if G_1 and G_2 are bisimilar.

The base case is $k = 1$, where the intermediate tree with the root node only. Obviously, the base case is true. Suppose that the hypothesis is true up to the tree size of a size m and the tree for G_1 and G_2 are T_1 and T_2, respectively.

w.l.o.g, suppose that Prim's algorithm adds an edge to T_1 to form T'_1 which makes $T'_1 \not\sim T_2$. We want to prove that the next intermediate T'_2 by Prim's algorithm, where $T'_2 \not\sim T_2$, is bisimilar to T'_1. Suppose that the previous edge added to T'_1 and T'_2 are (a_1, a'_1) and (a_2, a'_2). Due to the hypothesis, $a_1 \sim a_2$. Since G_1 and G_2 are bisimilar, there must be a node a''_2 such that (a_2, a''_2) $\in G_2$ and $a'_1 \sim a''_2$. If $a''_2 = a'_2$, then $T'_1 \sim T'_2$. Otherwise, let $a''_2 \neq a'_2$. If $\texttt{weight}(a''_2) < \texttt{weight}(a'_2)$, then Prim's algorithm adds (a_2, a''_2), not (a_2, a'_2), to T_2. If $\texttt{weight}(a''_2) > \texttt{weight}(a'_2)$, then Prim's algorithm would not have returned T'_1 (contradiction).

Proof (sketch) of Proposition 6.4. Consider that two bisimilar graphs G_1 and G_2. Assume that C_1 and C_2 are the minimum circuit bases of G_1 and G_2, respectively. w.l.o.g, assume $c_1 \in C_1$ is the smallest circuit such that $c_1 \not\sim c_2$ for all $c_2 \in C_2$. Let p_1 be a simple path from the root of G_1 to the "first" node n_1 in c_1. Then, n_1 is not bisimilar to any node in G_2 because we can always find a path $p_1.c_1^i$, where i is an integer representing the repetition of c_1, to distinguish n_1 and any node in G_2.

D. ADDITIONAL INFORMATION FOR EXPERIMENTS

The implementation used in the experiment is available at http://code.google.com/p/minimal-bisimulation-cyclic-graphs/. The program is written in JDK 1.5. The implementation is run on a laptop computer with a dual CPU at 2.0 GHz and 2GB RAM running Ubuntu hardy.

We used the XMark dataset [23] to test variou aspects of our algorithms. The cycles in XMark is essentially composed by IDREFs of open_auction to person and vice versa. We ran Gabow's algorithm on XMark. We note that there are few very large SCCs. It is easy to verify that very few, or none, of the SCCs are bisimilar. Hence, we modify the cycles of XMark in the following way: We define a parameter s to set the average number of open_auction nodes and another parameter r to define the ratio between open_auction and person nodes in an SCC. For example, when s and r are set to 10 and 1.2, respectively, an SCC contains approximately 10 open_auctions and 12 persons.

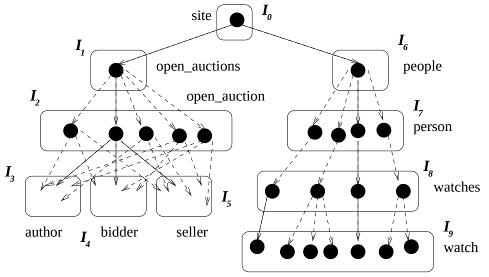


Figure 11: Bisimulation of the tree of XMark

In our experiment, the dataset generated directly from XMark is referred to Large. We set s and r to 10 and 1.2, respectively. The decomposed Large is refered to Cyclic.

In the experiment on Algorithm insert, we generated a dataset Base to test the performance difference between insert and Ke *et al.* The performance difference may be hardly shown systematically with Large because it only contains one large SCC. Cyclic contains numerous random non-bisimilar SCCs. In both cases, insert and Ke *et al.* return very similar bisimulation graphs. Therefore, we design Base to demonstrate the performance difference between the algorithms.

Base is constructed by connecting to XMark graphs with a s.f. 0.01 and removing 120 edges from the graph. Prior the removal of the edges, the graph has two bisimilar SCCs. When the edges are inserted by Algorithm insert, the bisimilar SCCs will be recovered and merged.

Acknowledgements. We are grateful to Yi Ke and Hao He for providing the implementation of [13]. We thank Yun Peng for his comments on the earlier drafts.

L-BFGS and Delayed Dynamical Systems Approach for Unconstrained Optimization

Xiaohui XIE

Abstract

The dynamical (or ode) systems approach for optimization problems has existed for two decades. The main feature of this approach is that a continuous path starting from the initial point can be generated and eventually the path will converge to the solution. This approach is quite different from conventional optimization methods where a sequence of points, or a discrete path, is generated. An advantage of the dynamical systems approach is that it is more suitable for large scale problems. Common examples of this approach are ode's based on the steepest descent direction or the Newton's direction. In this research we apply the L-BFGS scheme to the ode model, hopefully to improve on the rate of convergence over the steepest descent direction, but not to suffer from the large amount of computational work in the Newton's direction.

1 Problem background and introduction

This paper studies computational methods for a local or the global minimizer of an unconstrained optimization problem. Optimization problems are classified into:
(a) Unconstrained Problem:

$$\min_{x\in R^n} f(x) \qquad f: R^n \to R^1 \qquad (UP)$$

(b) Equality Constrained Problem:

$$\begin{aligned} &\min_{x\in R^n} f(x) \\ &\qquad h(x) \quad = 0 \qquad h: R^n \to R^p \end{aligned}$$

(c) Inequality Constrained Problem:

$$\begin{aligned} &\min_{x\in R^n} f(x) \\ &\qquad g(x) \quad \le 0 \qquad h: R^n \to R^p \end{aligned}$$

(d) General Constrained Problem:

$$\begin{aligned} &\min_{x\in R^n} f(x) \\ &\qquad g(x) \quad \le 0 \\ &\qquad h(x) \quad = 0 \end{aligned}$$

The motivation of unconstrained methods is to generate a sequence of points $\{x_k\}$ (x_0 given) such that (1) $f(x_k) > f(x_{k+1})$;(2) $\{x_k\}$ is convergent, and (3) the limit point of the sequence is a stationary point of (UP). Different methods advance from x_k to x_{k+1} differently. Well-known methods include the steepest descent method, Newton's method and quasi-Newton method. A common theme behind all these methods is to find a direction $p \in R^n$ so that there exists an $\bar{\epsilon} > 0$ such that

$$f(x + \epsilon p) < f(x) \qquad \forall \epsilon \in (0, \bar{\epsilon})$$

This direction is called a descent direction of $f(x)$ at x. Once we have found a descent direction, we may go along this direction to approach one more step toward the optimum solution.

The following paragraphs summarize the advantages and disadvantages of these methods.

1.1 Steepest descent method

Using directional derivative in multivariable calculus, it is clear that for (UP), p is a descent direction at $x \Leftrightarrow \nabla f(x)^T p < 0$. Hence $p = -\nabla f(x)$, or equivalently, $p = -\nabla f(x)/\|\nabla f(x)\|_2$ is obviously a descent direction for $f(x)$. This direction is called the steepest descent direction.

Method of Steepest Descent: At each iteration k: find the lowest point of f in the direction $-\nabla f(x_k)$ from x_k, i.e., find λ_k that solves

$$\min_{\lambda>0} f(x_k - \lambda \nabla f(x_k))$$

Then $x_{k+1} = x_k - \lambda \nabla f(x_k)$.

Unfortunately, the steepest descent method converges only linearly, and sometimes very slowly linearly. In fact, if $\{x_k\}$, which is generated by the steepest descent method, converges to a local minimizer x^* where $\nabla^2 f(x^*)$ is positive definite (p.d.), and λ_{max} and λ_{min} are the largest and smallest eigenvalues of $\nabla^2 f(x^*)$, then one can show that $\{x_k\}$ satisfies

$$\lim_{k\to\infty} sup \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq c, \quad c = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}},$$

in a particular weighted l_2 norm, and the bound on c is tight for some starting x_0. This property indicates that the steepest descent method is q-linearly convergent. When λ_{max} and λ_{min} are far apart, then c is close to 1, and the convergence will be slow.

1.2 Newton's method

At point x_k, if $\nabla^2 f(x_k)$ is p.d., the function $f(x)$ can be approximated by a quadratic function based on the Taylor expansion:

$$\begin{aligned} f(x) \cong\ & f(x_k) + \nabla f(x_k)^T (x - x_k) \\ & + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k) \end{aligned} \tag{1}$$

Then the minimizer of (1) is given by

$$\begin{aligned} \nabla f(x) &= 0 \\ &\Rightarrow \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0 \\ &\Rightarrow x = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k), \end{aligned}$$

where $-[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$, is called the Newton's direction. Then we define

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k),$$

and the resulting method of computing x_k is called the Newton's method.

Newton's method

Given x_0, compute

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k), \qquad k \leftarrow k + 1.$$

A key requirement for Newton's method is the p.d. of $\nabla^2 f(x_k)$. Descent directions guarantee that $f(x)$ can be further reduced and therefore they form the basis of some global methods. However, in some real applications, if the starting point is far away from the optimal solution, or the Hessian is not positive definite, Newton's direction is not adopted. Although Newton's method converges very fast ($\{x_k\}$ converges to x^* q-quadratically), the Hessian matrix is difficult to compute. So we would like to find more feasible methods with (A) no second-order information, i.e., no Hessian; and (B) fast convergence. A rule of thumb is that first-order information normally gives slow (linear) convergence, while second-order information normally gives fast (quadratic) convergence. Let us discuss several practical considerations. In general, the convergence is quadratic: the error is essentially squared at each step (that is, the number of accurate digits doubles in each step). There are some caveats, however. Firstly, Newton's method requires that the derivative be calculated directly. (If the derivative is approximated by the slope of a line through two points on the function, the secant method results; this can be more efficient depending on how one measures computational effort.) Secondly, if the initial value is too far from the true zero, Newton's method can fail to converge. Because of this, most practical implementations of Newton's method put an upper limit on the number of iterations and perhaps on the size of the iterates. Thirdly, if the root being sought has multiplicity greater than one, the convergence rate is merely linear (errors reduced by a constant factor at each step) unless special procedures are taken. Finding the inverse of the Hessian is an expensive operation. Therefore the descent direction $-[\nabla^2 f(x)]^{-1} \nabla f(x)$ is often solved approximately (but to great accuracy) using methods such as the conjugate gradient method. There also exist various quasi-Newton methods, where an approximation for the Hessian is used instead.

Table 1 compares the advantages and disadvantages between the steepest descent method and the Newton's method.

Table 1: **Comparison of the methods**

	Advantage	Disadvantage
Steepest descent method	Simple and inexpensive, guarantees descent	Slow convergence
Newton's method	Very fast convergence if applicable	Expensive, second-order information matrix inversion

1.3 Quasi-Newton method—BFGS

Instead of using the Hessian matrix, the quasi-Newton methods approximate it.

Quasi-Newton methods are based on Newton's method to find the stationary point of $f(x)$, where the gradient $\nabla f(x)$ is 0. In Quasi-Newton methods the Hessian matrices of second derivatives of $f(x)$ do not need to be computed. The Hessian is updated by analyzing successive gradient vectors instead. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. In multi-dimensions the secant equation is under-determined, and quasi-Newton methods differ in how they constrain the solution, typically by adding a simple low-rank update to the current estimate of the Hessian.

In quasi-Newton methods, the inverse of the Hessian matrix is approximated in each iteration by a p.d. matrix, say H_k, where k is the iteration index. Thus, the k th iteration has the following basic structure:
(a) set $p_k = -H_k g_k, \quad (g_k = \nabla f(x_k))$,
(b) line search along p_k giving $x_{k+1} = x_k + \lambda_k p_k$,
(c) update H_k giving H_{k+1}.

The initial matrix H_0 can be any positive definite symmetric matrix, although in the absence of any better estimate, the choice $H_0 = I$ often suffices. Potential advantages of the method are:
(1) only first-order information is required;
(2) H_k being symmetric and p.d. implies the descent property; and
(3) $O(n^2)$ multiplications per iteration.

The most important quasi-Newton formula was suggested by Broyden, Fletcher, Goldfarb, and Shanno independently in 1970, and is subsequently known as the BFGS formula. It is used to solve an unconstrained nonlinear optimization problem.

$$H_{k+1}^{BFGS} = H_k + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k}\right)\frac{s_k s_k^T}{s_k^T y_k} - \left(\frac{s_k y_k^T H_k + H_k y_k s_k^T}{s_k^T y_k}\right) \tag{2}$$

where $s_k = x_{k+1} - x_k$,
$y_k = \nabla f(x_{k+1}) - \nabla f(x_k) = g_{k+1} - g_k$.

We have the following theorem.

Theorem 1.1 *If H_k^{BFGS} is a p.d. matrix, and $s_k^T y_k > 0$, then H_{k+1}^{BFGS} in (2) is also positive definite.*

Proof: For any $z \neq 0$, it is sufficient to prove that

$$z^T\left[H_k + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k}\right)\frac{s_k s_k^T}{s_k^T y_k} - \left(\frac{s_k y_k^T H_k + H_k y_k s_k^T}{s_k^T y_k}\right)\right] z > 0,$$

In the rest of the proof, the subscript k will be omitted. Since H is p.d., we can write $H = LL^T$, and let
$a = L^T z$ and $b = L^T y$, then

$$
\begin{aligned}
z^T H_{k+1} z &= z^T \left[H_k + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k} \right. \\
&\quad \left. - \left(\frac{s_k y_k^T H_k + H_k y_k s_k^T}{s_k^T y_k} \right) \right] z \\
&= z^T \left[H + \left(1 + \frac{y^T H y}{s^T y} \right) \frac{ss^T}{s^T y} - \left(\frac{sy^T H + Hys^T}{s^T y} \right) \right] z \\
&= a^T a + \frac{z^T b^T b s s^T z}{(s^T y)^2} + \frac{(z^T s)^2}{s^T y} - \frac{z^T s b^T a}{s^T y} - \frac{a^T b s^T z}{s^T y} \\
&= \left(a - \frac{z^T s b}{s^T y} \right)^T \left(a - \frac{z^T s b}{s^T y} \right) + \frac{(z^T s)^2}{s^T y} \\
&= \left\| a - \frac{z^T s b}{s^T y} \right\|^2 + \frac{(z^T s)^2}{s^T y} \\
&\geq 0
\end{aligned}
$$

If the norm above equals zero, i.e.,

$$\left\| a - \frac{z^T s b}{s^T y} \right\| = 0,$$

then we have

$$a = \frac{z^T s b}{s^T y}, \text{ or } L^T z = \frac{z^T s L^T y}{s^T y},$$

which means that $y \propto z$.

However, since $s^T y > 0$,

$$\frac{(z^T s)^2}{s^T y} > 0,$$

as $y \propto z$. Thus the theorem is proved. □

1.4 Limited-Memory Quasi-Newton Methods—L-BFGS

Limited-memory quasi-Newton methods are useful for solving large problems whose Hessian matrices cannot be computed at a reasonable cost or are not sparse. These methods maintain simple and compact approximations of Hessian matrices: instead of storing fully dense $n \times n$ approximations, they save only a few vectors of length n that represent the approximations implicitly. Despite these modest storage requirements, they often yield an acceptable rate of convergence. Various limited-memory methods have been proposed; we focus mainly on an algorithm known as L-BFGS, which, as its name suggests, is based on the BFGS updating formula. The main idea of this method is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behavior of the Hessian at the current iteration, is discarded in the interest of saving storage.

As we have discussed in section 1.3, each step of the BFGS method has the form

$$x_{k+1} = x_k - \alpha_k H_k \nabla f_k,$$

where α_k is the step length and H_k is updated at every iteration by means of the formula

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \tag{3}$$

where

$$\rho_k = \frac{1}{y_k^T s_k}, \qquad V_k = I - \rho_k y_k s_k^T, \tag{4}$$

and

$$s_k = x_{k+1} - x_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k. \tag{5}$$

Since the inverse Hessian approximation H_k will generally be dense, the cost of storing and manipulating it is prohibitive when the number of variables is large. To circumvent this problem, we store a modified version of H_k implicitly, by storing a certain number (say, m) of the vector pairs $\{s_i, y_i\}$ used in the formulas (3)-(5). The product $H_k \nabla f_k$ can be obtained by performing a sequence of inner products and vector summations involving ∇f_k and the pairs $\{s_i, y_i\}$. After the new iterate is computed, the oldest vector pair in the set of pairs $\{s_i, y_i\}$ is replaced by the new pair $\{s_k, y_k\}$ obtained from the current step (5).

We now describe the updating process in a little more detail. At iteration k, the current iterate is x_k and the set of vector pairs is given by $\{s_i, y_i\}$ for $i = k - m, \ldots, k - 1$. We first choose some initial

Hessian approximation H_0 (in contrast to the standard BFGS iteration, this initial approximation is allowed to vary from iteration to iteration) and find by repeated application of the formula (3) that the L-BFGS approximation H_{k+1} satisfies the following formula [12]:

In general, we have for $k+1 \leq m$ the usual BFGS updated

$$\begin{aligned} H_{k+1} &= V_k^T V_{k-1}^T \cdots V_0^T H_0 V_0 \cdots V_{k-1} V_k \\ &\quad + V_k^T \cdots V_1^T \rho_0 s_0 s_0^T V_1 \cdots V_k \\ &\quad \vdots \\ &\quad + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} v_k \qquad (6) \\ &\quad + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &\quad + \rho_k s_k s_k^T. \end{aligned}$$

For $k+1 > m$ we have the update

$$\begin{aligned} H_{k+1} &= V_k^T V_{k-1}^T \cdots V_{k-m+1}^T H_0 V_{k-m+1} \cdots V_{k-1} V_k \\ &\quad + V_k^T \cdots V_{k-m+2}^T \rho_{k-m+1} s_{k-m+1} s_{k-m+1}^T \\ &\qquad \cdot V_{k-m+2} \cdots V_k \\ &\quad \vdots \qquad (7) \\ &\quad + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} V_k \\ &\quad + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &\quad + \rho_k s_k s_k^T. \end{aligned}$$

A method for choosing H_0 that has proved effective in practice is to set $H_0 = \gamma_k I$, where

$$\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}.$$

The strategy of keeping the m most recent correction pairs $\{s_i, y_i\}$ works well in practice; indeed no other strategy has yet proved to be consistently better. However, the main weakness of the L-BFGS method is that it converges slowly on ill-conditioned problems — specifically, on problems where the Hessian matrix contains a wide distribution of eigenvalues. On certain applications, the nonlinear conjugate methods are competitive with limited-memory quasi-Newton methods.

Algorithm (L-BFGS)

```
Choose starting point $x_0$, integer $m > 0$;
$k \leftarrow 0$;
repeat
  Choose $H_0$
  Compute $p_k \leftarrow -H_k \nabla f_k$
  Compute $x_{k+1} \leftarrow x_k + \alpha_k p_k$ where $\alpha_k$ is chosen
     to satisfy the Wolfe conditions;
  if $k > m$
     Discard the vector pair $\{s_{k-m}, y_{k-m}\}$ from
       storage;
  Compute and save
     $s_k \leftarrow x_{k+1} - x_k, y_k = \nabla f_{k+1} - \nabla f_k$;
  $k \leftarrow k + 1$;
until convergence.
```

2 Analysis for dynamical systems with time delay

2.1 Introduction of dynamical systems

For the easiness of reading, the (UP) problem is reproduced here:

$$\min_{x \in R^n} f(x) \qquad f : R^n \to R^1 \tag{8}$$

It is very important that the optimization problem (8) itself is posted in the continuous form, i.e., x can be changed continuously. In the literature, the necessary and sufficient conditions of a local optimum are also presented in the continuous form. Furthermore, almost all the theoretical study for problem (8) is in the continuous form. However, it is very interesting to say that when it comes down to the numerical solution of (8), most of the conventional methods, such as the gradient/ steepest descent method, Newton's method and quasi-Newton's method, are all addressed in the discrete form. This interesting situation is mainly due to the fact that the computer's computation can be only done discretely. However, is it possible to study both the optimization problem and the solution methods in its original form, i.e., continuous form? In this sense, we may use the dynamical system approach or neural network approach to solve the original optimization problem.

- Dynamical system approach. The essence of this approach is to convert problem (8) into a dynamical system or an ordinary differential equation (ode) so that the solution of problem (8) corresponds to a stable equilibrium point of this dynamical system.
- Neural network approach. The mathematical representation of neural network is an ordinary differential equation which is asymptotically stable at any isolated solution point. A companion of this neural network is an energy function which is a Lyapunov function. And as time evolves, the solution of the ode will converge to the optimum, and in this whole process, the energy function will decrease monotonically in time.

The following discussion reviews the research results in the dynamical system approach, and identifies the merits of this approach.

Consider the following simple dynamical system or ordinary differential equation

$$\frac{dx(t)}{dt} = p(x). \tag{9}$$

We first state some classical result on the existence and uniqueness of the solution, and some stability definitions for the dynamical system (9) [21,24].

Theorem 2.1 [24] *Assume that $p(x)$ is a continuous function from R^n to R^n. Then for arbitrary $t_0 \geq 0$ and $x_0 \in R^n$ there exists a local solution $x(t)$ satisfying $x(t_0) = x_0$, $t \in [t_0, \tau)$ to (9) for some $\tau > t_0$. If furthermore $p(x)$ is locally Lipschitz continuous at x_0, then the solution is unique, and if $p(x)$ is Lipschitz continuous in R^n then τ can be extended to ∞.*

Definition 2.2 *(Equilibrium point) A point $x^* \in R^n$ is called an equilibrium point of (9) if $p(x^*) = 0$.*

Definition 2.3 *(Stablility in the sense of Lyapunov) Let $x(t)$ be the solution of (9). An isolated equilibrium point x^* is Lyapunov stable if for any $x_0 = x(t_0)$ and any scalar $\epsilon > 0$, there exists a $\delta > 0$ such that if $\|x(t_0) - x^*\| < \delta$, then $\|x(t) - x^*\| < \epsilon$ for $t \geq t_0$.*

Definition 2.4 *(Convergence) Let $x(t)$ be the solution of (9). An isolated equilibrium point x^* is convergent if there exists a $\delta > 0$ such that if $\|x(t_0) - x^*\| < \delta$, $x(t) \to x^*$ as $t \to \infty$.*

A dynamical system or ode (9) arising from an optimization problem needs to have $p(x)$ being a descent direction for the objective function $f(x)$. Some well-known versions are:

Dynamical system based on the steepest descent direction:

$$\frac{dx(t)}{dt} = -\nabla f(x(t))$$

Dynamical system based on the Newton direction:

$$\frac{dx(t)}{dt} = -[\nabla^2 f(x(t))]^{-1} \nabla f(x(t)) \tag{10}$$

As in the discrete optimization methods in the previous chapter, the steepest descent direction has a slow convergence rate - meaning that it takes a very "large" value of t to approach the equilibrium point. The Newton direction has a much faster convergence rate, but the amount of work in evaluating the Jacobian is much greater.

Some other dynamical systems in the literature are:

$$\frac{dx(t)}{dt} = s(t) \cdot p(x(t)), \tag{11}$$

$$a(t) \cdot \frac{d^2x(t)}{dt^2} + b(t) \cdot B(x(t)) \cdot \frac{dx(t)}{dt} = p\left(x(t)\right), \tag{12}$$

where $p(x)$is a descent direction for $f(x)$, $B(x) \in R^{n \times n}$ is a positive definite matrix, $a(t)$ and $b(t)$ are scalar functions in t, and $s(t)$ is a positive scalar function in t and bounded above.

A major advantage of the dynamical systems approach is that very large problems can be solved [13]. No matter whether we use any of (9)-(12), existing ode methods are quite mature to tackle these problems. Solving systems with tens or hundreds of thousands of unknowns poses no problem to ode solvers. The problem size handled can be much larger than traditional methods described in Chapter 1, which are of order of magnitude in the thousands. The research in the ode approach is to find a "good" $p(x)$ in (9) that balances the convergence rate and the amount of work.

The dynamical systems approach normally consists of the following three steps:

(a) to establish an ode system;

(b) to study the convergence of the solution $x(t)$ of the ode as $t \to \infty$; and

(c) to solve the ode system numerically.

The convergence study of $x(t)$ as $t \to \infty$ and the stability of the corresponding dynamical system have mostly been addressed on a case by case base. No standard theory and/or methodology are given. This phenomenon certainly limits the systematic study of the dynamical system approach and its application potential as well. Two papers are worth mentioning, one by Tanabe [18] which used the stability theory of the dynamical system to study the ode system, and the other one by Yamashita [24] which employed Lyapunov's direct method to study the ode system.

Even though the solutions of ode systems are continuous, yet the actual computation has to be done discretely. In all the dynamical systems (10)-(12), the numerical solutions were mainly solved by either discrete optimization methods or finite difference methods.

In summary, the main attractiveness of this approach is its simplicity and its originality in pursuing the continuous form. Furthermore, there is not any restriction on the form of the objective function $f(x)$ in (8).

2.2 Delayed dynamical systems approach

As stated above, the steepest descent direction and the Newton direction of the dynamical systems approach both have their weakness. The main idea in this paper is to apply the theme of the L-BFGS algorithm in Chapter 1 to the dynamical systems approach, making it a bridge between the steepest descent direction and the Newton direction. The resulting dynamical system is a delayed ode, thus we call it the delayed dynamical systems approach.

The delayed dynamical systems approach solves the delayed ode:

$$\begin{aligned}\frac{dx(t)}{dt} &= -H(x(t), x(t-\tau_1(t)), \ldots, x(t-\tau_m(t))) \\ &\qquad \cdot\nabla f(x(t)),\end{aligned} \tag{13}$$

where H and $t-\tau_1(t), \ldots, t-\tau_m(t)$ are to be defined below.

As the delayed ode (13) is numerically solved, we compute approximations $x_0, x_1, \ldots, x_k, x_{k+1}, \ldots$ to $x(t)$ at time points $t_0, t_1, \ldots, t_k, t_{k+1}, \ldots$. We define $H = H_k$ to be a different function in the interval $(t_{k-1}, t_k]$ iteratively by:

Given $t_0, x(t_0) = x_0$, and an initial H_0, for $t_0 \le t$, we define

$$\begin{aligned}H(x(t), x(t_0)) \quad &:= \quad H_1(x(t), x(t_0)) \\ &:= \quad V_0(t)^T H_0 V_0(t) + \rho_0(t) s_0(t) s_0(t)^T,\end{aligned}$$

where

$$s_0(t) = x(t) - x_0,$$

$$y_0(t) = \nabla f(x(t)) - \nabla f(x_0),$$

$$\rho_0(t) = 1/y_0(t)^T s_0(t),$$

$$V_0(t) = I - \rho_0(t) y_0(t) s_0(t)^T,$$

in the R.H.S. of (13) and determine a stepsize $h_1 = (t_1 - t_0)$ to compute, using some numerical ode method, an approximation x_1 to $x(t_1)$ at t_1. Then for $t_1 \le t$, we define

$$\begin{aligned}&H(x(t), x(t_1), x(t_0)) \\ &\quad := H_2(x(t), x(t_1), x(t_0)) \\ &\quad := V_1(t)^T V_0(t_1)^T H_0 V_0(t_1) V_1(t) \\ &\qquad + V_1(t)^T \rho_0(t_1) s_0(t_1) s_0(t_1)^T V_1(t) \\ &\qquad + \rho_1(t) s_1(t) s_1(t)^T\end{aligned}$$

where

$$s_1(t) = x(t) - x(t_1),$$

$$y_1(t) = \nabla f(x(t)) - \nabla f(x(t_1)),$$

$$\rho_1(t) = 1/y_1(t)^T s_1(t),$$

$$V_1(t) = I - \rho_1(t) y_1(t) s_1(t)^T,$$

in the R.H.S. of (13) and determine a stepsize $h_2 = (t_2 - t_1)$ to compute, using some numerical ode method, an approximation x_2 to $x(t_2)$ at t_2. Of course, computationally x_1 is used instead of $x(t_1)$.

This process is repeated until we have accepted x_{m-1} at t_{m-1}. Then for $t_{m-1} \leq t$, we use

$$\begin{aligned}
&H(x(t), x(t_{m-1}), \dots, x(t_1), x(t_0)) \\
&\quad := H_m(x(t), x(t_{m-1}), \dots, x(t_1), x(t_0)) \\
&\quad := V_{m-1}(t)^T V_{m-2}(t_{m-1})^T \dots V_1(t_2)^T V_0(t_1)^T \\
&\qquad \cdot H_0 V_0(t_1) V_1(t_2) \dots V_{m-2}(t_{m-1}) V_{m-1}(t) \\
&\qquad + V_{m-1}(t)^T V_{m-2}(t_{m-1})^T \dots V_1(t_2)^T \\
&\qquad \cdot \rho_0(t_1) s_0(t_1) s_0(t_1)^T V_1(t_2) \dots V_{m-2}(t_{m-1}) \\
&\qquad \cdot V_{m-1}(t) \\
&\qquad + \dots \\
&\qquad + V_{m-1}(t)^T \rho_{m-2}(t_{m-1}) s_{m-2}(t_{m-1}) \\
&\qquad \cdot s_{m-2}(t_{m-1})^T V_{m-1}(t) \\
&\qquad + \rho_{m-1}(t) s_{m-1}(t) s_{m-1}(t)^T \qquad (13A)
\end{aligned}$$

where

$$\begin{aligned}
s_{m-1}(t) &= x(t) - x(t_{m-1}), \\
y_{m-1}(t) &= \nabla f(x(t)) - \nabla f(x(t_{m-1})), \\
\rho_{m-1}(t) &= 1/y_{m-1}(t)^T s_{m-1}(t), \\
V_{m-1}(t) &= I - \rho_{m-1}(t) y_{m-1}(t) s_{m-1}(t)^T.
\end{aligned}$$

to compute x_m at t_m. Beyond this point we save only m previous values of x. The definition of H is now, for $m \leq k$,
for $t_k \leq t$,

$$\begin{aligned}
&H(x(t), x(t_k), \dots, x(t_{k-m+2}), x(t_{k-m+1})) \\
&\quad := H_{k+1}(x(t), x(t_k), \dots, x(t_{k-m+2}), x(t_{k-m+1})) \\
&\quad := V_k(t)^T V_{k-1}(t_k)^T \dots V_{k-m+2}(t_{k-m+3})^T \\
&\qquad \cdot V_{k-m+1}(t_{k-m+2})^T H_0 V_{k-m+1}(t_{k-m+2}) \\
&\qquad \cdot V_{k-m+2}(t_{k-m+3}) \dots V_{k-1}(t_k) V_k(t) \\
&\qquad + V_k(t)^T V_{k-1}(t_k)^T \dots V_{k-m+2}(t_{k-m+3})^T \\
&\qquad \cdot \rho_{k-m+1}(t_{k-m+2}) s_{k-m+1}(t_{k-m+2}) \\
&\qquad \cdot s_{k-m+1}(t_{k-m+2})^T V_{k-m+2}(t_{k-m+3}) \dots \\
&\qquad \cdot V_{k-1}(t_k) V_k(t) \\
&\qquad + \dots \\
&\qquad + V_k(t)^T \rho_{k-1}(t_k) s_{k-1}(t_k) \\
&\qquad \cdot s_{k-1}(t_k)^T V_k(t) \\
&\qquad + \rho_k(t) s_k(t) s_k(t)^T \qquad (13B)
\end{aligned}$$

where

$$\begin{aligned}
s_k(t) &= x(t) - x(t_k), \\
y_k(t) &= \nabla f(x(t)) - \nabla f(x(t_k)), \\
\rho_k(t) &= 1/y_k(t)^T s_k(t), \\
V_k(t) &= I - \rho_k(t) y_k(t) s_k(t)^T.
\end{aligned}$$

It is obvious that the delayed ode (13) is a continuous version of the L-BFGS scheme. The $H = H_k$ in (13) attempts to approximate the inverse of the Jacobian in the Newton method. It is worth mentioning that the matrix H_k is never computed explicitly. We only need to compute the R.H.S. of (13), i.e., the product of H_k and a vector.

2.3 Uniqueness property of dynamical systems

The positive definite property of the dynamical system (13) is proved in Appendix I, hence we conclude that the solution exists. The next step is to show the Lipschitz continuity of this system, which implies that the solution is also unique.

2.3.1 Definition of Lipschitz continuity

Let $F : R^n \rightarrow R^m$ be a function, we say that F is **Lipschitz continuous** with **Lipschitz constant** L if there is a nonnegative constant L such that

$$\|F(x_1) - F(x_2)\| \leq L\|x_1 - x_2\|$$

for all x_1 and x_2 in R^m.

2.3.2 Lipschitz continuity of method (13)

By Theorem 2.1, the solution of our new unconstrained optimization method (13) is unique if the right-handed-side of (13) is Lipschitz continuous. We firstly rewrite (13) as

$$\frac{dx(t)}{dt} = -H(x(t), x(t-\tau))\nabla f(x(t)) \ .$$

Let $u = x(t)$ and $w = x(t-\tau)$, then our aim is to prove that $-H(u, w)\nabla f(u)$ is Lipschitz continuous with respect to u and w. In other words, we want to prove that

$$\|H(u,w)\nabla f(u) - H(\bar{u},w)\nabla f(\bar{u})\| \leq L_1\|u - \bar{u}\|, \quad (14)$$

$$\|H(u,w)\nabla f(u) - H(u,\bar{w})\nabla f(u)\| \leq L_2\|w - \bar{w}\|. \quad (15)$$

This problem is a difficult one and the following theorem may give some hint to our goal:

Theorem 2.5 *(Kurdyka, subanalytic, semi-algebraic) Let F: $X \subset R^n \to R$ be a definable C^1-function such that*

$$\big|\partial F/\partial x_i\big| < M$$

for some M and each i.
Then there exist a finite partition of X and $C > 0$ such that on each piece, the restriction of F to this piece is C-Lipschitz.
Moreover, this finite partition only depends on X and not on F. (And C only depends on M and n.)

It is obvious that $H(u,w)\nabla f(u)$ is a column vector. In order to transform a vector into a scalar we can use $e_i^T H(u,w)\nabla f(u)$ and problem (14) and (15) can be converted to

$$\Big|\frac{\partial}{\partial u}[e_i^T H(u,w)\nabla f(u)]\Big| < M_1,$$

$$\Big|\frac{\partial}{\partial w}[e_i^T H(u,w)\nabla f(u)]\Big| < M_2$$

Equation (13B) shows that $H(u,w)$ depends in a complicated way on

$$y(u,w) = \nabla f(u) - \nabla f(w) \ ,$$

$$s(u,w) = u - w \ .$$

The resulting partial derivatives $\frac{\partial}{\partial w}[e_i^T H\nabla f(u)]$ contain so many fractions whose denominators all have terms $y^T s$ that the bound must be controlled properly because the denominators tend to 0. The main problem is therefore to make the numerators and denominators have the same order so that they canceled out each other.

The following lemma seems to play a critical role in this cancellation:

Lemma 2.6 *Let $F : R^n \to R^m$ be continuously differentiable in the open convex set $D \subset R^n$, $x \in D$, and let $J = \frac{\partial F}{\partial x}$ be Lipschitz continuous at x in the neighborhood D, using a vector norm and the induced matrix operator norm and the Liptschitz constant γ. Then, for any $x + p \in D$,*

$$\|F(x+p) - F(x) - J(x)p\| \leq \frac{\gamma}{2}\|p\|^2.$$

If we define $x = w, p = u - w = s, F(x) = \nabla f(x)$, then the lemma converts to

$$\|\nabla f(u) - \nabla f(w) - \nabla^2 f(w)(u - w)\| \leq \frac{\gamma}{2}\|u - w\|$$

Substituting y and s into the above inequality yields

$$\|y - \nabla^2 f(w)s\| \leq \frac{\gamma}{2}\|s\|.$$

3 Numerical testing

Based on the analysis in the previous sections, we hope to conclude that the continuous-time Limited Memory BFGS method has a better performance than other traditional methods. In this section we will show the computational results of several examples.

3.1 The test problems

We have tested method (13) on the following unconstrained optimization problems:
Problem 1: Extended Rosenbrock function

$$f(x) = \sum_{i=1}^{n}[100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2],$$
$$[x_0]_{2i-1} = -1.2, [x_0]_{2i} = 1, [x^*]_i = 1, f(x^*) = 0.$$

Problem 2: Penalty function I

$$f(x) = \sum_{i=1}^{n} 10^{-5}(x_i - 1)^2 + [(\sum_{i=1}^{n} x_i^2) - \frac{1}{4}]^2,$$
$$[x_0]_i = i.$$

Problem 3: Variable dimensioned function

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 + [\sum_{i=1}^{n} i(x_i - 1)]^2,$$

$$+[\sum_{i=1}^{n} i(x_i - 1)]^4,$$

$$[x_0]_i = 1 - i/n, [x^*]_i = 1, f(x^*) = 0.$$

Problem 4: Linear function-rank 1

$$f(x) = \sum_{i=1}^{m}[i(\sum_{j=1}^{n} jx_j) - 1]^2, (m \geq n)$$

$$[x_0]_i = \frac{1}{i}, f(x^*) = \frac{m(m-1)}{2(2m+1)} \text{ at any point}$$

$$\text{where } \sum_{j=1}^{n} jx_j = \frac{3}{2m+1}.$$

3.2 Comparison between continuous-time L-BFGS and continuous-time steepest descent

Our main platform of numerical computation is Matlab. The Matlab library contains nonstiff ode solvers ode113, ode23, and ode45, and stiff ode solvers ode15s, ode23s, and ode23tb. There is also a nonstiff delayed differential solver ddesd. Nonstiff solvers are efficient for ode problems without a wide-spread spectrum of eigenvalues, whereas stiff solvers are good for problems with both large and small eigenvalues.

The difficulty with solving (13) is the time delay. In order to get familiar with the ode solvers, and eventually to take into account the delayed equation, we first test ode (and not dde) solvers on the above test problems.

In the first phase we start with solving a modified form of Problem 1:

Modified Extended Rosenbrock function:

$$f(x) = \sum_{i=1}^{n}[100(x_{2i} - x_{2i-1})^2 + (1 - x_{2i-1})^2],$$

$$[x_0]_{2i-1} = -1.2, [x_0]_{2i} = 1, [x^*]_i = 1, f(x^*) = 0.$$

The difference between the original Rosenbrock and the modified one is the square of x_{2i-1} in the first part of the right-handed-side.

We use the nonstiff ode solver ode113 on this problem with dimension $n = 100$ and tolerance $= 10^{-4}$. The result is given in the following table, where t denotes the iteration time, *value* denotes the computed optimal solution value, and *step* denotes the number of iterations.

Table 2: **Modified Rosenbrock Problem using ode113**

	t	value	step
L-BFGS	2	0	497
Steepest descent	23.2813	0.0006	53557

We can see that the continuous-time L-BFGS is obviously faster than the continuous-time steepest descent method in the value of t and number of integration steps.

Based on the performance in phase 1, we then move on to the second phase, where the delayed equation solver ddesd is used. This time we focus on all the test problems mentioned above and also use the original Extended Rosenbrock. However, the numerical results are not so good as we have expected – the continuous-time steepest descent method shows better performance than the continuous-time L-BFGS. After an analysis on the problems, it is found that many of them have a wide spectrum of eigenvalues. In other words, the problems are stiff.

In the third phase, the Matlab stiff ode solver ode15s is being used on the above four problems. Matlab does not have a stiff solver, and hence we are unable to take into account the effect of time delay. In the following tables, P denotes the problem number, N the dimension of variables. Table 4 to Table 6 give the performance on function value when the solution reach to the optimum value.

Table 3: **Comparison of the two methods for $m = 2$ on function value**

P	*N*	*Steepest descent*	*Limited-Memory BFGS*
1	10^3	0	5.8852×10^3
2	10^3	1.2030×10^5	9.7257×10^{-3}
3	10^3	5.2945×10^{-4}	42.307
4	10^3	6.0317	1.3153×10^{23}

Table 4: **Comparison of the two methods for $m = 4$ on function value**

P	N	*Steepest descent*	*Limited-Memory BFGS*
1	10^3	0	4.2692×10^3
2	10^3	1.5114×10^5	9.7304×10^3
3	10^3	0	2.9868×10^{-1}
4	10^3	6.0317	32.595

Table 5: **Comparison of the two methods for $m = 6$ on function value**

P	N	*Steepest descent*	*Limited-Memory BFGS*
1	10^3	0	1.1273×10^3
2	10^3	1.5068×10^5	9.7296×10^{-3}
3	10^3	7.1928×10^{-8}	1.0795×10^{-5}
4	10^3	6.0317	6.0317

Table 7 to Table 9 focus on the comparison for the norm of gradient as solution tends to the optimum value.

Table 6: **Comparison of the norm of gradient of the two methods for $m = 2$**

P	N	*Steepest descent*	*Limited-Memory BFGS*
1	1000	0	57.060
2	1000	1.4146×10^3	6.9366×10^{-5}
3	1000	4.6056×10	3.1173
4	1000	8.5570×10^{-7}	2.8544×10^{15}

Table 7: **Comparison of the norm of gradient of the two methods for $m = 4$**

P	N	*Steepest descent*	*Limited-Memory BFGS*
1	1000	0	6.5250
2	1000	1.6787×10^3	1.8992×10^{-4}
3	1000	0	8.1280×10
4	1000	2.1221×10^{-7}	1.9531×10^2

Table 8: **Comparison of the norm of gradient of the two methods for $m = 6$**

P	N	*Steepest descent*	*Limited-Memory BFGS*
1	1000	0	3.1648
2	1000	1.6749×10^3	1.7542×10^{-4}
3	1000	5.3639×10^{-1}	1.5238
4	1000	3.7336×10^{-7}	3.7253×10^{-5}

By comparing these tables one may find that the continuous-time L-BFGS performs better in Problem 2, while the continuous-time steepest descent method performs better in Problem 4. In Problems 1 and 3, there is not much difference.

In order to show the full potential of the continuous-time L-BFGS, it is our next goal to use a stiff dde solver, which will be performed in the next phase of this research.

3.3 A new code using Radar5

For stiff problems, including differential-algebraic and neutral delay equations with constant or state-dependent (eventually vanishing) delays, the code RADAR5 written by Ernest Hairer is more appropriate. This code uses collocation methods based on Radau nodes and solves DDE problems.

Consider initial value problems for delay differential equations:

$$My'(t) = f\left(t, y(t), y(\alpha_1(t, y(t))), \ldots, y(\alpha_m(t, y(t)))\right)$$

$$y(t_0) = y_0,\ y(t) = g(t) \quad \text{for} \quad t < t_0,$$

where M is a constant $d \times d$ matrix and $\alpha_i(t, y(t)) < t$ for all $t \geq t_0$ and for all i. The value $g(t_0)$ may be different from y_0, allowing for a discontinuity at t_0. (Collocation methods have been proved to have excellent stability properties for delay equations.)

Since the matrix M is singular, the above formulation includes all kinds of differential-algebraic delay equations. And if we introduce a new variable $z(t) = y'(t)$ for the equation. Actually this problem equals to

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y'(t) \\ z'(t) \end{pmatrix} = \left(\begin{array}{r} z(t) \\ -z(t) + f(t, y(t), y(\alpha(t, y(t))), z(t), z(\alpha \end{array} \right.$$

Next mission is to explain how to apply implicit Runge-Kutta methods to solve delay differential equation, and here mainly focus on collocation methods based on Radau nodes. For delay equations we hope to have a stable solution especially excellent stability. In this sense, collocation methods based on Radau nodes have been successfully applied to stiff ordinary differential equations. Hence to solve stiff problems like the above equation, we use radar5 program which is based on collocation methods. Radau IIA methods are implicit Runge-Kutta methods, whose coefficient matrix $A = (a_{ij})$ is invertible. In order to represent DDE simple enough, only one lag term is involved so that we rewrite the problem:

$$My'(t) = f(y(t), y(\alpha(t, y(t)))), \tag{16}$$

where $\alpha(t, y(t)) \leq t$, $y(t_0) = y_0$, and $y(t) = g(t)$ for $t < t_0$.

To implement Radau IIA methods to DDE problems consider a mesh $t_0 < t_1 < t_2 \ldots$, and denote the stepsize by $h_n = t_{n+1} - t_n$, and also denote $c_i = \sum_j a_{ij}$. Suppose approximations $y_n \approx y(t_n)$ and only in this condition can we apply Radau IIA methods to the problem (16), we have

$$M(Y_i^{(n)} - y_n) = h_n \sum_{j=1}^{s} a_{ij} f(Y_j^{(n)}, Z_j^{(n)}), y_{n+1} = Y_s^{(n)}, \tag{17}$$

where $Z_i^{(n)}$ is a suitable approximation to $y(\alpha_i^{(n)})$ with

$$\alpha_i^{(n)} = \alpha(t_n + c_i h_n, Y_i^{(n)})$$

We put

$$Z_i^{(n)} = \begin{cases} g(\alpha_i^{(n)}) & \text{if } \alpha_i^{(n)} < t_0 \\ u_m(\alpha_i^{(n)}) & \text{if } \alpha_i^{(n)} \in [t_m, t_{m+1}], \end{cases}$$

where $u_m(t)$ is a polynomial approximation of the solution $y(t)$ on the interval $[t_m, t_{m+1}]$. It is obvious to use collocation polynomial for u_m, which is of degree s and passes through the values y_m, and $Y_i^{(m)}$ for $i = 1, \ldots, s$.

Order of convergence:

Since we allow the matrix M to be singular, it is obvious that M is invertible. And also if the delay is larger than the stepsize, which means that $t - \alpha(t, y(t)) \geq h$, it follows from the theory for ODE that the local error at mesh points is $O(h^{2s})$. For the state-dependent delay case, this gives an additional $O(h^{s+2})$ contribution to the local error. For stiff problems, a further order reduction to $O(h^s)$ is possible, which is in complete analogy to stiff ordinary differential equations. On the other hand if the delay is smaller than the stepsize, the theory for ODE can no longer be applied, and a more complicated analysis is necessary.

Breaking points:

During the calculation, we find that discontinuities will occur in different orders of the derivative of the solution, no matter what the regularity of the right hand side of the DDE problems is. For the case that the solution is not smooth at t_0, which means left-hand derivative of the solution at t_0 is different from the corresponding right-hand derivative, the discontinuity at t_0 may propagate along the integration interval by means of the deviating argument $\alpha(t, y(t))$. Such discontinuity points are referred as *breaking points.*

Detection of the breaking point

To detect the breaking point we find the value of t to make the function $d(t) = \alpha(t, u(t)) - \zeta$ zero, where ζ is a previous breaking point and $u(\cdot)$ a suitable continuous approximation to the solution.

Algorithm 1

1). Assume that the step $[t_n, t_n + \bar{h}_n]$ is not accepted;

2). Look for zeros if the functions

$$d_i(s) = \alpha(s, u_{n-1}(s)) - \zeta_i$$

for $s \in [t_n, t_n + \bar{h}_n]$ and for all previously computed breaking points $\zeta_i (i = 1, 2, \ldots)$;

3). Let $\hat{\imath}$ such that $d_{\hat{\imath}}(t_n) \cdot d_{\hat{\imath}}(t_n + \bar{h}_n) < 0$;

The breaking point will then be close to a zero of $d_{\hat{\imath}}(s)$. We indicate it by $\hat{\xi}$, that is

$$\alpha(\hat{\xi}, u_{n-1}(\hat{\xi})) - \zeta_{\hat{\imath}} = 0.$$

Computation of the breaking point

After a breaking point is detected next is to calculate it to the desired accuracy.

Algorithm 2

4). Solve equations (17) with

$$\alpha(t_n + h, u_n(t_n + h)) - \hat{\zeta} = 0$$

with respect to the unknowns Y and h.

5a). **If** the step is accepted, which means the estimated local error is smaller than the required error tolerance, the point $\xi = t_n + h$ is added to the set of computed breaking points;

5b). **Otherwise** the stepsize is reduced according to the classical stepsize selection strategy.

Notice that here the stepsize of Runge-Kutta methods is not a constant but is variable; this allows for an accurate computation of the breaking point to the discrete order p of the method.

In order to discuss the property of DDE problem, here we introduce a theorem refer to R-K method:

Theorem 3.1 *Consider the DDE*

$$\begin{cases} y'(t) = f(t, y(t), y(t - \tau(t, y(t)))), & t_0 \lessdot t \lessdot t_f, \\ y(t) = \phi(t) \end{cases}$$

where $f(t, y, x)$ is C^p-continuous in $[t_0, t_f] \times R^d \times R^d$, the initial function $\phi(t)$ is C^p-continuous and the delay $\tau(t, y)$ is C^p-continuous in $[t_0, t_f] \times R^d$. Moreover, assume that the mesh $\Delta = \{t_0, t_1, \ldots, t_n, \ldots, t_N = t_f\}$ includes all hte discontinuity points of order$\lessdot p$ lying in $[t_0, t_f]$. If the underlying CRK method has discrete order p and uniform order q, then the DDE method has discrete global order and uniform global order $q' = min\{p, q + 1\}$; that is

$$\max_{1 \lessdot n \lessdot N} \|y(t_n) - y_n\| = O(h^{q'})$$

and

$$\max_{t_o \lessdot t \lessdot t_f} \|y(t) - \eta(t)\| = O(h^{q'}),$$

where $h = max_{1 \lessdot n \lessdot N} h_n$.

Suppose the CRK method has discrete order p and uniform order q, then we can either be satisfied with a DDE method with, possibly lower, uniform global order $q' = min\{p, q + 1\}$, or increase the uniform order of the underlying interpolant up to at least $p - 1$ in order to preserve the uniform global order p.

When comes to this theorem, we find a crucial condition for solving this kind of DDE problem, that is the continuous property of the initial function $\phi(t)$. In theorem 3.1 this is a fundamental request for the conclusion. In previous experiments we didn't pay much attention to $\phi(t)$, therefore the results are not good enough to show the advantage of L-BFGS approach compare to the original methods. In this case the next step for solving DDE problem is to find the suitable initial function $\phi(t)$ to satisfy with the conditions in Theorem 3.1.

4 Main stages of this research

(A) Prove that the function H in (13) is positive definite. (Done and shown in Appendix I).
(B) Prove that H is Lipschitz continuous. (Still ongoing)
(C) Show that the solution to (13) is asymptotically stable. (A simple consequence of (B).)
(D) Show that (13) has a better rate of convergence than the dynamical system based on the steepest descent direction.
(E) Perform numerical testing.
(F) Apply this new optimization method to practical problems.

APPENDIX I: To show that H in (13) is positive definite.

Without loss of ambiguity, in the subsequent proof, we drop the $t, t_0, t_1, \ldots, t_k, t_{k+1}$, ect. in $s_k(t), y_k(t), V_k(t)$, and so on below.

Property 1 If H_0 is positive definite, the matrix H defined by (13) is positive definite (provided that $y_i^T s_i > 0$ for all i).

Proof: We prove the result by induction. From the above discussion we know that (13), the continuous analog of the L-BFGS formula, has two cases. Hence our proof needs to cater for each of them.

For the first case $k+1 > m$, note that when $m = 1$

$$\begin{aligned} H_{k+1} &= V_k^T H_0 V_k + \rho_k s_k s_k^T \\ &= H_0 - \frac{H_0 y_k s_k^T}{y_k^T s_k} - \frac{s_k y_k^T H_0}{y_k^T s_k} \\ &\quad + \frac{s_k y_k^T H_0 y_k s_k^T}{(y_k^T s_k)^2} + \frac{s_k s_k^T}{y_k^T s_k} \end{aligned}$$

It is obvious that the proof of p.d. of this matrix is the same as that of Theorem 1 in section 1.3. Therefore, H_{k+1} is p.d. when $m = 1$.

Now suppose they are true for $m = l$, we show that they are true for $m = l + 1$.

When $m = l$, we have (denoting H_{k+1} by H_{k+1}^l to emphasize $m = l$)

$$\begin{aligned}
H^l_{k+1} \quad = \quad & V_k^T V_{k-1}^T \cdots V_{k-l+1}^T H_0 V_{k-l+1} \cdots V_{k-1} V_k \\
& + \{ V_k^T \cdots V_{k-l+2}^T \rho_{k-l+1} s_{k-l+1} s_{k-l+1}^T \\
& \quad \cdot V_{k-l+2} \cdots V_k \\
& + V_k^T \cdots V_{k-l+3}^T \rho_{k-l+2} s_{k-l+2} s_{k-l+2}^T \\
& \quad \cdot V_{k-l+3} \cdots V_k \\
& \ \vdots \\
& + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} V_k \\
& + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\
& + \rho_k s_k s_k^T \}.
\end{aligned}$$

being positive definite. (There are $l+1$ terms in H^l_{k+1})
If $m = l+1$, from (13B)

$$\begin{aligned}
H^{l+1}_{k+1} \quad = \quad & V_k^T V_{k-1}^T \cdots V_{k-l}^T H_0 V_{k-l} \cdots V_{k-1} V_k \\
& + V_k^T \cdots V_{k-l+1}^T \rho_{k-l} s_{k-l} s_{k-l}^T \\
& \quad \cdot V_{k-l+1} \cdots V_k \\
& + \{ V_k^T \cdots V_{k-l+2}^T \rho_{k-l+1} s_{k-l+1} s_{k-l+1}^T \\
& \quad \cdot V_{k-l+2} \cdots V_k \\
& + V_k^T \cdots V_{k-l+3}^T \rho_{k-l+2} s_{k-l+2} s_{k-l+2}^T \\
& \quad \cdot V_{k-l+3} \cdots V_k \\
& \ \vdots \\
& + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} V_k \\
& + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\
& + \rho_k s_k s_k^T \}.
\end{aligned}$$

(There are $l+2$ terms in H^{l+1}_{k+1}.)

Comparing these two equations we find that the terms in curly braces are the same, and let

$$\begin{aligned}
(*) \quad = \quad & V_k^T \cdots V_{k-l+2}^T \rho_{k-l+1} s_{k-l+1} s_{k-l+1}^T \\
& \quad \cdot V_{k-l+2} \cdots V_k \\
& + V_k^T \cdots V_{k-l+3}^T \rho_{k-l+2} s_{k-l+2} s_{k-l+2}^T \\
& \quad \cdot V_{k-l+3} \cdots V_k \\
& \ \vdots \\
& + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} V_k \\
& + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\
& + \rho_k s_k s_k^T
\end{aligned}$$

Thus,

$$\begin{aligned}
H^l_{k+1} \quad = \quad & V_k^T V_{k-1}^T \cdots V_{k-l+1}^T H_0 V_{k-l+1} \cdots V_{k-1} V_k \\
& + (*)
\end{aligned}$$

$$\begin{aligned}
H^{l+1}_{k+1} \quad = \quad & V_k^T V_{k-1}^T \cdots V_{k-l}^T H_0 V_{k-l} \cdots V_{k-1} V_k \\
& + V_k^T \cdots V_{k-l+1}^T \rho_{k-l} s_{k-l} s_{k-l}^T \\
& \quad \cdot V_{k-l+1} \cdots V_k + (*) \\
= \quad & V_k^T V_{k-1}^T \cdots V_{k-l+1}^T (V_{k-l}^T H_0 V_{k-l} + \rho_{k-l} \\
& \quad \cdot s_{k-l} s_{k-l}^T) V_{k-l+1} \cdots V_{k-1} V_k + (*)
\end{aligned}$$

Since we have assumed that H^l_{k+1} is p.d., if we try to prove that H^{l+1}_{k+1} is also p.d., we should prove that $V_{k-l}^T H_0 V_{k-l} + \rho_{k-l} s_{k-l} s_{k-l}^T$ and H_0 have the same property, i.e., $V_{k-l}^T H_0 V_{k-l} + \rho_{k-l} s_{k-l} s_{k-l}^T$ is also p.d..

Now we move forward to prove that $V_{k-l}^T H_0 V_{k-l} + \rho_{k-l} s_{k-l} s_{k-l}^T$ is p.d..

From the proof before, we have the following conclusion:

Consider the formula

$$B = V_k^T A V_k + \rho_k s_k s_k^T \qquad k = 0, 1, \cdots$$

(The definition of V_k, ρ_k, s_k are the same as in the L-BFGS formula.) If we know A is a p.d. matrix, then B is also p.d..

In this sense, it is easy to know that $V_{k-l}^T H_0 V_{k-l} + \rho_{k-l} s_{k-l} s_{k-l}^T$ is p.d.. As we have assumed that H^l_{k+1} is p.d. when H_0 is p.d., we conclude that H^{l+1}_{k+1} is p.d..

For the second case $k+1 \leq m$,

$$\begin{aligned} H_{k+1} &= V_k^T V_{k-1}^T \cdots V_0^T H_0 V_0 \cdots V_{k-1} V_k \\ &\quad + V_k^T \cdots V_1^T \rho_0 s_0 s_0^T V_1 \cdots V_k \\ &\quad \vdots \\ &\quad + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} v_k \\ &\quad + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &\quad + \rho_k s_k s_k^T. \end{aligned}$$

We also use induction to prove H_{k+1} is p.d..

Firstly, $H_1 = V_0^T H_0 V_0 + \rho_0 s_0 s_0^T$ from above, so it is clearly that H_1 is p.d..

Secondly, assumed that H_k is p.d.. We are going to prove that H_{k+1} is also p.d.. We have assumed

$$\begin{aligned} H_k &= V_{k-1}^T V_{k-2}^T \cdots V_0^T H_0 V_0 \cdots V_{k-2} V_{k-1} \\ &\quad + V_{k-1}^T \cdots V_1^T \rho_0 s_0 s_0^T V_1 \cdots V_{k-1} \\ &\quad + V_{k-1}^T \cdots V_2^T \rho_1 s_1 s_1^T V_2 \cdots V_{k-1} \\ &\quad \vdots \\ &\quad + V_{k-1} \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^T. \end{aligned}$$

is p.d.. Hence,

$$\begin{aligned} H_{k+1} &= V_k^T V_{k-1}^T \cdots V_0^T H_0 V_0 \cdots V_{k-1} V_k \\ &\quad + V_k^T \cdots V_1^T \rho_0 s_0 s_0^T V_1 \cdots V_k \\ &\quad \vdots \\ &\quad + V_k^T V_{k-1}^T \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} v_k \\ &\quad + V_k \rho_{k-1} s_{k-1} s_{k-1}^T V_k \\ &\quad + \rho_k s_k s_k^T \\ &= V_k^T (V_{k-1}^T V_{k-2}^T \cdots V_0^T H_0 V_0 \cdots V_{k-2} V_{k-1} \\ &\quad + V_{k-1}^T \cdots V_1^T \rho_0 s_0 s_0^T V_1 \cdots V_{k-1} \\ &\quad + V_{k-1}^T \cdots V_2^T \rho_1 s_1 s_1^T V_2 \cdots V_{k-1} \\ &\quad + \cdots + V_{k-1} \rho_{k-2} s_{k-2} s_{k-2}^T V_{k-1} \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^T) V_k + \rho_k s_k s_k^T \\ &= V_k^T H_k V_k + \rho_k s_k s_k^T. \end{aligned}$$

Therefore, we have proved that H_{k+1} is p.d. when $k+1 \leq m$.

So we have proved the property for both cases $k+1 \leq m$ and $k+1 > m$. □

The proof of the property above is part of our work on the delayed dynamical systems approach for unconstrained optimization. The requirement that $y_i^T s_i > 0$ for all i in Property 1 is not a major issue, because we work with a continuous ode and the numerical method can always be restarted.

References

[1] Aluffi-Pentini, F., Parisi, V. and Zirilli, F. Algorithm 617 DAFNE: A differential equations algorithm for nonlinear equations, ACM Trans. on Math. Software, 10 (3), 1984, 317-324.

[2] Boggs, P. T., The solution of nonlinear systems of equations by A-stable integration techniques, SIAM J. Numer. Anal. 8 (4), 1971, 767-785.

[3] Botsaris, C. A. and Jacobson, D. H., A Newton-type curvilinear search method for optimization, JMAA, 54, 1976, 217-229.

[4] Brown, A. A. and Bartholomew-Biggs, M. C., ODE versus SQP methods for constrained optimization, JOTA 62 (3), 1989, 371-386.

[5] Chen, Y. H. and Fang, S. C., Solving convex programming problem with equality constraints by neural networks, Computers Math. Appl. 36, 1998, 41-68.

[6] Chu, M. T., On the continuous realization of iterative processes, SIAM Review 30 (3), 1988, 375-387.

[7] Han, Q., Liao, L.-Z., Qi, H. and Qi, L., Stability analysis of gradient-based neural networks for optimization problems, J. Global Optim. 19 (4), 1962, 363-381.

[8] Hassan, N. and Rzymowski, W., An ordinary differential equation in nonlinear programming. Nonlinear Analysis, Theory, Method & Applications 15 (7), 1990, 597-599.

[9] aykin, S. S., Neural Networks: A Comprehensive Foundation, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[10] He, B. S., Inexact implicit methods for monotone general variational inequalities, Mathematical Programming, 86 (1), 1999, 199-217.

[11] Incerti, S., Parisi, V. and Zirilli, F., A new method for solving nonlinear simultaneous equations, SIAM J. Numer. Anal. 16, 1979, 779-789.

[12] Jorge Nocedal, Updating Quasi-Newton Matrices With Limited Storage, Mathematics of Computation, Vol 35, No 151, July 1980, pp. 773-782.

[13] Liao, L.-Z. and Qi, H., A neural network for the linear complementarity problem, Math. Comput. Modelling 29 (3), 1999, 9-18.

[14] L.Z. Liao, L.Q. Qi, and H.W. Tam, A gradient-based continuous method for large-scale optimization problems, Journal of Global Optimization, Vol 31 , Apr 2005, pp. 271-286.

[15] Liao, L.-Z., Qi, H. and Qi, L., Solving nonlinear complementarity problems with neural networks: a reformulation method approach, JCAM 131 (12), 2001, 343-359.

[16] LI-ZHI LIAO, HOUDUO QI and LIQUN QI, (2004), Neurodynamical Optimization, Journal of Global Optimization 28: 175-195.

[17] Novakovi?c, Z. R., Solving systems of non-linear equations using the Lyapunov direct method, Computers Math. Applic. 20 (12), 1990, 19-23.

[18] Tanabe, K., A geometric method in nonlinear programming, JOTA 30 (2), 1980, 181-210.

[19] Tank, D. W. and Hopfield, J. J., Simple neural optimization networks: An A/D convert, signal decision circuit, and a linear programming circuit, IEEE Trans. Circuits Syst. 33, 1986, 533-541.

[20] Wilde, N. G., A note on a differential equation approach to nonlinear programming, Management Science 15 (11), 1969, 739-739.

[21] Williems, J. L., Stability Theory of Dynamical Systems, Nelson, 1970.

[22] Wu, X., Xia, Y., Li, J. and Chen, W. K., A high performance neural network for solving linear and quadratic programming problems, IEEE Trans. Neural Networks, 7, 1996, 643-651.

[23] Xia, Y., A new neural network for solving linear programming problems and its applications, IEEE Trans. Neural Networks 7, 1996, 525-529.

[24] Yamashita, H., A differential equation approach to nonlinear programming, Math. Prog. 18,1980, 155-168.

[25] Zabcayk, J ., Mathematical Control Theory: An Introduction, Birkhauser, Boston, 1992.

Speeding up K-Means Algorithm by GPUs

You Li, Kaiyong Zhao, Xiaowen Chu
Department of Computer Science
Hong Kong Baptist University
{youli, kyzhao, chxw}@comp.hkbu.edu.hk

Abstract

Clustering algorithm is always facing the efficiency challenge due to the continuously fast increasing data volume. Exploiting parallel computing is one of the most promising solutions. In this paper, we conduct systematic research on paralleling the most important clustering algorithm k-Means on GPUs. We find that data dimension is an important parameter that should be taken into consideration when designing the parallel algorithms. Particularly, two algorithms have been designed for low and high dimension data respectively to make the best use GPUs. For low dimension data, we mainly utilize GPU registers to decrease data access latency. For high dimension data, we and then design a novel algorithm which simulates matrix multiplication and exploits GPU shared memory to achieve high compute to global memory access ratio. As a result, our GPU-based k-Means algorithm is four to ten times faster than the best reported GPU-based algorithm.

I. Introduction

Clustering is a method of unsupervised learning that partitions a set of records into clusters, such that intra-cluster similarity is maximized while inter-cluster similarity is minimized [1, 2]. The *k*-Means algorithm is one of the most popular clustering algorithms [3] and is widely used in many fields such as statistical data analysis, pattern recognition, image analysis and bioinformatics [4, 5]. The running time of *k*-Means algorithm grows with the increase of the data size and data dimension. Hence clustering large-scale datasets is usually a time-consuming task. Parallelizing *k*-Means is a promising approach to overcoming the challenge of the huge computational requirement [6-8]. In [6], P-CLUSTER uses a client-server model, in which a server process partitions data into blocks and sends the initial centroid list and blocks to each of clients. P-CLUSTER has been further enhanced by pruning as much computation as possible while preserving the clustering quality [7]. In [8], the *k*-Means clustering algorithm has been parallelized by exploiting the inherent data-parallelism and utilizing message passing.

Recently, as a general-purpose and high performance parallel hardware, Graphics Processing Units (GPUs) develop continuously, and supply another platform for parallelizing *k*-Means. GPUs are dedicated hardware for manipulating computer graphics. Due to the huge computing demand for real-time and high-definition 3D graphics, the GPUs have evolved into highly parallel many-core processors. The advances of computing power in GPUs have driven the development of general-purpose computing on GPUs (GPGPU). In this paper, we use a general-purpose parallel programming model, namely Compute Unified Device Architecture (CUDA) [9, 10] to implement our parallel *k*-Means algorithm.

CUDA has been used for speeding up a large number of applications [11, 12]. Some clustering algorithms have also been implemented on the GPUs, including *k*-Means. There are three main GPU-based *k*-Means algorithms: *GPUMiner* [13], *UV_k-Means* [14], and *HP_k-Means* [15]. *UV_k-Means* achieves a speedup of ten to forty as compared with a four-threaded *Minebench* [16] running on a dual-core, hyper-threaded CPU. *HP_k-Means* claims another speedup of two to four compared with *UV_k-Means* and twenty to seventy speedup compared with *GPUMiner*. Obviously, those works reveal the high performance advantage of the GPU. However, they only investigated some general optimization rules and utilized parts. Therefore, it is still worth analyzing how to apply the optimization rules in the design of the algorithm and how to better utilize the GPU.

Thus, in this paper, we conduct systematic research on paralleling the *k*-Means using CUDA and optimizing the algorithm in detail. Particularly, considering the character of data dimension, we design two strategies for low and high dimension data respectively. For low dimension data, we adopt a relatively simple workflow and mainly utilize the register to achieve a low global memory access times and latency. For high dimension data, we present a novel idea on the relationship between *k*-Means and Matrix multiplication, and design a shared memory based *k*-Means algorithm. The experiment shows that our *k*-Means compares very favorably with *GPUMiner*, *UV_k-Means* and *HP_k-Means*, and yet achieves a speedup of 100, 10 and 5 around respectively.

The paper is organized as follows: section II introduces the existing GPU-based *k*-Means algorithms; section III presents the design strategy and implementation of our *k*-Means algorithm; section IV presents our experimental results and compares our *k*-Means algorithm with existing ones. Section V concludes this paper.

II. Related work

To the best of our knowledge, there are mainly three GPU-based *k*-Means algorithms, *UV_k-Means*, *GPUMiner*, and *HP_k-Means*, the former two of which are open source. To well understand the GPU-based algorithm, we briefly introduce the architecture first.

A. The GPU architecture

We take NVIDIA GTX280 as an example to show the GPU architecture. GTX 280 has 30 Streaming Multiprocessors (SMs), and each SM has 8 Scalar Processors (SPs), resulting a total of 240 processor cores. The SMs have a Single-Instruction Multiple-Data (SIMD) architecture: At any given clock cycle, each SP executes the same instruction, but operates on different data. Each SM has four different types of on-chip memory, namely registers, shared memory, constant cache, and texture cache, as shown in Fig.1. Constant cache and texture cache are both read-only memories shared by all SPs, but with very limited size. Off-chip memories such as local memory and global memory have relatively long access latency, usually 400 to 600 clock cycles [10]. The properties of the different types of have been summarized in [10, 17]. In general, the scarce shared memory should be carefully utilized to amortize the global memory latency cost.

In CUDA model, the GPU is regarded as a coprocessor capable of executing a great number of threads in parallel. A single source program includes host codes running on CPU and also kernel codes running on the GPU. Compute-intensive and data-parallel kernel codes run on the GPU. The threads are organized into thread blocks, and each block of threads are executed concurrently on one SM. Threads in a thread block can share data through the shared memory and can perform barrier synchronization. But there is no synchronization mechanism for different thread blocks besides terminating the kernel. Another important concept in CUDA is *warp*, which is formed by 32 parallel threads and is the scheduling unit of each SM. When a warp stalls, the SM can schedule another warp to execute. A warp executes one instruction at a time, so full efficiency can only be achieved when all 32 threads in the warp have the same execution path. There are two consequences: first, if the threads in a warp have different execution paths due to conditional branch, the warp will serially execute each branch which increases the total time of instructions executed for this warp; secondly, if the number of threads in a block is not a multiple of warp size, the remaining instruction cycles will be wasted.

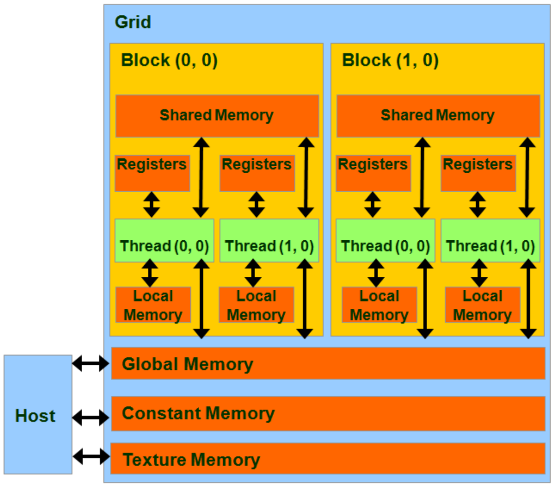


Figure 1. Hardware architecture of the GPU

B. UV_k-Means

In the *UV_k-Means*, in order to avoid the long time latency of global memory, they copy all the data to the texture, which has cache mechanism. Then, they use constant memory to store *k* centroids, which is also more efficient than global memory. The grid and block are organized as follows: each thread is responsible for one data point, finding the nearest centroid, and each block has 256 threads, so the grid has [*n*/256] blocks.

The work flow is straight forward: firstly, each thread calculates the distance from one corresponding data point to every centroid and finds the minimum distance and corresponding centroid. Secondly, each block calculates a temp centroid set based on several data points, and each thread calculates one dimension of the temp centroid. Thirdly, CPU copies the temporal centroid sets from the GPU to the CPU, and serially calculates the final new centroid set by adding the temporal centroid sets.

UV_k-Means has achieved a speed-up of twenty to forty through our experiment, mainly by assigning each data point to one thread and utilizing the cache mechanism to get a high reading efficiency. However, the efficiency still could be further improved by using another memory mechanism, shared memory, as well as considering not only one data point once a time, which are the two key points considered in this paper.

C. GPUMiner

GPUMiner puts all the input data in the global memory, and loads k centroids to the shared memory. Each block has 128 threads, and the grid has $n/128$ blocks. The workflow is also straight forward: firstly, each thread calculates the distance from one data point to every centroid, and changes the suitable bit into true in the bit array, which stores the nearest centroid for each data point; secondly, each thread is responsible for one centroid, finds all the corresponding data points from the bit array and takes the mean of those data points as the new centroids.

The main problem of *GPUMiner* is the utilization of memory in the GPU, since *GPUMiner* accesses most of the data (input data point) from global memory, which is obviously the slowest one, and thus results in a low efficiency. Besides, the main characteristic of *GPUMiner* is designing a bitmap-based algorithm, which makes it easy to find each data set. However, as *HP_k-Means* points out, bitmap approach is elegant in expressing the problem, but it is not a good method for performance, since bitmap takes more space when k is large and requires more shared memory. We will present its performance in detail in section IV.

III. Design and implementation

The k-Means algorithm is one of the most useful clustering methods. Given a set of n data points $R = \{r_1, r_2, ..., r_n\}$ in a d dimensional space, the task of k-Means is to partition R into k clusters ($k < n$) $S = \{S_1, S_2, ..., S_k\}$ such that $\sum_{i=1}^{k}\sum_{x_j \in S_i} \| x_j - \mu_i \|^2$ is minimized, where μ_i is the mean of S_i.

The k-Means algorithm iteratively partitions a given dataset into k clusters. It first selects k data points as the initial centroids. Then the algorithm iterates as follows: (1) Calculate the Euclidean distance between each pair of data point and the centroid; (2) Assign each data point to its closest centroid; (3) Calculate the new centroid by taking the mean of all the data points in each cluster. The iteration terminates when the changes in the centroids are less than some threshold or some given iteration time. The whole process is shown in Algorithm 1.

The computational complexity of a single round of k-Means is: O(nkd) in step (1), O(nk) in step(2), O(nd) in step (3). We mainly focus on speeding up step (1). Considering the parameter of data dimension d, we design two GPU-based algorithms for low and high dimension data respectively. For low dimension data, we propose to utilize register and combine Step (1) and Step (2) together. For high dimension data sets, we adopt the shared memory to parallelize Step (1) and apply the most efficient reduction method to speed up Step (2). Step (3) has a relatively low computational complexity of O(nd), and it is difficult to be fully parallelized due to write conflict. So we let GPU handle part of the task that is worthy to be performed on GPU, and then send the remaining part to CPU for execution.

Algorithm 1: CPU-based k-Means

```
// flag: shows whether it still needs to iterate;
// iter: the current round of iteration;
// Max_iter:  the maximum number of iterations;
// d(r, s): the distance between r and the cluster s;
while flag && iter <= Max_iter
    for each r in R and each s in S
        Compute d(r, s);
    Find the closest centroid based on the distance;
    Compute new centroids;
    if the changes of the centroids are less than threshol
        flag ← false;
    iter = iter + 1;
end of while
```

A. Finding closest centroid

The CPU-based algorithm of finding closest centroid is straightforward, as shown in Algorithm 2. Since the algorithm computes the distance between each data point and each centroid, our first method to parallelize Algorithm 2 is dispatching one data point to one thread, and then each thread calculates the distance from one data point to all the centroids, and maintains the minimum distance and the corresponding centroid, as shown in Algorithm 3. Line 1 and 2 show how the algorithm designs the block and gird; line 3 tells how to calculate the position of the corresponding data point for each thread in global memory; line 4-5 load the data point into the register; line 6-11compute the distance and maintain the minimum one.

It is worth pointing out that the key step of achieving high efficiency is loading the data point into the register, which ensures reading the data point from global memory only once when calculating the distances between the data point and k centroids. Obviously, reading from register is much faster than reading from global memory. The experiment in section IV shows the advantage of Algorithm 3 compared with the best published results. However, the problem of Algorithm 3 is the limited size of the register. In fact, users are not able to control the register right now, and could only utilize register when the data size is appropriate. When the data point cannot be loaded into the register as the dimension grows, it will be stored in local memory, which will increase the reading latency.

In fact, the input data point and the centroid could be viewed as two matrixes *data*[*n*][*d*] and *centroid*[*d*][*k*]; the result distance could be denoted as *d*[*n*][*k*]; and the distance computing process could be described as Algorithm 4, which shares the same flow as matrix multiplication. Based on this character, we design Algorithm 5 for high dimension data, adopting the idea of matrix operation and mainly utilizing the shared memory.

Algorithm 2: finding closest centroid based on CPU

```
//min_D: a temp variable, stores the minimum distance;
// index: stores the min centroid ID for each data point;
for $r_i$ in $R$
    for $s_j$ in $S$
        Compute $d(r_i, s_j)$;
        if $d(r_i, s_j) < min\_D$
            $min\_D \leftarrow d(r_i, s_j)$;
        index[$i$] ← $j$;
    end of for;
end of for;
```

Algorithm 3: finding closest centroid based on the register of the GPU

```
// threadDim: the dimension of the thread in each block;
// blcokDim: the dimension of the block in each grid;
// blockIdx.x: the current block ID;
// threadIdx.x: the current thread ID;
// data: the address of R;
// i: the ID of data point;
// Gdata: the address of the corresponding data point;
//S: the set of the centroid;
threadDim ← 16x16;
blockDim ← n/256;
$i \leftarrow blockIdx.x \times blockDim + threadIdx.x \times threadDim$;
$Gdata \leftarrow data + i \times d$;
Load the data point from Gdata to the register.
for $s_j$ in $S$
    Compute $d(r_i, s_j)$; // read $r_i$ the register
    if $d(r_i, s_j) < min\_D$
        $min\_D \leftarrow d(r_i, s_j)$;
        index[$i$] ← $j$;
end of for;
```

Algorithm 4: distance computing

```
for $i$ from $1$ to $n$
    for $j$ from $1$ to $k$
        for $m$ from $1$ to $d$
            $d[i][j] \mathrel{+}= (data[i][m]-centroid[m][j])^2$;
        $d[i][j] \leftarrow \mathrm{sqrt}(d[i][j])$;
```

Algorithm 5: finding closest centroid based on the shared memory of the GPU

```
// THIGH: the high of the tile;
// TWIDTH: the width of the tile;
// thread: the dimensions of the block;
// grid: the dimensions of the grid;
// SMData: stores the tile in shared memory;
// TResult: stores the temp distance in shared memory;
// SR: stores the temp distance in global memory;
// Alast : is the upper bound address of data point;
1.  THIGH ← 32, TWIDTH ← 32;
2.  dim thread(THIGH, 2);
3.  dim grid (k/ TWIDTH, n/ THIGH);
4.  indexD points to the corresponding data;
5.  indexC points to the corresponding centroid;
6.  indexR points to the corresponding result;
7.  SMData[TWIDTH][THIGH] in shared memory;
8.  TResult[TWIDTH][ THIGH] in shared memory;
9.  Alast ← indexD + d;
10. do
11. {
12.     Load data from global memory to SMData;
13.     indexD is added by TWIDTH;
14.     Compute the temp distance;
15.     Add the temp distance in TResult ;
16. }while(indexD < Alast);
17. __syncthreads();
18. Write the minimum distance in TResult back to SR;
```

The main idea of Algorithm 5 is decreasing the global memory access times and latency by loading the data into the shared memory tile by tile. Thus, Algorithm 5 reads each data point from global memory only once, the same as Algorithm 3. The key point of Algorithm 5 is how to access the shared memory efficiently, which is achieved by adopting coalescing reading, accessing sixteen continuous address for the thread in a half warp to avoid the bank conflict. The details are described as follows.

As shown in Algorithm 5, each block is in charge of computing a sub result matrix *SR*[*TWIDTH*][*THIGH*]. Each block has *THIGH**2 threads, and each thread computes a column of *sr*. line 4-6 finds the right position of the data, the centroid and the result matrix for each thread; in the loop from line 11 to 16, the algorithm loads a tile data from global memory to the shared memory, and computes the temp distance saved in *TResult*; the loop ends when the whole row has been calculated, as shown in Fig.2.

After calculating the temp minimum distances from each sub matrix, we need to get the final minimum one. The computational complexity of CPU-based algorithm is O(*nk*). On GPU, if each data point is assigned to one thread, the computational complexity decreases to O(*k*). If each data point has *k* threads, the complexity will be O(log*k*). However, the cost of the parallel algorithm is (*k*/log*k*) and not efficient, which is measured by threads multiplying with complexity. In fact, according to Brent's theory, we can assign O(k/log$k$) threads for each data point, finding the minimum distance from a k array; each thread does O(log$k$) sequential work; then all O(k/log$k$) threads cooperate for O(log$k$) steps. And the computational complexity is O(log$k$) while the cost is O(k). This paper will not discuss the implementation in detail but adopt Brent's theory.

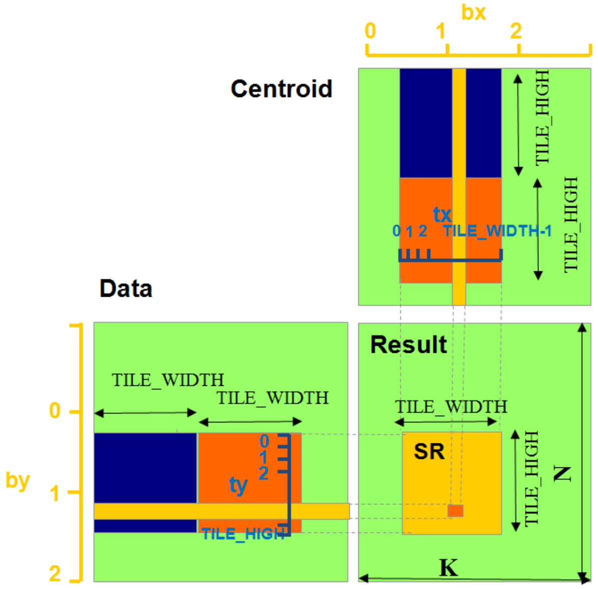


Figure 2 tile-based distance computing process

B. *Calculating the new centroid*

The result of finding the closest centroid is an array *index*[*n*], which stores the closest centroid for each data point. The data points belonging to the same centroid constitute one cluster. Calculating the new centroids is taking the mean of all the data points in each cluster. As shown in Algorithm 6, the computational complexity is O(nd+kd), and it is difficult to be fully parallelized. Since if we assign each data point to a thread, it will generate write conflict when adding the data to the shared centroid. On the other hand, if we assign each centroid to a thread, the computing power of the GPU is has limited utilization.

In this paper, we design an algorithm, which adopts the “divide and conquer” strategy: divide the data into groups; reduce each group and get temp centroids; then divide the temp centroids and reduce iteratively on the GPU until n' is smaller than M, which means the GPU has no advantage than the CPU for further computing; calculate the final centroids on the CPU, as shown in Algorithm 7. By dividing the data into groups, the write conflict decreases, since each group writes its own temp centroids and has no influence to other groups. And it is still a profitable method when we consider the additional data transfer through our experiments (see section IV). Besides, it is necessary to point out that M in Algorithm 7 line 1 should be the multiple of the number of SM, which can ensure a high schedule efficiency on the GPU.

```
Algorithm 6: CPU_based method
for Calculating the new centroids
// count: stores the number of data points in each clusters;
// new_cen: the address of the new centroid;
// data: is the address of data point set R;
for i from 1 to n
    ++count[ index[i] ];
    for j from 1 to d
        new_cen [index[i]][j] += data[i][j];
for i from 1 to k
    for j from 1 to d
        new_cen [i][j] /= count[i];
```

```
Algorithm 7: GPU_based method
for Calculating the new centroids
// n': is the number of groups to be divided;
1.  M is the multiple of the number of SM;
2.  n' ← n/M;
3.  Divide n data points in to n' groups;
4.  Compute n' temp centroids on the GPU;
5.  while n' > M
6.  {
7.      Divide n' temp centroids into n'/M groups;
8.      n' ← n'/M;
9.      Compute n' temp centroids on the GPU;
10. }
11. Reduce n' temp centroids into final centroids on CPU;
```

IV. Experiments

The experiments were conducted on a PC with an NVIDIA GTX280 GPU and an Intel(R) Core(TM) i5 CPU. GTX 280 has 30 SIMD multi-processors, and each one contains eight processors and performs at 1.29 GHz. The memory of the GPU is 1GB with the peak bandwidth of 141.7 GB/sec. The CPU has four cores running at 2.67 GHz. The main memory is 8 GB with the peak bandwidth of 5.6 GB/sec. We use Visual Studio 2008 to write and compile all the source code. The version of CUDA is 2.3. We calculate the time of the application after the file I/O, in order to show the speedup effect more clearly.

The experiments contain two parts: first, we compare our results with the best published results of *HP_k-Means*, which is mainly on low dimension data. Second, we compare our k-Means with *UV_k-Means* and *GMiner* on high dimension data. Each of the experiments is repeated ten times and the average results are reported.

A. *On low dimension data*

Here we choose exactly the same data sets with *HP_k-Means* as follows: *n* has two values, two million and four million; *k* has two values, one hundred and four hundred; *d* also has two values, two and eight. Each dimension is a floating point number, and generated randomly.

Table 1: Speed of k-Means on low dimension data

n	*k*	*d*	*Our* *k-Means*	*HP* *k-Means*	*UV* *k-Means*	*GPU* *Miner*
2 million	100	2	0.22	1.45	2.84	61.39
	400	2	0.64	2.16	5.96	63.46
	100	8	0.24	2.48	6.07	192.05
	400	8	0.65	4.53	16.32	226.79
4 million	100	2	0.31	2.88	5.64	130.36
	400	2	1.22	4.38	11.94	126.38
	100	8	0.42	4.95	12.85	383.41
	400	8	1.26	9.03	34.54	474.83

Note: the time is in second. The hardware environment of HP is as follows: NVIDIA GTX280 GPU; Intel Xeon CPU, 2.33GHz; 4GB memory.

As is shown in Table 1, our *k*-Means is the most efficient one among the four algorithms. It is four to ten times faster than the best published results: *HP_k-Means*, ten to twenty faster than *UV_k-Means* and one hundred to three hundred faster than *GPUMiner*. Since *HP* only says some optimization rules without publishing the source code. We mainly analyze the difference between our *k*-Means and *UV_k-Means*.

The workflows of the two algorithms are very similar: each thread finds the minimum centroid for each data point. The main difference is the memory utilization: *UV_k-Means* puts the data on the texture and puts the centroids on the constant; our *k*-Means firstly loads the data on the register, and reads the data from the register each time when calculating the distance from each centroid, resulting in a low global memory access times and latency, since reading from register is by far faster than reading from other memories.

Also shown in Table 1, our *k*-Means is insensitive with dimension, since the time differs a little when the dimension changes from two to eight, which also results from the utilization of the register. On the other hand, when *k* grows, the algorithm has to access the global memory more, which is proportional to the *k*.

Through our experiment, when the dimension is larger than sixteen, the data point cannot be loaded into the register (The compiling and building information of Visual Studio can indicate the memory that the program will use), and the speed decreases sharply because of accessing the local memory. So, we use Algorithm 5, shared memory based algorithm to deal with the high dimension data.

B. *On high dimension data*

Here we use the data from the KDD Cup 1999 [10], and choose two data sets, which have 51200 and 494080 data points. Each data point contains 34 features, and each one is floating point. We compare our algorithm with *GPUMiner* and *UV_k-Means*.

The results are shown in Table 2. Our *k*-Means is four to eight faster than *UV_k-Means*, ten to forty faster than *GPUMiner*, and one hundred to two hundred faster than a CPU based *k*-Means algorithm which is also developed by us, using Algorithm 1 and Algorithm 6. We also mainly analyze the difference between our *k*-Means and *UV_k-Means*.

Table 2: Speed of k-Means on high dimension data

Data set	Our *k*-Means	UV *k*-Means	GPU Miner	CPU *k*-Means
51200	0.43	1.86	4.26	35.79
494080	1.15	8.67	40.6	224.47

Table 3: Time distribution of our k-Means algorithm

Function/data set		Find the closest centroid	Compute new centroid
51200	GPU	0.07	0.16
	CPU	33.5	2.28
494080	GPU	0.87	0.18
	CPU	207.78	16.67

When dealing with high dimension data, larger than sixteen, our algorithm loads the data tile by tile into the shared memory. Thus it accesses the global memory only once for each data point. *UV_k-Means* adopts texture to store the data point and decreases the global memory reading latency. However, it depends on the cache mechanism, and if the cache missing grows, the efficiency would lower down. On the other hand, the shared memory could perform more stably.

As shown in Table 3, *finding the closest centroid* achieves a speedup of forty to two hundred compared with our CPU-based algorithm, while *computing new centroid* achieves a speedup around ten, which further prove the advantage of our algorithm.

V. Conclusions

In this paper, we proposed a GPU-based *k*-Means algorithm. It presents mainly two novel ideas: first, based on the dimension of the data, our *k*-Means algorithm chooses two different strategies. For low dimension data, our algorithm utilizes GPU registers, and achieves a speedup of four to ten than *HP_k-*

Means. For high dimension data, our algorithm firstly observes the connections of each data point, analyzes the relationship with matrix multiplication and reduction, adopts shared memory to avoid multiple accessing the global memory, increases the number of the computing operation for each global memory access, and achieves a speedup of four to eight as compared with *UV_k-Means*.

The algorithm presented in this paper could deal with a finite scale of data set, limited by the global memory size of the GPU. Consequently, when the data set is larger than it, new strategies have to be designed. A possible method is to divide the data set into several parts, each of which could be loaded into global memory of the GPU; find the closest centroid for each data point in each part; accumulate the data point to their corresponding centroid part by part and get the new centroids. It is worth pointing out that the above method could adopt the algorithm in this paper when dealing with each part. Another feasible approach is using the GPU cluster, which is a computer cluster and each node is equipped with a GPU, and conducted as follows: divide the data into several parts, each of which is assigned to one node; on each node, find the closest centroid for each data point, using Algorithm 5; adopt the divide and conquer strategy as Algorithm 7 to calculate the new centroid.

In summary, the results of this paper prove that adopting GPUs is a promising acceleration method for clustering algorithms to deal with large scale data. Moreover, the analysis method and optimization rules presented in this paper could also be applied in speeding up other data mining algorithms.

References

[1] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley Companion Book Site 2006.

[2] A. K. Jain and R. C. Dubes, Algorithms for clustering data, Prentice-Hall, 1988.

[3] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, “Top 10 algorithms in data mining,” knowledge information systems, vol. 14, pp. 1-37, 2008.

[4] X. Wang and M. Leeser, “K-means Clustering for Multispectral Images Using Floating-Point Divide,” in Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines: IEEE Computer Society, 2007.

[5] H. Zhou and Y. Liu, “Accurate integration of multi-view range images using k-means clustering,” Pattern Recogn., vol. 41, pp. 152-175, 2008.

[6] D. Judd, P. K. McKinley, and A. K. Jain, “Large-Scale Parallel Data Clustering,” in Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume IV-Volume 7472 - Volume 7472: IEEE Computer Society, 1996.

[7] D. Judd, P. K. McKinley, and A. K. Jain, “Large-Scale Parallel Data Clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, pp. 871-876, 1998.

[8] I. S. Dhillon and D. S. Modha, “A Data-Clustering Algorithm on Distributed Memory Multiprocessors,” in Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD: Springer-Verlag, 2000.

[9] NVIDIA CUDA: http://developer.nvidia.com/object/cuda.html.

[10] NVIDIA CUDA Compute Unified Device Architecture: Programming Guide, Version 2.0, June 2008.

[11] S. A. Manavski, “CUDA compatible GPU as an efficient hardware accelerator for AES cryptography,” In Proceedings of IEEE International Conference on Signal Processing and Communication, p. 4, Nov. 2007.

[12] S. Ryoo, C. I. Rodrigues, S. S. Baghsorkhi, S. S. Stone, D. B. Kirk, and W. Hwu, “Optimization principles and application performance evaluation of a multithreaded GPU using CUDA,” in Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming Salt Lake City, UT, USA: ACM, 2008.

[13] W. Fang, K. K. Lau, M. Lu, X. Xiao, C. K. Lam, P. Y. Yang, B. He, Q. Luo, P. V. Sande, and K. Yang, “Parallel Data Mining on Graphics Processors,” Technical Report HKUSTCS08, 2008.

[14] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, “A Performance Study of General-Purpose Applications on Graphics Processors Using CUDA,” Journal of Parallel and Distributed Computing, 2008.

[15] R. Wu, B. Zhang, and M. Hsu, “Clustering billions of data points using GPUs,” in UCHPC-MAW '09: Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop, Ischia, Italy, 2009, pp. 1-6.

[16] J. Pisharath, Y. Liu, W.-k. Liao, A. Choudhary, G. Memik, and J. Parhi, “NU-MineBench 2.0,” CUCIS Technical Report CUCIS-2005-08-01, Center for Ultra-Scale Computing and Information Security, Northwestern University, 2005.

[17] J. C. Shafer, R. Agrawal, and M. Mehta, “SPRINT: A Scalable Parallel Classifier for Data Mining,” VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, pp. 544-555, 1996.

Design and Implementation of Multiple-precision Integer Library for GPUs

Kaiyong Zhao
Department of Computer Science
Hong Kong Baptist University
Hong Kong, P.R.C
kyzhao@comp.hkbu.edu.hk

Xiaowen Chu
Department of Computer Science
Hong Kong Baptist University
Hong Kong, P.R.C
chxw@comp.hkbu.edu.hk

Abstract

Multiple-precision modular multiplications are the key components in security applications, like public-key cryptography for encrypting and signing digital data. But unfortunately they are computationally expensive for contemporary CPUs. By exploiting the computing power of the many-core GPUs, we implemented a multiple-precision integer library with CUDA. In the previous articles, there are some GPU application to accelerate the multiplication of large numbers and model multiplication. Under normal circumstances, our expression of Multiple-precision numbers are in accordance with the order of memory, but the GPU in the thread access case accessed only on a consolidated to achieve maximum bandwidth.. In this case, the traditional arrangement of large numbers method suited GPU demand, we consider the use of alignment on the way to access memory. Similarly, because we are dealing with the data, and there is no order in terms of special requirements, data encryption, or data encoding, we have the data interpreted as a two-dimensional matrix, the traditional approach is to store large numbers in accordance with lines, here it can follow the column to store large integer numbers, for the data itself is not much conflict. Therefore, under the new access methods, we can achieve higher processing efficiency.

Keywords

Multiple-precision, Big Integer, GPU Computing, CUDA, Memory alignment Access, Multiple-precision algorithm

1. Introduction

Non-symmetric encryption is usually used in the network, and his calculation of the core is the processing of large numbers. Large amounts of data on the network to encode or to perform non-symmetric encryption, you need to use a lot of computation time, processing of large numbers will take up much of time. How to reduce computing time, speed up the encoding and encryption of data in the current environment of great challenges. In particular, the amount of data is particularly large when the calculation of encryption or encoding, particularly time-consuming. The emergence of GPU makes this situation changed. GPU is good at handling large amounts of data work.

Recent advances in Graphics Processing Units (GPUs) open a new era of GPU computing [20]. For example, commodity GPUs like NVIDIA's GTX 280 has 240 processing cores and can achieve 933 GFLOPS of computational horsepower. More importantly, the NVIDIA CUDA programming model makes it easier for developers to develop non-graphic applications using GPU [1] [4]. In CUDA, the GPU becomes a dedicated coprocessor to the host CPU, which works in the principle of Single-Program Multiple Data (SPMD) where multiple threads based on the same code can run simultaneously.

While the GPU computing power is high, but our actual test results did not meet the highest performance. But how can we make to play a higher performance GPU now? Analysis we have found that the GPU code, GPU computing power of the performance bottleneck is the memory read section. Our research GPU threading model and memory access model of consolidation of large numbers using the new memory layout model to modify the traditional large numbers in the memory access methods, allowing GPU threads can follow the way to access the memory alignment of data to speed up the memory of the read.

The rest of the paper is organized as follows. Section 2 provides background information on Multiple-precision, GPU architecture, and CUDA programming model， GPU thread model and memory access model. Section 3 presents the design of multiple-precision integer arithmetic on GPU. Section 4 we design a new memory access model for multiple-precision. Experimental results are presented in Section 5, and we conclude the paper in Section 6.

2. Background and Related Work

In this section, we provide the required background knowledge of Multiple-precision, GPU architecture and CUDA programming model.

2.1 Multiple-precision Integer

First, we describe that deal with large integer numbers, why do we need to deal with large numbers. In the non-symmetric encryption, using large numbers more difficult to break down characteristics, to encrypt the data. The number of larger, more difficult to break greater, and now usually used in large numbers are 1024bit bit. Coding in the network will deal with the situation of large numbers, only the larger situation of large numbers, including the amount of information before it was. For example, we will then be used to make the analysis of an algorithm, a vector is multiplied by a data and then seek mode, the process is the encoding process of handling large numbers. Location of large numbers of data over 1024bit position, then we know the computer inside the existing system is 32 bit position, or the 64bit bit. How to represent the present large numbers now? We know that number can be expressed as $A = x_0 \cdot 10^0 + x_1 \cdot 10^1 + \ldots x_n \cdot 10^n$; such a way that a number of the same, we can also be used inside the computer the same way to represent a large numbers, here we have the b-bit 0x100000000, using 2^{32} as a binary. We define binary bit b, then the large numbers can be expressed as $A = x_0 \cdot b^0 + x_1 \cdot b^1 + \ldots x_n \cdot b^n$. These circumstances, we can express the location of large numbers of more than 32bit. For example, large numbers 1024bit bit, we can use 8-bit integer type of representation is 32bit.

We can see large integer is an expression of the polynomial. So we have large numbers of addition and subtraction multiplication and division rules. In fact, the conversion to polynomial multiplication and division addition and subtraction. In our realization of the large numbers library, we used a way to carry out large numbers polynomial arithmetic. Because large numbers have been a lot of processing algorithms, where we do not deal with large numbers of the algorithm to do too much to explain, but we choose the best and most suitable for GPU algorithms. Here we will discuss what kind of large numbers algorithm is suitable for GPU, for the present CUDA programming. Processing of large numbers, especially multiplication, exponentiation, modular computing, and power-mode operation in practical application a large proportion of, so large numbers in our database, we will focus on the realization of multiplication, power-mode operation.

2.2 GPU Computing and CUDA

GPUs are dedicated hardware for manipulating computer graphics. Due to the huge computing demand for real-time and high-definition 3D graphics, the GPU has evolved into a highly parallel, multithreaded, manycore processor. The advances of computing power in GPUs have driven the development of general-purpose computing on GPUs (GPGPU). The first generation of GPGPU requires that any non-graphics application must be mapped through graphics application programming interfaces (APIs).

Recently one of the major GPU vendors, NVIDIA, announced their new general-purpose parallel programming model, namely Compute Unified Device Architecture (CUDA) [1] [4], which extends the C programming language for general-purpose application development. Meanwhile, another GPU vendor AMD also introduced Close To Metal (CTM) programming model which provides an assembly language for application development [2]. Intel also exposed Larrabee, a new many-core GPU architecture specifically designed for the market of GPU computing this year [23].

Since the release of CUDA, it has been used for speeding up a large number of applications [17] [18] [20] [21] [22].

The NVIDIA GeForce 8800 has 16 Streaming Multiprocessors (SMs), and each SM has 8 Scalar Processors (SPs), resulting a total of 128 processor cores. The SMs have a Single-Instruction Multiple-Data (SIMD) architecture: At any given clock cycle, each SP of the SM executes the same instruction, but operates on different data. Each SP can support 32-bit single-precision floating-point arithmetic as well as 32-bit integer arithmetic.

Each SM has four different types of on-chip memory, namely registers, shared memory, constant cache, and texture cache. For GeForce 8800, each SM has 8192 32-bit registers, and 16 Kbytes of shared memory which are almost as fast as registers. Constant cache and texture cache are both read-only memories shared by all SPs. Off-chip memories such as local memory and global memory have relatively long access latency, usually 400 to 600 clock cycles [4]. The properties of the different types of memories have been summarized in [4] [17]. In general, the scarce shared memory should be carefully utilized to amortize the global memory latency cost. Shared memory is divided into equally-sized banks, which can be simultaneously accessed. If two memory requests fall into the same bank, it is referred to as bank conflict, and the access has to be serialized.

In CUDA model, the GPU is regarded as a coprocessor capable of executing a great number of threads in parallel. A single source program includes host codes running on CPU and also kernel codes running on GPU. Compute-intensive and data-parallel kernel codes run on GPU in the manner of Single-Process Multiple-Data (SPMD). The threads are organized into blocks, and each block of threads are executed concurrently on one SM. Threads in a thread block can share data through the shared memory and can perform barrier synchronization. Each SM can run at most eight thread blocks concurrently, due to the hard limit of eight processing cores per SM. As a thread block terminate, new blocks will be launched on the vacated SM. Another important concept in CUDA is warp, which is formed by 32 parallel threads and is the scheduling unit of each SM. When a warp stalls, the SM can schedule another warp to execute. A warp executes one instruction at a time, so full efficiency can only be achieved when all 32 threads in the warp have the same execution path. Hence, if the number of threads in a block is not a multiple of warp size, the remaining instruction cycles will be wasted.

Of particular importance is the CUDA model of GPU threads to access the memory model. Under normal circumstances this place will become a bottleneck GPU acceleration. Each thread reads a 32 bit data, put the rules into alignment here. When all the threads in a half-warp (16 threads) read the collocation memory the access will be make one step.

3. Multiple-Precision Modular Arithmetic for CUDA

In this section, we present a set of library functions of multiple-precision modular arithmetic implemented on GPUs. These library functions are the cornerstones of the network coding system and homomorphic hash functions. It is of critical importance to implement these library functions efficiently. In modular arithmetic, all operations are performed in a group Z_m, i.e., the set of integers $\{0,1,2,\cdots,m-1\}$. In the following, the modulus m is represented in radix b as $(m_n m_{n-1} \cdots m_1 m_0)_b$ where $m_n \neq 0$. Each symbol $m_i, 0 \leq i \leq n$, is referred to as a radix b digit. Non-negative integers x and y, $x < m, y < m$, are represented in radix b as $(x_n x_{n-1} \cdots x_1 x_0)_b$ and $(y_n y_{n-1} \cdots y_1 y_0)_b$ respectively.

We have implemented the following multiple-precision library functions for CUDA:

Multiple-precision comparison

Multiple-precision subtraction

Multiple-precision modular addition

Multiple-precision modular subtraction

Multiple-precision multiplication

Multiple-precision division

Multiple-precision multiplicative inversion

Due to the space limitation, we do not present the implementation details in this paper.

3.1 Modular Addition and Subtraction

Algorithm 1 Multiple-precision Comparison

INPUT: non-negative integers *x* and *y*, each with $n+1$ radix *b* digits.

OUTPUT: 1, if $x > y$; 0, if $x = y$; -1, if $x < y$.

```
$i \leftarrow n$;
while ( $x_i == y_i$ and $i > 0$ )
    $i \leftarrow i-1$;
end while
if ( $x_i > y_i$ ) then return 1;
else if ( $x_i == y_i$ ) then return 0;
else return -1;
```

Algorithm 2 Multiple-precision Subtraction

INPUT: non-negative integers *x* and *y*, each with $n+1$ radix *b* digits, $x \geq y$.

OUTPUT: $x - y = (z_n z_{n-1} \cdots z_1 z_0)_b$.

```
$c \leftarrow 0$; /* carry digit */
for ( $i$ from 0 to $n$ ) do
    $z_i \leftarrow (x_i - y_i + c)$ mod $b$;
    if ( $x_i - y_i + c \geq 0$ ) then $c \leftarrow 0$;
    else $c \leftarrow -1$;
end for
return $(z_n z_{n-1} \cdots z_1 z_0)_b$;
```

Algorithm 3 Multiple-precision Modular Addition

INPUT: non-negative integers *x* and *y*, each with $n+1$ radix *b* digits, $x < m, y < m$.

OUTPUT: $(x+y)$ mod m = $(z_n z_{n-1} \cdots z_1 z_0)_b$.

```
$c \leftarrow 0$; /* carry digit */
for ( $i$ from 0 to $n$ ) do
    $z_i \leftarrow (x_i + y_i + c)$ mod $b$;
    if ( $x_i + y_i + c < b$ ) then $c \leftarrow 0$;
    else $c \leftarrow 1$;
end for
$z_{n+1} \leftarrow c$; $m_{n+1} \leftarrow 0$;
if ( $(z_{n+1} z_n z_{n-1} \cdots z_1 z_0)_b >= (m_{n+1} m_n m_{n-1} \cdots m_1 m_0)_b$ ) then
    $(t_{n+1} t_n t_{n-1} \cdots t_1 t_0)_b \leftarrow$ $(z_{n+1} z_n z_{n-1} \cdots z_1 z_0)_b -$
                                   $(m_{n+1} m_n m_{n-1} \cdots m_1 m_0)_b$;
    return $(t_n t_{n-1} \cdots t_1 t_0)_b$;
else return $(z_n z_{n-1} \cdots z_1 z_0)_b$;
```

Algorithm 4 Multiple-precision Modular Subtraction

INPUT: non-negative integers *x* and *y*, each with $n+1$ radix *b* digits, $x < m, y < m$.

OUTPUT: $(x-y)$ mod m = $(z_n z_{n-1} \cdots z_1 z_0)_b$.

```
if ( $x >= y$ ) then return $x - y$;
else
    $t \leftarrow (m - y)$;
    return $(x + t)$ mod $m$;
end else
```

Complexity Analysis: Obviously all the above algorithms have computational complexity of $O(n)$.

3.2 Modular Multiplication

One straightforward method to implement modular multiplication of $x \cdot y$ mod *m* is to calculate $x \cdot y$ first and then calculate the remainder of $x \cdot y$ divided by *m*. Hence we first give two algorithms to calculate multiple-precision multiplication and division respectively.

Algorithm 5 Multiple-precision Multiplication

INPUT: non-negative integers *x* and *y*, each with $n+1$ radix *b* digits and $s+1$ radix *b* digits respectively.

OUTPUT: $x \cdot y$ = $(z_{n+s+1} z_{n+s} \cdots z_1 z_0)_b$.

```
for ( $i$ from 0 to $n+s+1$ ) do
    $z_i \leftarrow 0$;
end for
for ( $i$ from 0 to $s$ ) do
    $c \leftarrow 0$; /* carry digit */
    for ( $j$ from 0 to $n$ ) do
        $(uv)_b \leftarrow z_{i+j} + x_j \cdot y_i + c$;
        $z_{i+j} \leftarrow v$; $c \leftarrow u$;
    end for
    $z_{n+i+1} \leftarrow u$;
end for
return $(z_{n+s+1} z_{n+s} \cdots z_1 z_0)_b$;
```

Algorithm 6 Multiple-precision Division

INPUT: non-negative integers *x* and *y*, each with $n+1$ radix *b* digits and $s+1$ radix *b* digits respectively, $n \geq s \geq 1$, $y_s \neq 0$.

OUTPUT: the quotient $q = (q_{n-s} \cdots q_1 q_0)_b$ and remainder $r = (r_s \cdots r_1 r_0)_b$ such that $x = q \cdot y + r$, $0 \leq r < y$.

```
for ( $i$ from 0 to $n-s$ ) do
    $q_i \leftarrow 0$;
end for
while ( $x \geq y \cdot b^{n-s}$ ) do
    $q_{n-s} \leftarrow q_{n-s} + 1$;
    $x \leftarrow x - y \cdot b^{n-s}$;
```

```
end while
for ( $i$ from $n$ down to $t+1$ ) do
    if ( $x_i == y_s$ ) then $q_{i-s-1} \leftarrow b-1$;
    else $q_{i-s-1} \leftarrow \lfloor (x_i \cdot b + x_{i-1}) / y_s \rfloor$;
    while $(q_{i-s-1} \cdot (y_s \cdot b + y_{s-1}) > x_i \cdot b^2 + x_{i-1} \cdot b + x_{i-2})$ do
        $q_{i-s-1} \leftarrow q_{i-s-1} - 1$;
    end while
    $x \leftarrow x - q_{i-s-1} \cdot y \cdot b^{i-s-1}$;
    if ( $x < 0$ ) then
        $x \leftarrow x + y \cdot b^{i-s-1}$;
        $q_{i-s-1} \leftarrow q_{i-s-1} - 1$;
    end if
end for
$r \leftarrow x$;
return $(q, r)$;
```

The classical modular multiplication is suitable for normal operations. However, when performing modular exponentiations, Montgomery multiplication shows much better performance advantage [7]. The following gives the Montgomery reduction and Montgomery multiplication algorithms.

Let m be a positive integer, and let R and A be integers such that $R > m$, gcd(m, R) = 1, and $0 \le A < m \cdot R$. The Montgomery reduction of A modulo m with respect to R is defined as $A \cdot R^{-1}$ mod m. In our applications, R is chosen as b^n to simply the calculation.

Algorithm 7 Multiple-precision Montgomery Reduction

INPUT: integer m with n radix b digits and gcd(m, b) = 1, $R = b^n$, $m' = -m^{-1}$ mod b, and integer A with $2n$ radix b digits and $A < m \cdot R$.

OUTPUT: $T = A \cdot R^{-1}$ mod m.

```
$T \leftarrow A$;
for ( $i$ from $0$ to $n-1$ )
    $u_i \leftarrow T_i \cdot m'$ mod $b$;
    $T \leftarrow T + u_i \cdot m \cdot b^i$;
end for
$T \leftarrow T / b^n$;
if ( $T \ge m$ ) then $T \leftarrow T - m$;
return $T$;
```

Algorithm 8 Multiple-precision Montgomery Multiplication

INPUT: non-negative integer m, x, y with n radix b digits, $x < m, y < m$, and gcd(m, b) = 1, $R = b^n$, $m' = -m^{-1}$ mod b.

OUTPUT: $T = x \cdot y \cdot R^{-1}$ mod m.

```
$T \leftarrow 0$;
for ( $i$ from $0$ to $n-1$ )
    $u_i \leftarrow (T_0 + x_i \cdot y_0) \cdot m'$ mod $b$;
    $T \leftarrow (T + x_i \cdot y + u_i \cdot m) / b$;
end for
if ( $T \ge m$ ) then $T \leftarrow T - m$;
return $T$;
```

3.3 Barrett Modular Reduction Algorithm

Barrett Reduction is a method of reducing a number modulo another number. Barrett reduction by precomputing some values, one can easily far exceed the speed of normal modular reductions.

Barrett reduction's benefits are most visible when it is used to reduce various numbers modulo a single number many times. Barrett reduction is not particularly useful when used with small numbers (32 or 64 bits); it's benefits occur when using numbers that are implemented by multiple precision arithmetic libraries, such as when implementing the network coding in GF(2^{32}).

Next I will give the implementation of the Barrett reduction algorithm. But first, keep in mind that Barrett Reduction can only reduce numbers that are, at most, twice as long (in words) as the modulus. We define the modulus, called m, which is k words long (numbered k-1...0, with 0 being the least significant word). First we need pre-calculate the value: $\mathrm{u} = \lfloor \mathrm{b}^{2\mathrm{k}} / \mathrm{m} \rfloor$ where b is the "base" of the integers used. For example, if you represented the numbers as a sequence of 32-bit values, b is 2^{32}, or 0x100000000. You will keep this value u across function calls so you can reuse it.

Now, given a number x, which is an arbitrary integer of size (at most) 2k words (2k-1...0), this procedure (in pseudocode) will return the value of x mod m:

Algorithm 9 Barrett Modular Reduction Algorithm

INPUT: positive integers $x = (x_{2k-1} \dots x_1 x_0)_b$, $m = (m_{k-1} \dots m_1 m_0)_b$ (with $m_{k-1} \neq 0$), and $u = \lfloor b^{2k} / m \rfloor$.

OUTPUT: $r = x \bmod m$

```
$\mathrm{q}_1 = \lfloor x / b^{k-1} \rfloor$, $q_2 = q_1 \cdot u$, $\mathrm{q}_3 = \lfloor q_2 / b^{k+1} \rfloor$;
$r_1 = x \bmod b^{k+1}$, $r_2 = q_3 \cdot m \bmod b^{k+1}$, $r = r_1 - r_2$;
if ($r < 0$) then $r = r_1 + b^{k+1}$;
while ($r \ge m$) do $r = r_r - m$;
return r;
```

Note that the divisions and modular reductions in this procedure can be replaced by right shifts and AND operations because the b is 2^{32}. This results in the remaining operations being addition and multiplication, both of which are much cheaper than division for multiple precision integers.

3.4 Multiplicative Inversion

Traditionally multiplicative inversion is obtained through extended Euclidean algorithm. In order to avoid the expensive multiple-precision division operations, we implement multiple-precision multiplicative inversion using an extended binary GCD algorithm.

Algorithm 10 Multiple-precision Multiplicative Inversion

INPUT: odd prime number m with n radix b digits, positive integer a with n radix b digits, $a < m$.

OUTPUT: integer $b \in Z_m$ such that $a \cdot b \equiv 1 \pmod{m}$.

```
$u \leftarrow m$; $v \leftarrow a$; $B \leftarrow 0$; $D \leftarrow 1$;
while $u$ is even
    $u \leftarrow u / 2$;
    if $B$ is even then $B \leftarrow B / 2$;
    else $B \leftarrow (B - m) / 2$;
end while
while $v$ is even
```

```
8:      $v \leftarrow v/2$;
9:      if $D$ is even then $D \leftarrow D/2$;
10:     else $D \leftarrow (D-m)/2$;
11: end while
12: if ($u \geq v$) then $u \leftarrow u-v$; $B \leftarrow B-D$;
13: else $v \leftarrow v-u$; $D \leftarrow D-B$;
14: if $u == 0$ then return $D$;
15: else go to Step 2.
```

3.5 Modular Exponentiation

Algorithm 11 Multiple-precision Montgomery Exponentiation

INPUT: integer m with n radix b digits and gcd(m, b) = 1, $R = b^n$, positive integer x with n radix b digits and $x < m$, and positive integer e = $(e_t \cdots e_0)_2$.

OUTPUT: x^e mod m.

```
1:  $\tilde{x} \leftarrow Mont(x, R^2 \ mod\ m)$;
2:  $A \leftarrow R$ mod $m$;
3:  for ( $i$ from $n$ down to 0)
4:      $A \leftarrow Mont(A,\ A)$;
5:      if $e_i == 1$ then $A \leftarrow Mont(A,\ \tilde{x})$;
6:  end for
7:  $A \leftarrow Mont(A,\ 1)$;
8:  return $A$;
```

3.6 Exponentiation with Multi-Exponentiation

There exist multi-exponentiation algorithms which perform much better than calculating the exponents individually. The following algorithm is a variation of Straus's algorithm [5], by integrating the Montgomery method. To evaluate $\prod_{i=0}^{k-1} g_i^{e_i}$ where the maximum bit-length of all the exponents is n, we first form a $k \times n$ bit matrix whose rows are the binary representations of e_i, $0 \leq i \leq k-1$. Let C_j be the non-negative integer whose binary representation is the jth column, $1 \leq j \leq n$, of the bit matrix, where the least significant bit is at the top of the column.

Algorithm 12 Montgomery Multi-Exponentiation

INPUT: integers $g_0, g_1, ..., g_{k-1}$, $e_0, e_1, ..., e_{k-1}$, R, and m.

OUTPUT: $\prod_{i=0}^{k-1} g_i^{e_i}$ mod m.

```
1:  for ( $i$ from 0 to $2^k - 1$ )
2:      $\tilde{G}_i \leftarrow (\prod_{j=0}^{k-1} g_j^{i_j})R$ mod $m$, where $i = (i_{k-1} \cdots i_0)_2$
3:  end for
4:  $A \leftarrow R$ mod $m$;
5:  for ( $j$ from 1 to $n$)
6:      $A \leftarrow Mont(A, A)$;
7:      if $c_j \neq 0$ then $A \leftarrow Mont(A, \tilde{G}_{C_i})$;
8:  end for
9:  $A \leftarrow Mont(A,\ 1)$;
10: return $A$;
```

The above algorithm requires at most $2^k + 2(n-2)$ multiplication operations, with a memory space of $2^k - 1$. The optimal value of k depends on the value of n, and it is usually small for contemporary cryptography applications.

Given the structure of homomorphic hash function, we can divide the product $h(e) = \prod_{i=1}^{m} g_i^{e_i}$ into groups of multi-exponentiations: $h(e) = \prod_{i=1}^{\lceil m/k \rceil} (\prod_{j=1}^{k} g_{(i-1)k+j}^{e_{(i-1)k+j}})$, and assign each multi-exponentiation to an individual processing core.

3.7 Exponetiation with Precomputation

When applying homomorphic hash function in network coding enabled P2P applications, the same homomorphic hash function, i.e., with the same set of parameters, will be used for a large data set such as a whole file or a video streaming session. Under this special circumstance, it is possible to speed up the modular exponentiations by precomputation [8]. To calculate g^e, we first represent the exponent e using radix $b = 2^k$: $e = \sum_{i=0}^{n-1} a_i b^i$, where $0 \leq a_i < b$ and $a_{n-1} \neq 0$. It is easy to see that $n = \lceil (\lfloor \log_2 e \rfloor + 1)/k \rceil$. The fast modular exponentiation algorithm requires the precomputation of $g^{2^{ki}}$ mod m for $1 \leq i \leq n-1$. Then we can use the following algorithm to calculate g^e mod m.

Algorithm 13 Exponentiation with Precomputation

INPUT: integers m, g, $e = \sum_{i=0}^{m-1} a_i b^i$, R, and $Rg^{2^{ki}}$ mod m for $1 \leq i \leq n-1$

OUTPUT: g^e mod m.

```
1:  $A \leftarrow R,\ B \leftarrow R$;
2:  for ( $j$ from $b-1$ down to 1)
3:      for $i$ from 0 to $m-1$
4:          if $a_i == j$ then $B \leftarrow Mont(B, Rg^{2^{ki}})$ mod $m$;
5:      end for
6:      $A \leftarrow Mont(A, B)$;
7:  end for
8:  $A \leftarrow Mont(A,1)$;
9:  return $A$;
```

As shown by [8], the above algorithm takes $m+b-3$ multiplications. For e with 257-bit, the optimal value of b is 16, which takes only 78 multiplications in the worst case, as compared with 512 multiplications required by the binary method.

4. Implementation Multiple-precision Integer library on GPUS

This chapter, we will discuss the use of memory with alignment approach to the implementation of Multiple-precision integer numbers. In the GPU, in order to achieve maximum access to bandwidth, we need a CUDA half-warp access the data inside the data, when, in accordance with coalescing way to access memory.

Next part we will discuss the way, how the half-warp access the memory by coalescing .

4.1 Data Structure of Multiple-precision Integer

There are two ways to organize the data, one way is the CPU data structure, one is suitable for GPU data structures. We define large numbers over more than 64bit or 32bit, so we need to re-definition of the structure of the big integer. It based on 2^{32}, each element of big integer is 32bit, according to 2^{32} as a binary, a total number of bits into the data structure of the last one. In our implementation, each of a large numbers are all composed of two parts, first part is the actual data of large numbers, the second part of the numbers of bits, these two can be separated is placed, it is not released to the same array.

The first way is to put the data in accordance with the habit of CPU, in order, put each number into a 32bit memory inside. Another way to put the data in accordance with GPU appropriate manner, and put 32 large numbers in each row, each a large numbers of data are arranged according to the column.

Our actual calculations will be used simultaneously to both structures, a number of constants, or only once the data, we can put inside GPU computing, or stored in a number of high-performance memory inside, place the data in accordance with the first approach. Be counted some of the data, we will follow for GPU-series data to store data

4.2 Using Constant Value with Cache Memory

In the calculation process, we usually use to duplicate data, or to be multiple uses of data. The data we need to try to put to efficient memory cache inside. In Nvidia's GPU we can use texture and constant memory storage of constants. Texture and the constant has two caches, multiple use of the data will be loaded into the cache inside, so that you can efficiently use the constant data. For example, exponentiation of the base, or are seeking modules modulus.

4.3 Using Shared Memory for Temp Value

We can see that some of the above formula to the process need to use temporary variables. Because the performance of local memory and global memory as well, so this part of the variable we will use shared memory to be stored. All of the intermediate process is the need to reiterate the use of variables. And then passed to a function as a temporary variable to use it.

4.4 Balance the Computing Resource

Balance the number of threads for each block, making a maximum of active threads. CUDA programming model Stream Multiprocessor each one simultaneously active 8 blocks, but the need to registers for each block and the shared memory used in the calculation. The need to co-inside with the 4.3, sometimes without direct use of shared memory as global memory can achieve higher performance

4.5 Example Implementation in GPUs

This part, we design a example for benchmarking this library. There is a vector array A multiplication with big integer matrix B, and then mod 1024bit big integer. Large numbers in the original matrix arranged in memory when the inside is based on a lump sum for each line to represent. A lump sum for each of a lump sum and the next will be the location of 256bit interval, so that the GPU inside the access and causing access can't be merged. This will allow each half-warp access memory produce 16 times the time to visit. Each visit is approximately 500 cycle, if it is can be merged into one visit, then these 16 threads will be merged into one visit, which will reduce the access time period, increasing access to bandwidth.

So in this case, we have designed to store large numbers under the column of the memory layout. Can see the figure below, we show that the yellow unit One-big-integer-256bits, says that a large numbers. So, 16 threads to access the matrix B when the alignment means in accordance with the order would be to visit each of the first data row, so that you can make 16 threads to access the data alignment.

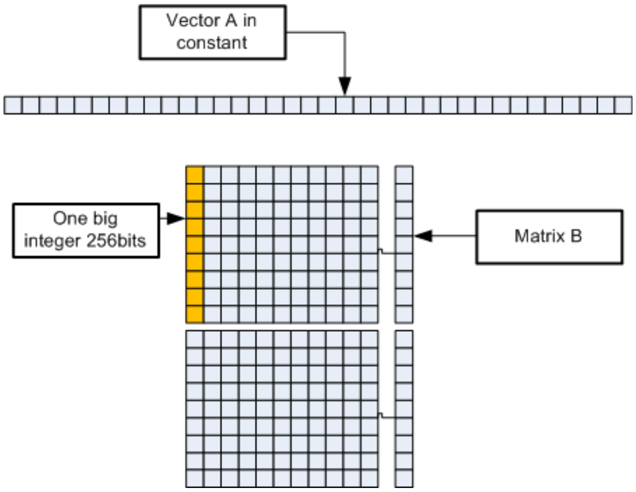


Figure 1. Vector A multiplication with Big integer matrix B

Because the vector A will be used multiple times, so we will put into the vector A into constant memory, because there will be two constant caches, you can speed up the data read

5. Implementation and experimental Result

We have implemented this function using CUDA. We tested these implementations on XFX GTX280 graphic card which contains an NVIDIA GeForce GTX280 GPU. The GTX280 GPU uses the GT200 architecture with 240 processing cores working at 1.24 GHz.

5.1 Result of the Example

Figure 2 shows the results of our CUDA version of network coding encoding with GT200 GPU. Since most of the computing task is the modular, the throughput is almost independent of the value of m. The encoding speed about 400 MBPs.

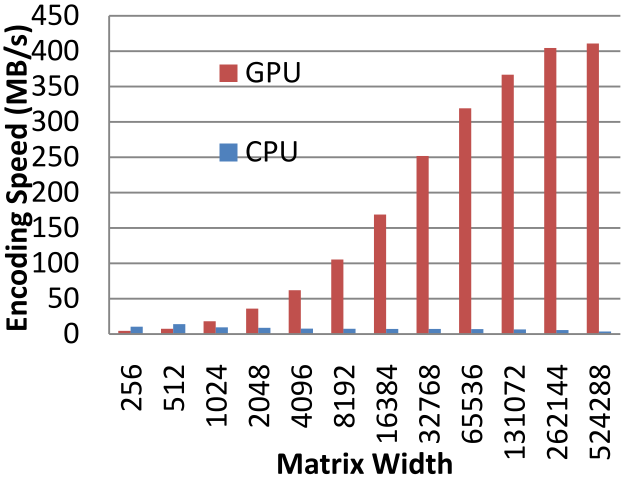


Figure 2． Network Coding Encoding Speed with CUDA

6. Conclusions

Multiple-precision modular is an important component in public-key cryptography for encrypting and signing digital data. In this paper, we describe the design, implementation and optimization of multiple-precision modular using GPU and CUDA. Especially in the case of dealing with large amounts of data, how to optimize the access speed of GPU, we have changed the traditional way of their large numbers of memory, which is more suitable for GPU access the memory data.

7. References

[1] NVIDIA CUDA. http://developer.nvidia.com/object/cuda.html

[2] AMD CTM Guide: Technical Reference Manual. 2006. http://ati.amd.com/companyinfo/researcher/documents/ATI_CTM_Guide.pdf

[3] GNU MP Arithmetic Library. http://gmplib.org/

[4] NVIDIA CUDA Compute Unified Device Architecture: Programming Guide, Version 2.0beta2, Jun. 2008.

[5] Montgomery, P., 1985. Multiplication without trial division, Math. Computation, vol. 44, 1985, 519-521.

[6] Menezes, A., van Oorshot, P., and Vanstone S., 1996. Handbook of applied cryptography. CRC Press, 1996.

[7] Ahlswede, R., Cai, N., Li S. R., and Yeung, R. W. 2000. Network information flow. IEEE Transactions on Information Theory, 46(4), July 2000, 1204-1216.

[8] Ho, T., Koetter. R., Médard, M., Karger, D.R. and Effros, M. 2003. The benefits of coding over routing in a randomized setting. In Proceedings of IEEE ISIT, 2003.

[9] Li, S.-Y.R., Yueng, R.W., and Cai, N. 2003. Linear network coding. IEEE Transactions on Information Theory, vol. 49, 2003. 371-381.

[10] Krohn, M., FreedMan, M., and Mazieres, D. 2004. On-the-fly verification of rateless erasure codes for efficient content distribution. In Proceedings of IEEE Symposium on Security and Privacy, Berkeley, CA, 2004.

[11] Gkantsidis, C. and Rodriguez, P. 2005. Network coding for large scale content distribution. In Proceedings of IEEE INFOCOM 2005.

[12] Gkantsidis, C. and Rodriguez, P. 2006. Cooperative security for network coding file distribution. In Proceedings of IEEE INFOCOM'06, 2006.

[13] Li, Q., Chiu, D.-M., and Lui, J. C.S. 2006. On the practical and security issues of batch content distribution via network coding. In Proceedings of IEEE ICNP'06, 2006, 158-167.

[14] Chou, P. A. and Wu, Y. 2007. Network coding for the Internet and wireless networks. Technical Report. MSR-TR-2007-70, Microsoft Research.

[15] Wang, M. and Li, B. 2007. Lava: a reality check of network coding in peer-to-peer live streaming. In Proceedings of IEEE INFOCOM'07, 2007.

[16] Wang, M. and Li, B. 2007. R^2: random push with random network coding in live peer-to-peer streaming. In IEEE Journal on Selected Areas in Communications, Dec. 2007, 1655-1666.

[17] Ryoo, S., Rodrigues, C. I., Baghsorkhi, S. S., Stone, S. S., Kirk, D. B., and Hwu, W. 2008. Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In Proceedings of ACM PPoPP'08, Feb. 2008.

[18] Falcao, G., Sousa, L., and Silva, V. 2008. Massiv parallel LDPC decoding in GPU. In Proceedings of ACM PPoPP'08, Feb. 2008.

[19] Yu, Z., Wei, Y., Ramkumar, B., and Guan, Y. 2008. An efficient signature-based scheme for securing network coding against pollution attacks. In Proceedings of IEEE INFOCOM'08, Apr. 2008.

[20] Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. 2008. GPU computing. IEEE Proceedings, May 2008, 879-899.

[21] Al-Kiswany, S., Gharaibeh, A., Santos-Neto, E., Yuan, G., and Ripeanu, M. 2008. StoreGPU: exploiting graphics processing units to accelerate distributed storage systems. In Proceedings of IEEE Symposium on High Performance Distributed Computing (HPDC), Jun. 2008.

[22] Silberstein, M., Geiger, D., Schuster, A., Patney, A., Owens, J. D. 2008. Efficient computation of sum-products on GPUs through software-managed cache. In Proceedings of the 22nd ACM International Conference on Supercomputing, Jun. 2008.

[23] Seiler, L., et. al., 2008. Larrabee: a many-core x86 architecture for visual computing. ACM Transactions on Graphics, 27(3), Aug. 2008.

Hiding Emerging Pattern with Local Recoding Generalization

Michael Wai-Kit Cheng

Abstract

Establishing strategic partnership often requires organizations to publish and share meaningful data to support collaborative business activities. An equally important concern for them is to protect sensitive patterns like unique emerging sales opportunities embedded in their data. In this paper, we contribute to the area of data sanitization by introducing an optimization-based local recoding methodology to hide emerging patterns from a dataset but with the underlying frequent itemsets preserved as far as possible. We propose a novel heuristic solution that captures the unique properties of hiding `EP`*s to carry out iterative local recoding generalization. Also, we propose a metric which measures (i) frequent-itemset distortion that quantifies the quality of published data and (ii) the degree of reduction in emerging patterns, to guide a bottom-up recoding process. We have implemented our proposed solution and experimentally verified its effectiveness with a benchmark dataset.*

Keywords: Emerging patterns, pattern hiding, data sanitization, frequent itemsets

1 Introduction

Organizations often publish and share their data to support business collaboration. In the context of marketing and sales, companies can leverage on the customer pools of each other for cross-selling so that the involved parties can gain sales volume increase. Due to the equally important need of privacy protection, customers often expect their data to be anonymized before sharing [26] and studies on privacy-preserving data publishing have bloomed [11]. Furthermore, trade secrets embedded in data are valuable to organizations [**?**] and needed to be properly protected. For instance, patterns like recent increase in the sales volume of a product line for a certain customer group (emerging marketing trends) can be an example. Leaking of related intelligence could cause company loss in gaining the first-mover advantage. Even though companies understand that data sharing is unavoidable to support collaborative activities like cross-selling, they may face a great hindrance to data sharing if the emerging sales opportunities of their own business cannot be hidden.

Among others, *emerging patterns* [18] embedded in data carry sensitive information, that data owners may prefer to hide. In fact, previous studies have revealed that emerging patterns are highly discriminative when used as features for classification [28, 10, 7], and thus carry salient features of the data. The hiding, however, is technically challenging as collaborative data analysis is still often expected to facilitate collaboration. That is, some statistical properties of the data to-be-shared are preserved as far as possible. In particular, frequent itemset mining has already been well-supported in most commercial data-mining packages. Therefore, in this paper, we study how to hide emerging patterns while preserving frequent itemsets.

To hide emerging patterns, we adopt recoding generalization methods. In particular, we adopt local recoding which is (intuitively) a value-grouping generalization process, given an attribute generalization hierarchy. To ensure that the generalized data neither (i) reveal sensitive information nor (ii) produce a highly distorted mining result, we propose metrics for quantifying the two competing objectives. With the metrics, we present an iterative, bottom-up optimization framework. Compared with hiding frequent itemsets [24], hiding emerging patterns is more technically challenging. In particular, the *a priori anti-monotone* property does not hold in emerging patterns. Thus, the search space of emerging patterns is huge. Worst still, a local recoding may hide an emerging pattern while generating new emerging patterns. To the best of our knowledge, there has not been work on hiding emerging patterns.

2 Related Work

Studies on data sanitization can be dated back to the earlier work on statistical disclosure control [1]. Recent development in privacy preserving data mining [25] has contributed to some advances in privacy measure and data sanitization method. For example, to avoid personal identity to be recovered from an anonymized demographic dataset, a number of privacy measures were proposed in the literature, e.g., k-anonymity [26] and ℓ-diversity [21]. Other privacy measures include k^m-anonymity [**?**] and (h,k,p)-coherence [**?**]. Given a particular measure, recoding generalization [18, 13, 17, 25, 8, 30, 19] and perturbation [2, 9, 16] are two commonly adopted data sanitization approaches. Recoding generalization is often preferred over the perturbation approach as the dataset sanitized by recoding generalization is still semantically consistent with the original one, even though it is "blurred". While this study aims at hiding emerging patterns instead of personal identities, the concepts like equivalence classes and recoding generalization are adopted in the proposed methodology.

To control the distortion of the data caused by the sanitization, attempts have been made to preserve as much information of the original dataset as possible to, say, preserve the subsequent classification accuracy [14] and clustering structure [12]. In addition, there has been some recent work studying the tradeoff between privacy and utility [20] in the context of privacy-preserving data publishing. In this work, we try to preserve the frequent itemsets of the data as far as possible.

Recently, there has been work [24, 23] on hiding patterns like frequent itemsets where users specify a subset of frequent itemsets, namely sensitive frequent itemsets, that are not supposed to be disclosed to the public. In our study, we focus on hiding emerging patterns, which makes a unique contribution to the area of pattern hiding. Emerging patterns (`EP`) are features that are distinctive from one class to another and has been found to be effective for building highly accurate classifiers [15, 28, 10, 7]. Mining `EP`s from large databases is technically intriguing as the total number of `EP`s, in the worst case, is exponential to the total number of attributes in transactions, and there has not been a corresponding notion of the apriori anti-monotone property of frequent itemsets in `EP`s so that the search space can be pruned. Previous work on `EP`s mainly focuses only on the mining efficiency, e.g., using a border-based approach [4], a constraint-based approach [31], or focusing only on jumping `EP`s [3]. So far, there exists no related work on emerging pattern hiding.

3 Background and Problem Statement

In the following, we present the definitions, notations used and the problem statement.

A *transactional dataset* is a set of transactions. Let $I = \{i_1, i_2, ..., i_n\}$ be a finite set of distinct *items* in D. A *transaction t* has a set of nominal attributes $A = \{a_1, a_2, ..., a_m\}$ and each attribute a_i takes values from a set $V_i \subseteq I$. We make two simple remarks about these notations. (i) While we assume transactional data with nominal attributes, data of a continuous domain can be cast into nominal data, for example by defining ranges. (ii) One may consider a relation as a set of transactions of a fixed arity.

An *itemset* X is a (proper) subset of I. $Supp_D(X)$ denotes the support of an itemset X in a dataset D, which can be computed as $\frac{|\{t \mid X \in t \wedge t \in D\}|}{|D|}$. Given a support threshold σ, X is said to be a *σ-frequent itemset* if $Supp_D(X) \geq \sigma$. The growth rate of an itemset is the ratio of its support in one dataset to that in the other.

Definition 3.1: [6] Given two datasets, namely D_1 and D_2, the *growth rate* of an itemset X, denoted as $GR(X, D_1, D_2)$, from D_1 to D_2 is defined as $GR(X, D_1, D_2)$ =

$$\begin{cases} 0 & \text{, if } Supp_{D_1} = 0 \text{ and } Supp_{D_2} = 0 \\ \infty & \text{, if } Supp_{D_1} = 0 \text{ and } Supp_{D_2} > 0 \\ \frac{Supp_{D_2}(X)}{Supp_{D_1}(X)} & \text{, otherwise.} \end{cases}$$

■

Definition 3.2: [6] Given a growth rate threshold ρ and two datasets D_1 and D_2, an itemset X is a *ρ-emerging pattern* (ρ-`EP`) from D_1 to D_2 if $GR(X, D_1, D_2) \geq \rho$. ■

Intuitively, given two datasets, emerging patterns (`EP`s) [6] are the itemsets whose support increases significantly from one dataset to another. The formal definition of `EP`s is presented in Defintion 3.2. An emerging pattern with a growth rate ∞ (i.e., itemset that appears in one dataset but not the other) is called a *jumping emerging pattern*.

For ease of presentation, we may skip σ, ρ, D_1 and D_2 of `EP`s when they are not essential to our discussions.

Example 3.1: Figure 1 (a) shows a simplified hypothetical dataset D of the `Adult` dataset [27]. It contains some census information of the United States. More description of the dataset can be found in Section 8. We opt to present some nominal attributes of `Adult` for discussions. Each record (or transaction) represents a person. Consider two subsets D_1 containing married people and D_2 containing those who do not. From Figure 1 (a), we find the following emerging patterns, among many others.

- The pattern (`MSE`, `manager`) has a support of 75% in D_1 and 20% in D_2. Therefore, the growth rate of (`MSE`, `manager`) from D_1 to D_2 is 3.75. When we set ρ to 3, (`MSE`, `manager`) is a ρ-emerging pattern in D_2.
- High-school graduate (`HS`) has a support of 0% in D_1 but 20% in D_2. Hence, its growth rate from D_2 to D_1 is infinite. (`HS`) is a jumping emerging pattern in D_1. ■

Next, we state the formal problem statement of this paper below (Figure 1 (b)).

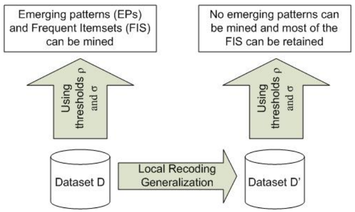
Problem statement. Given two datasets (D_1, D_2), σ and ρ, we want to sanitize (D_1, D_2) to (D'_1, D'_2) such that no ρ-EPs from D'_1 to D'_2 can be mined while the distortion between σ-frequent itemsets of (D_1, D_2) and those of (D'_1, D'_2) is minimized. ■

4 Multidimensional Local Recoding

Our algorithm for hiding emerging pattern is based on local recoding generalization (`multi-local-recode` in

ID	Edu.	Martial	Occup.	Rel.	Race	Sex
1	BA	married	executive	wife	black	F
2	MSE	married	manager	husband	black	M
3	MSE	married	manager	wife	white	F
4	MSE	married	manager	husband	black	M
5	BA	never	manager	NA	white	M
6	MSE	never	manager	NA	white	F
7	HS	never	repair	NA	black	M
8	BA	never	manager	NA	white	M
9	BA	never	manager	NA	black	F

(a)


Emerging patterns (EPs) and Frequent Itemsets (FIS) can be mined
No emerging patterns can be mined and most of the FIS can be retained
Using thresholds ρ and σ
Using thresholds ρ and σ
Dataset D
Local Recoding Generalization
Dataset D'

(b)

Figure 1. (a) A hypothetical subset of `Adult` and (b) The problem statement illustration

Figure 2). In this section, we give an overview of recoding generalization, or *recoding* for simplicity.

Recoding. As discussed in Section 1, recoding has been proposed for anonymization. The idea of recoding is to modify values into more general ones such that more tuples will share the same values and cannot be distinguished individually. Thus, anonymization can be achieved. Here, we recode values in emerging patterns with some non-emerging values. Thus, the recoded patterns become less emerging.

Multidimensional local recoding. In this work, we adopt the notion of multidimensional local recoding [8, 30, 19], from the context of k-anonymity. It recodes values at "cell level". It relies on equivalence classes. An *equivalence class* of attributes A is a set of tuples T, where $\pi_A(T)$ is a singleton. That is, the tuples in T have the same value in attributes A. In a recoding, the tuples in an equivalence class (of a set of attributes) and those in another equivalence class are recoded into the lowest common ancestors along the hierarchies. One subtle point is that this recoding does not require the entire equivalence class to be recoded, as long as anonymity can be achieved. Hence, both original and generalized values may co-exist in the recoded dataset.

Example 4.1: Let us revisit the dataset in Figure 1 (a) and the emerging pattern (`MSE`, `manager`). The emerging pattern is related to the attributes of education background (`Edu.`) and occupation (`Occup.`). Regarding (`Edu.`, `Occup.`), the equivalence classes in D_2 are {{5, 8, 9}, {6}, {7}}, where the numbers are the IDs. In multidimensional lcoal recoding, we may recode the `Edu.` attribute of the subset of {2, 3, 4} and {5, 8, 9}. For example, we may recode {2, 3, 4} with {8, 9}. and we may recode `BA` and `MSE` into degree holder `Deg`. The growth rate of (`Deg.`, `manager`) in the recoded dataset is 75%/40% = 1.875. Hence, (`Deg.`, `manager`) is not ρ-emerging when ρ = 3. In addition, after such a recoding, all `BA`, `MSE` and `Deg.` appear in the recoded dataset. ■

Other notions of recoding, including single-dimensional global recoding [5, 13, 17, 25] and multidimensional global recoding [18], generalize values in a relatively coarse granularity and very often result in over-generalization.

5 Algorithm for Hiding Emerging Patterns

In this section, we present the overall algorithm `hide-eps` (shown in Figure 2) for hiding emerging patterns with a minimal distortion in frequent itemsets.

```
Procedure hide-eps
Input: two datasets, $D_i$ and $D_j$, the threshold of growth rate and frequent
itemsets $\rho$ and $\sigma$, the heuristic parameters $p$ and $q$, an initial temperature $t_0$
and the cooling parameter $\alpha$
Output: transformed datasets ($D_i$, $D_j$)

01 $t = t_0$          $H$.init()                                   // initialization
02 $E$ := mine-eps($D_i$, $D_j$, $\rho$)                            // [31]
03 while $E \neq \emptyset$
04    $F$ := incr-mine-fis($D_i \cup D_j$, $\sigma$)                 // [?]
05    $e$ := next-overlapping-ep($E$)
06    if $e$ is not null
          ($D_i$, $D_j$) := local-recoding-sa($D_i$, $D_j$, $e$, $F$, $t$, $\alpha$, $H$)
07    if $t > 0.01$ then $t = \alpha \times t$
08    $E$ := mine-eps($D_i$, $D_j$, $\rho$)
09 return ($D_i$, $D_j$)

Procedure local-recoding-sa
Input: two datasets, $D_i$ and $D_j$, an emerging pattern $e$, a frequent itemset $F$,
a temperature $t$, a hashtable $H$ for caching
      the utility gain of local recodings
Output: transformed datasets ($D_i$, $D_j$)

10 let $D_i$ be the dataset where $e$ has a higher support
11 denote $c_e$ be the equiv. class of $e$ in $D_i$

12 compute equiv. classes $C$ of $D_j$ of the attributes of $e$
   // compute the utility gain of the local recoding of each equiv. class $c_k$ in $C$ with $c_e$
13 for each $c_k$ in $C$
14       if determine-missing-FIS($G_{(c_e,c_k)}$, $F$) = $\emptyset$ then
15          if determine-new-singleton-eps($G_{(c_e,c_k)}$, $E$) = $\emptyset$ then
16             if $H[c_e][c_k]$ is null then
17                $H[c_e][c_k]$ := util_gain($G_{(c_e,c_k)}$, $E$)        // Section 6

18 $c_k$ := get-next-step-sa($c_e$, $H$, $t$)
19 $D_i$ := multi-local-recode($D_i$, $c_e$, $c_k$)                  // Section 4
20 $D_j$ := multi-local-recode($D_j$, $c_e$, $c_k$)
21 return ($D_i$, $D_j$)
```

Figure 2. The overall algorithm

Overview of `hide-eps`. The main ideas of `hide-eps` can be described as follows. First, we determine the emerging patterns to be hidden (Line 02) and the frequent itemsets (incrementally) to be preserved (Line 04). We refer the details of Lines 02 and 04 to previous works [**?**, 31], since our focus is on *hiding* emerging patterns. For each selected emerging pattern (Line 05), we carry out a local recoding `local-recoding-sa` (Line 06, more details soon). This process (Lines 03-08) is repeated until there is no more emerging pattern to hide (Line 03). To avoid sub-optima, we present `hide-eps` in the style of simulated annealing search (Lines 01, 07 and 18).

Next, we discuss the details of the major steps of the algorithm.

Mining emerging patterns (`mine-eps`, Lines 02 and 08). During recoding, we invoke `mine-eps` [31] to determine if all the emerging patterns have been hidden (Line 08). To the best of our knowledge, there does not exist any incremental algorithm for mining emerging patterns. As verified by our experiments, `mine-eps` is a bottleneck of runtime of `hide-eps`. However, it should be remarked that the emerging patterns may often be altered slightly by most local recodings, in practice. To address this performance issue, in Section 8, we tested another version of `hide-eps`, where `mine-eps` is invoked *only* when all previously mined emerging patterns have been hidden.

Incremental mining of frequent itemsets (`incr-mine-fis`, **Line 04).** A local recoding may alter the existing frequent itemsets. Figure 3 (b) (ii) shows an example. Since a local recoding changes only part of D_1 and D_2, we need not mine the dataset from scratch but do it incrementally using algorithms like [**?**].

Selecting emerging patterns for recoding (`next-overlapping-ep`, **Line 05).** Given a set of emerging patterns E, `next-overlapping-ep` determines the emerging pattern e in E such that it overlaps with the remaining emerging patterns the most. The intuition is that reducing the growth rate of e may indirectly reduce the growth rate of the overlapping emerging patterns as well. We verify with some experiments that this approach consistently outperforms a number of other strategies (see Section 8).

Determining the next local recoding (`local-recoding-sa`, **Lines 06, 10-21).** Assume that c_e is the equivalence class of the emerging pattern e. We first compute the equivalence classes of the attributes of e to generalize with c_k (Line 12). We apply the utility gain defined in Section 6 to determine the goodness of local recodings (Line 16). Since there can be many equivalence classes, this is another bottleneck of runtime. We speed up that step using (i) a hashtable (Lines 01 and 16-17) to cache the utility gain values computed, and (ii) two filters on equivalence classes (Lines 14 and 15). The first filter is that we ignore the equivalence classes that would result in missing frequent itemsets, which is obviously undesirable. This can be computed by the change in support of itemsets in F due to a local recoding. Second, we discard a local recoding that would yield new single-attribute emerging patterns. This can be computed by determining the growth rate of the equivalence classes with the attributes of e. We did not compute possible new multi-attribute emerging patterns because of its daunting complexity.

With the utility gain of equivalence classes, we use a simulated annealing search (`get-next-step-sa`, Line 18), as a black box, to get the next local recoding.

Analysis. Given that A_E is the set of attributes of the emerging pattern E, and $\mathcal{D}$ and $\mathcal{H}$ are the overall domain size and the maximum height of the hierarchy of all possible A_E's, respectively. In the worst case, there can be $O(\mathcal{D}\times\mathcal{H})$ possible recodings. Also, local recoding allows tuple-wise recoding and thus in the worst case, $|D_1|$ + $|D_2|$ recoding operations can be carried out. Thus, the search space of finding the optimal recoding is $O((|D_1| + |D_2|) \times \mathcal{D} \times \mathcal{H})$. This work proposes a heuristic search for this problem. While the loop (Lines 03-08) may repeat many times in the worst case, the number of iterations needed was found small in practice. As discussed, `mine-eps` and the computation of `util_gain` are the bottlenecks of runtime. The runtime of the former is experimentally evaluated in [31]. The time complexity for the latter is $O(|A_e| \times \mathcal{D} \times \mathcal{H} \times |F|)$, where $e \in E$, $|A_e| \times \mathcal{D} \times \mathcal{H}$ is the number of possible equivalence classes and for each class, $O(|E|)$ and $O(|F|)$ are used to compute RG_{local} and RD_{local}, respectively.

6 Metric for Multidimensional Local Recoding

In this section, we define an utility gain (`util_gain`) to quantify the effectiveness of a local recoding. `util_gain` will guide the process for hiding emerging patterns `local-recoding`, in Figure 2. A recoding is effective if (i) the distortion of frequent itemsets is small and (ii) the reduction in the growth rate of emerging patterns is large.

Metric for the distortion of frequent itemsets. For presentation clarity, we will present our proposed metric for global recoding followed by its adaption for local recoding.

(A) Distortion metric for single-dimensional global recoding. Single-dimensional global recoding performs recoding on the domain of an attribute in a dataset. It recodes a value of the domain to another (generalized) value. That is, if a particular value is recoded, the attribute of all the tuples containing that particular value will be recoded. No frequent itemsets disappear but may appear in a generalized form after a recoding (Figure 3 (a)).

Inspired by the distortion metric proposed in [19], we propose a metric for measuring the *recoding distance* ($RDist$) between the original and generalized form of a tuple. Then, we define a metric called *value distance* (VD) which measures the distance between the original and generalized form of a single attribute value. We will use VD as a building block for the definition of distortion (RD). Since a recoding always assumes an attribute hierarchy, we may skip the hierarchy H when it is clear from the context.

Definition 6.1: *Recoding Distance (RDist):* Consider a recoding G which generalizes a set of non-generalized values V to a single generalized value v_g, where V is the set of values under v_g in an attribute hierarchy. The recoding distance of G $RDist(G)$ is $|V|$. ■

Definition 6.2: *Value Distance (VD):* Let h be the height of an attribute hierarchy H, where level h and 0 is the most generalized and specific level, respectively. Consider a value v at level p which is generalized to a value v' at level

q. Let G_i denotes the recoding that generalizes an attribute from level $i-1$ to i, where $0 < i \leq h$. The *value distance* between v and v' is: $VD(v, v') = \sum_{i=p}^{q} \frac{i \cdot RDist(G_i)}{h}$. ■

Value distance is unfavorable to recoding (i) many values into one single generalized value; and (ii) a value into a generalized value that is close to the top of the hierarchy. This gives a measure for the distortion of a value due to a recoding. Next, we extend VD to measure the distortion of a tuple and frequent itemsets due to recoding.

Definition 6.3: *Tuple Distance (TD):* Suppose a tuple $f = (v_1, v_2, \ldots, v_n)$ is generalized to $f' = (v'_1, v'_2, \ldots, v'_n)$. The tuple distance between f and f' is defined as: $TD(f, f') = \sum_{i=1}^{n} VD(v_i, v'_i)$. ■

Definition 6.4: *Recoding Distortion (RD):* Let F = $\{f_1$, f_2 $\ldots$ $f_n\}$ be a set of *σ-frequent itemsets* in D and F' = $\{f'_1$, f'_2 $\ldots$ $f'_m\}$ be the set of *σ-frequent itemsets* in D', where $m \leq n$. The corresponding frequent itemset of f_i due to global recoding is denoted as f'_j = $G(f_i)$. The recoding distance between F and F' is defined as: $RD(F, F') = \sum_{i=1}^{n} TD(f_i, G(f_i))$. ■

Example 6.1: Following up Example 4.1, we compute the (global) recoding distortion of generalizing (MSE, manager) to (Deg., manager). Figure 3 shows the attribute hierarchy of Edu.The recoding distortion $RD(\{$(MSE, manager)$\}$, $\{$(Deg., manager)$\}$), RD, can be computed as follows: $RD = TD((\texttt{MSE}, \texttt{manager}), (\texttt{Deg.}, \texttt{manager})) = VD(\texttt{MSE}, \texttt{Deg.}) + VD(\texttt{manager}, \texttt{manager}) = \sum_{i=0}^{2} \frac{i \cdot RDist(G_i)}{h} = \frac{1\times3}{2} + \frac{2\times0}{2}$ = 1.5 ■

(B) Distortion metric for local recoding. Since single-dimensional global recoding may often lead to over-generalization, we adopted local recoding. We remark that there are two unique challenges in computing recoding distance for local recoding (Figure 3 (b)).

(B.i) An itemset in F having no correspondence in F'. Local recoding allows part of the tuples that share the same attribute values to be generalized. Such recoding may generalize some supporting tuples of a frequent itemset which makes the itemset (in the original or generalized form) not frequent anymore. To address this, we measure the distortion of the disappeared frequent itemset to the most general form. The reason is that the frequent itemset can be trivially recovered when the entire dataset is generalized to the most general form.

Specifically, given a f in F, if we cannot find a corresponding frequent itemset in F', we first create an itemset, f_{max}, which contains the most generalized value of each value in f. Then, RD of f is the recoding distance between f and f_{max}.

Example 6.2: Reconsider the dataset in Figure 1 (a). Suppose we recode the Edu. attribute of Records 1 and 2 to Deg. When σ is 40%, $\{$BA$\}$ and $\{$MSE$\}$ were frequent itemsets (not minimal for illustration purposes) before recoding and there is no frequent itemset after recoding. ■

(B.ii) An itemset in F having more than one corresponding itemset in F'. As discussed, local recoding may generalize a frequent itemset f in F into more than one correspondence in F', denoted as F_f. In this case, we calculate the tuple distance of each of the corresponding itemsets in F_f and take the *minimum* tuple distance as the tuple distance of f. This is because the itemset with the minimum tuple distortion has been revealed in F', even when there may be more distorted itemsets.

With the above, we have the following recoding distance for local recoding:

Definition 6.5: *Recoding Distance for Local Recoding (RD_{local}):* Let F = $\{f_1$, f_2 $\ldots$ $f_n\}$ be a set of *σ-frequent itemsets* in D and F' = $\{f'_1$, f'_2 $\ldots$ $f'_m\}$ be the set of *σ-frequent itemsets* in D'. The corresponding frequent itemset(s) of f_i due to local recoding is denoted as F_f = $G(f_i)$. The recoding distance between F and F' is: $RD_{local}(F, F')$= $\frac{1}{n}\sum_{i=1}^{n} \frac{TD_{local}(f_i, G(f_i))}{TD_{local}(f_i, f_{max})}$, where $TD_{local}(f_i, G(f_i))$ =

$$\begin{cases} \theta_q \times TD(f_i, G(f_i)), & \text{if } f \text{ has 1 correspondent in } F' \\ (1-\theta_q) \times TD(f_i, f_{max}), & \text{if } f \text{ has no correspondent in } F' \\ \theta_q \times min(TD(f_i, f_j)), & \\ \qquad\qquad where\ f_j \in G(f_i), & \text{otherwise,} \end{cases}$$

θ_q is a parameter that specifies the relative importance of the itemset distortion and missing itemsets, due to G, and $TD_{local}(f_i, f_{max})$ is for normalizing RD_{local}. ■

Example 6.3: Following up Example 6.2, when σ is 30%, the frequent itemset $\{$(BA)$\}$ corresponds to the frequent itemsets $\{$(BA), (Deg)$\}$ in the recoded datasets. ■

Metric for the change in growth rate. The second component of our heuristics concerns the growth rate of the emerging patterns. Intuitively, we aim at a recoding that significantly reduces the growth rate of the emerging patterns in order to hide them. Given an emerging pattern e and the result of a local recoding e', the reduction in growth rate due to the recoding can be easily defined as the growth rate of e minus the growth rate of e'. Then, the growth rate reduction of E due to a local recoding G, denoted as $RG_{local}(G, E)$, can be defined as the total reduction in the growth rate of e in E divided by the total growth rate of e in E.

Putting all these together. Based on the derivations above, the utility gain due to a local recoding G for a set of emerging patterns E is defined as:

$\texttt{util\_gain}(G, E) = \theta_p RG_{local}(G, E) - (1-\theta_p) RD_{local}(F, F')$. The two parameters θ_p and θ_q, where θ_p, θ_q $\in$ [0,1], are specified by users.

7 Implementation Optimization

In this section, we discuss some implementation issues for optimizing the computation of the proposed algorithm.

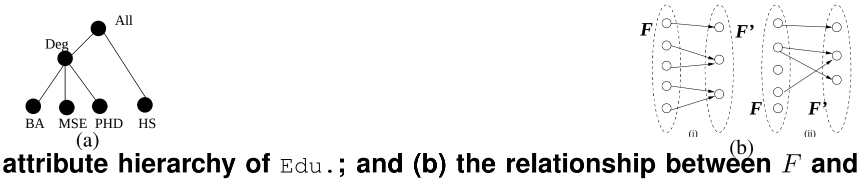


Figure 3. (a) An attribute hierarchy of `Edu.`; and (b) the relationship between F and F' in (i) global recoding and (ii) local recoding

1. The maintenance of the equivalence classes. Local recoding relies on equivalence classes (Lines 03, 06 and 07 in `local-recoding` in Figure **??**). A local recoding would change the equivalence classes in the datasets slightly. We used a hashtable to keep track of the update of the classes caused by a recoding (Line 07). Since an emerging pattern may not be hidden by simply one recoding, we note that we may compute the equivalence class of attributes multiple times (Line 03). With the hashtable, Line 03 does not recompute existing equivalence classes.

2. An index for checking correspondents of F in F'. Given two sets of frequent itemsets F and F', we check the correspondences of an itemset in F to measure distortion between F and F' (`util_gain`). The core of this is to check whether a value v is a generalized version of another value in an attribute hierarchy. While an attribute hierarchy is often a small tree, these checks occur in every iteration of `hide-eps`. We apply an index [29] for computing the ancestor-descendant relationship of nodes in a tree. This significantly reduces the runtime of our algorithm.

3. A data structure for checking correspondents of E in E'. The other task in computing `util_gain` is to keep track of the change of emerging patterns during local recoding, which is necessary to measure the reduction in growth rate. Hence, we associate an emerging pattern of e to its records. By comparing the records of e and e', we obtain the correspondence between e and e'.

8 Experimental Evaluation

To verify the effectiveness and efficiency of our proposed algorithms, we conducted several experiments on `Adult` dataset [27] using the attribute hierarchies from [13].

We implemented our algorithm in JAVA SDK 1.6[1]. We have run our experiments on a PC with a Quad CPU at 2.4GHz and 3.25GB RAM. The PC is running Windows XP operating system. We have used system calls to invoke the implementations from [31] and [22] to determine emerging patterns and frequent itemsets, respectively.

The simplified `Adult` dataset contains 8 attributes. We removed the records with missing values. The records in the dataset were divided into two classes - people who have more than \$50k/year (7508 records) and people who do not (22654 records).

The effect of the parameters θ_p and θ_q. The first experiment is to verify the effects on the parameters θ_p and θ_q on the heuristic algorithm. In this experiment, we do not apply any filter (i.e., Lines 14-15 in `hide-eps`) and `SA` search (Line 18) in order to observe the effects on the parameters clearly. Instead, we used a `Greedy` search. The *performance* was presented in *"distortion on the frequent itemsets / the number of missing frequent itemsets", unless otherwise specified.* When σ and ρ were set to 40% and 5, respectively, the frequent itemsets obtained are: {(`Husband`, `Married-civ-spouse`, `Male`), (`Married-civ-spouse`, `White`), (`Married-civ-spouse`, `United-States`), (`Male`, `Private`, `White`), (`Male`, `Private`, `United-States`), (`Male`, `White`, `United-States`), (`Private`, `White`, `United-States`)}.

When we recode all attributes to `All` in the frequent itemsets, we obtain the maximum distortion of the frequent itemsets of `Adult` 623.1.

To illustrate the possible effect of recodings on the resulting frequent itemsets, we list the frequent itemsets after we applied `Greedy`, where θ_p and θ_q were both set to 0.8: {(`Relationship`, `United-States`), (`Married`, `White`, `United-States`), (`Male`, `Private`, `White`), (`Male`, `Private`, `United-States`), (`Male`, `White`, `United-States`), `Private`, `White`, `United-States`)}. The distortion obtained is 21.5 (out of 623.1).

Next, we varied θ_p and θ_q and measured the performance of `Greedy`. The performance is shown in Table 1 (LHS). The average runtime is 58 mins and 16 out of 58 mins is spent on mining `EP`s. The average number of local recodings is 14.

We make four observations from Table 1 (LHS). Firstly, when θ_p is set to 0, the algorithm concerns only distortion (regardless the corresponding reduction in growth rate) during local recodings. In such a case, the distortion on frequent itemsets of various θ_q's is in general large. The reason is that when θ_p is 0, the heuristics does not effectively reduce the growth rate and the search takes more recodings that are not relevant to hiding emerging patterns. Secondly, when θ_p is 1, the algorithm concerns only the reduction in growth rate. Note that 3 out of 7 frequent itemsets have disappeared. Thirdly, when we set θ_q to 1, we do not concern the missing frequent itemsets. Hence, more frequent item-

[1] The implementation is available at http://www.comp.hkbu.edu.hk/∼michael/source.rar.

Table 1. The effect of the parameters in util_gain on Greedy's performance (LHS) and the performance of the determine-new-singleton-eps filter (RHS)

$\theta_p \backslash \theta_q$	0.0	0.2	0.4	0.6	0.8	1.0
0	73.3/1	50.0/1	50.0/1	50.0/1	50.0/3	50.0/1
0.2	73.3/1	59.7/1	38.2/1	38.2/1	46.5/1	11.5/4
0.4	73.3/1	59.7/1	38.2/1	21.5/1	46.5/1	11.5/4
0.6	73.3/1	59.7/1	21.5/1	38.2/1	46.5/1	11.5/4
0.8	73.3/1	59.7/1	21.5/1	21.5/1	38.2/1	0/5
1.0	11.5/3	11.5/3	11.5/3	11.5/3	11.5/3	11.5/3

$\theta_p \backslash \theta_q$	0.0	0.2	0.4	0.6	0.8	1.0
0	62.7/0	43.7/1	43.7/1	35.8/3	35.8/3	11.1/4
0.2	62.7/0	32.0/1	32.0/1	15.7/2	9.7/3	7.5/4
0.4	62.7/0	32.0/1	16.8/1	21.9/2	9.7/3	7.5/4
0.6	71.5/0	32.0/1	16.8/1	21.9/2	7.2/3	0/5
0.8	71.5/0	32.0/1	16.8/1	15.7/2	7.2/3	0/5
1.0	73.1/1	73.1/1	73.1/1	73.1/1	73.1/1	73.1/1

sets were lost. Similarly, when we set θ_q to 0, we concern only the frequent itemsets that do not disappear. Since there is one missing frequent itemset during recodings, overlooking this led to more distortion. Fourthly, we found that a significant runitme (42 mins) was spent on calculating the utility gain of equivalence classes. The reason is that no filters had been applied yet.

In all, we found that on Adult, Greedy yields frequent itemsets with a distortion 21.5 (out of 623.1) and 1 missing frequent itemset when both θ_p and θ_q are moderate.

The effect of the determine-new-singleton-eps filter. This filter is used to avoid recoding equivalence classes that would yield new single-attribute EPs (Line 15 of hide-eps). The performance of Greedy with this filter is shown in Table 1 (RHS).

We observe from Table 1 (RHS) that there are similar trends on the performance with various θ_p and θ_q. The distortion is sometimes smaller but the missing frequent itemsets may sometimes be more. However, the average runtime of this experiment is 26 mins (compared to 58 previously). Specifically, the time for computing utility gain has been reduced from 42 to 15 mins. The number of recodings reduces from 14 to 9. The runtime improvement is due to (i) the smaller number of equivalence classes for computing the utility gain and (ii) fewer (if any) new EPs generated during hide-eps.

The effect of the determine-missing-FIS filter. From the previous experiments, we note that there are missing frequent itemsets in most cases. Here, we test the effectiveness of determine-missing-FIS filter (Line 14). The performance is shown in Table 2 (LHS). The average runtime for computing the utility gain increased from 15 to 21 mins and the number of recodings increased from 9 to 14. At first glance, the distortion might have increased. However, there is no missing frequent itemset for all θ_p's and θ_q's. This improvement comes at the expense of a slight increase in runtime.

Table 2. The performance of the determine-missing-FIS filter (LHS) and the performance of invoking mine-eps only when E is empty (RHS)

$\theta_p \backslash \theta_q$	0.0	0.2	0.4	0.6	0.8	1.0
0	NA	89.2/0	89.2/0	89.2/0	71.5/0	71.5/0
0.2	105.3/0	89.2/0	78.6/0	78.6/0	41.5/0	41.5/0
0.4	105.3/0	78.6/0	50/0	50/0	41.5/0	41.5/0
0.6	105.3/0	78.6/0	50/0	50/0	41.5/0	41.5/0
0.8	105.3/0	78.6/0	61.3/0	61.3/0	41.5/0	41.5/0
1.0	105.3/0	105.3/0	105.3/0	105.3/0	105.3/0	105.3/0

$\theta_p \backslash \theta_q$	0.0	0.2	0.4	0.6	0.8	1.0
0	NA	97.2/0	97.2/0	81.3/0	70.1/0	81.3/0
0.2	105.3/0	97.2/0	97.6/0	81.3/0	59.3/0	59.3/0
0.4	105.3/0	97.6/0	64.9/0	64.9/0	59.3/0	59.3/0
0.6	105.3/0	78.6/0	64.9/0	64.9/0	59.3/0	59.3/0
0.8	105.3/0	78.6/0	64.9/0	70.1/0	59.3/0	59.3/0
1.0	105.3/0	105.3/0	105.3/0	105.3/0	105.3/0	105.3/0

Table 3. The performance of hiding the EP with the minimum overlapping (LHS) and the performance of simulated annealing search (RHS)

$\theta_p \backslash \theta_q$	0.0	0.2	0.4	0.6	0.8	1.0
0	NA	101.8/0	101.8/0	95.6/0	95.6/0	95.6/0
0.2	127.1/0	98.3/0	75.8/0	75.8/0	53.7/0	53.7/0
0.4	127.1/0	98.3/0	50.0/0	50.0/0	53.7/0	53.7/0
0.6	127.1/0	78.6/0	67.2/0	58.5/0	53.7/0	53.7/0
0.8	127.1/0	80.2/0	61.3/0	65.3/0	48.1/0	48.1/0
1.0	127.1/0	127.1/0	127.1/0	127.1/0	127.1/0	127.1/0

$\theta_p \backslash \theta_q$	0.0	0.2	0.4	0.6	0.8	1.0
0	67.4/0	27.8/0	58.5/0	50.0/0	44.3/0	23.6/0
0.2	80.1/0	62.5/0	22.4/0	53.8/0	47.8/0	52.1/0
0.4	84.1/0	81.4/0	55.7/0	29.3/0	53.7/0	37.4/0
0.6	85.0/0	50.0/0	97.0/0	31.8/0	40.5/0	22.8/0
0.8	64.9/0	45.2/0	30.0/0	59.6/0	39.5/0	32.3/0
1.0	84.1/0	31.7/0	47.9/0	44.1/0	28.9/0	51.6/0

The effect of calling mine-eps when E is empty. In the last experiment, 15 out of 36 mins was spent on mining EPs. In this experiment, we attempt to improve the runtime by hiding all existing EPs first before calling mine-eps, as opposed to calling mine-eps after each recoding. The performance is shown in Table 2 (RHS). From the result, we found that there is a slight increase in distortion. However, the time for mining EPs is reduced from 15 to 8 mins. The average runtime for computing the utility gain increased from 21 to 23 mins and the number of iterations remains unchanged.

The effect of hiding the EP with the minimum overlapping. To justify the decision of hiding the maximum overlapping EP in Line 05 of hide-eps, we conducted an experiment which first hides the EP with the minimum overlapping. The result is shown in Table 3 (LHS). We observed that the distortion is slightly larger than that of the maximum overlapping. However, the average runtime for computing the utility gain increased from 23 to 39 mins and the time for mining EP increased from 8 to 23 mins.

Simulated annealing search. After demonstrating the effects of various settings with Greedy, we applied SA on the algorithm (Line 18 of hide-eps). We set a low temperature (T=10) of SA with a high cooling rate (α=0.4). Hence, SA initially has some chances to avoid local sub-optima and then converges to Greedy quickly. To explore the search space more, each SA was allowed to restart fifty times. The results are shown in Table 3 (RHS). SA introduces some randomness in the performance. Compared to the best versions (Table 2), SA often produces better results, at the expense of runtime.

9 Conclusions

We presented a heuristic local-recoding algorithm for hiding emerging patterns of a dataset while preserving its frequent itemsets as far as possible. We tested our algorithm with a benchmark dataset and showed its effectiveness.

References

[1] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[2] D. Agrawal and C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of PODS*, 2001.

[3] J. Bailey, T. Manoukian, and K. Ramamohanarao. Fast algorithms for mining emerging patterns. In *Proc. of ECML/PKDD*, 2002.

[4] J. R. Bayardo. Efficiently mining long patterns from databases. In *Proc. of SIGMOD*, pages 85–93, 1998.

[5] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proc. of ICDE*, pages 217–228, 2005.

[6] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. of SIGKDD*, pages 43–52, 1999.

[7] G. Dong, X. Zhang, and L. Wong. CAEP: Classification by aggregating emerging patterns. In *Proc. of DS'99*, pages 30–42, 1999.

[8] Y. Du, T. Xia, Y. Tao, D. Zhang, and F. Zhu. On multidimensional k-anonymity with local recoding generalization. In *Proc. of ICDE*, pages 1422–1424, 2007.

[9] A. Evfimievski, R. Strikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proc. of SIGKDD*, 2002.

[10] H. Fan and K. Ramamohanarao. A Bayesian approach to use emerging patterns for classification. In *Proc. of ADC*, pages 39–48, 2003.

[11] B. Fung, K. Wang, A. Fu, and P. Yu. *Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman & Hall/CRC, 2010.

[12] B. Fung, K. Wang, L. Wang, and M. Debbabi. A framework for privacy-preserving cluster analysis. In *Proc. of ISI*, page 4651, 2008.

[13] B. Fung, K. Wang, and P. Yu. Top-down specialization for information and privacy preservation. In *Proc. of ICDE*, pages 205–216, 2005.

[14] B. Fung, K. Wang, and P. Yu. Anonymizing classification data for privacy preservation. *TKDE*, 10(5):711–725, 2007.

[15] H. F. K. Ramamohanarao. Pattern based classifiers. In *Proc. of WWW*, pages 71–83, 2007.

[16] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *KAIS*, 7(4):387–414, 2005.

[17] K. LeFevre, D. Dewitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of SIGMOD*, pages 49–60, 2005.

[18] K. LeFevre, D. Dewitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Proc. of ICDE*, page 25, 2006.

[19] J. Li, R. Wong, A. Fu, and J. Pei. Anonymization by local recoding in data with attribute hierarchical taxonomies. *TKDE*, 20(9):1181–1194, 2008.

[20] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *Proc. of SIGKDD*, 2009.

[21] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Vẽnkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *TKDD*, 1(1):3, 2007.

[22] MAFIA. *Mining Maximal Frequent Itemsets*. *http://himalaya-tools.sourceforge.net/Mafia/.*

[23] G. Moustakides and V. Verykios. A maxmin approach for hiding frequent itemsets. *DKE*, 65(1):75–79, 2008.

[24] S. Oliveira and O.R.Zaiane. Privacy preserving frequent itemset mining. In *Proc. of ICDM Workshop on Privacy, Security and Data Mining*, volume 14, pages 43–54, 2002.

[25] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *IJUFKS*, 10(5):571–588, 2002.

[26] L. Sweeney. k-anonymity: A model for protecting privacy. In *IJUFKS*, pages 557–570, 2002.

[27] UCI Machine Learning Repository. *Adult Data Set*. *http://archive.ics.uci.edu/ml/datasets/Adult.*

[28] Z. Wang, H. Fan, and K. Ramamohanarao. Exploiting maximal emerging patterns for classification. In *Proc. of AUS-AI*, pages 1062–1068, 2004.

[29] X. Wu, M. L. Lee, and W. Hsu. A prime number labeling scheme for dynamic ordered xml trees. In *Proc. of ICDE*, pages 66–78, 2004.

[30] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. Fu. Utility-based anonymization using local recoding. In *Proc. of SIGKDD*, pages 785–790, 2006.

[31] X. Zhang, G. Dong, and K. Ramamohanarao. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *Proc. of SIGKDD*, pages 310–314, 2000.

Enhanced Location Estimation in Wireless LAN environment using Hybrid method

Kevin C. Shum, and Joseph K. Ng

Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
cyshum,jng@comp.hkbu.edu.hk

Abstract

Location Estimation in Wireless LAN (WLAN) environment become a importance part for network administrator to monitor the traffic within the network. 2.4Ghz 802.11(G/B) Wireless Router become more popular. Most of them can run an open-source firmware like dd-wrt, open-wrt in order to obtain useful information like Received Signal Strength (RSS) for positioning. Programmable firmware like open-wrt for Linksys WRT54G[2] is handy for us to store important Location Fingerprint (LF) and RF propagation loss model(PL) Location information into database, which helping network administrator to monitor the wireless network for surveillance purpose.

1 Introduction

Recent electronics, computer and wireless device become more in-expensive, low-power usage and multi-functional. Recent open-sourced router have fast data processor in order to do the best job in data retrieval for wireless network surveillance as well as wireless location estimation.

A programmable Linksys WRT54G burned with open-sourced custom-made firmware is act as a wireless sensor in order to obtain information from the wireless environment, Service Set Identifier (SSID), Extended Service Set Identifier (ESSID), Signal Strength (RSSI) and Noise Level. A custom-made wireless data-retrieval application is written for the WRT54G, a cross-compiled binary for 32-bit MIPS architecture processors manufactured by Broadcom is generating the best information to archieve our goals.

The RSSI information obtained from the AP is varying in a fixed position, due to interference, multi-path effect etc. However, according to the data collected from the AP, we find some hints on the effect of the interference which may helps estimating a position of the system.

2 Requirements

For Location Fingerprint (LF)[6], a server machine to store the database which hold the offline-phrase training data, time-stamp, ESSID and RSSI etc. A trained data is used to estimate the position for the on-line user.

Multiple Linksys WRT54G burned with a cross-compiled binary is used to store wireless information into database server.

A Online and Off-line trained data also act as a surveillance system to let the network administrator to monitor the network behavior.

Mobile Device which is WLAN enabled to obtain RSSI information from AP. Nokia N96 which running Symbian 3.2 is programmed to extract WLAN information. When the mobile device is associated to AP, both side can obtain a RSSI from each other.

RF propagation loss model(PL)[1] is used to calibrate and enhance the Location Fingerprint (RF) in the future works.

3 Positioning Technique

3.1 Location fingerprinting (LF)

Location fingerprinting method requires a training dataset, which is the collection of data (Fi, Li), i = 1, . .

. , N, for N locations in the site, where Li is the known location of the ith measurement and Fi = (Fi1, . . . , FiN) is the RSSI vector when the AP is at Ci. Fi is the fingerprint of the location Li. When a new fingerprint F is observed from AP with unknown location A, search the database fingerprint Fi that is closest to fingerprint, and so we can estimate fingerprint F and location L.

3.2 RF propagation Loss Model and calibration

A Free space path loss equation:
Lp(db)=20*LOG(f)+20*LOG(d)-function(fx)

d is distance in m
f is frequency in Mhz (i.e. about 2400 in 802.11 standard due to difference channel)
function(fx) is the signal loss function due to obstacles

Calibration for function(fx) is needed, some component and module manufacturers use this model to predict their operating range, but sure can no be see in the real world. Multi-path fading and interference, the real world is full of obstacles that absorb, reflect, and scatter RF energy, including the earth itself. Looking for these factors to attempt to create a better pat loss model to predict a real world Location Estimation. In addition to multi-path fading, ISM band radios are also subject to interference from other unlicensed radios, raising the noise floor in the transmission channel. If the noise floor is too high, the receiver effective and performance will be dropped.

4 Experiment

4.1 Experimental Testbed Preparation 1

Figure 1 shows our Experimental Test bed with obstacles in RRS 716, the experimental laboratory. The laboratory detected about 25 APs to demonstrates interferences. There are three AP obtains a signal Data, which is BUAP7, BUAP9 and BUMAIN, others are the mobile device. The number in the middle is the distance unit between the AP and the mobile device. For example, distance from BUAP7 to BUAP1 is 129 units. About 50000 Samples was taken for every mobile device. The Distance and the received RSSI is used to plot the graphic.

4.2 Experimental Testbed Preparation 2

Figure 2 shows our Experimental Test bed without obstacles in the corridor outside RRS716. The laboratory detected about 25 APs to demonstrates interferences, about 500 Samples was taken for more than 10 distances, distance unit marked has been taken a sample for analysis.

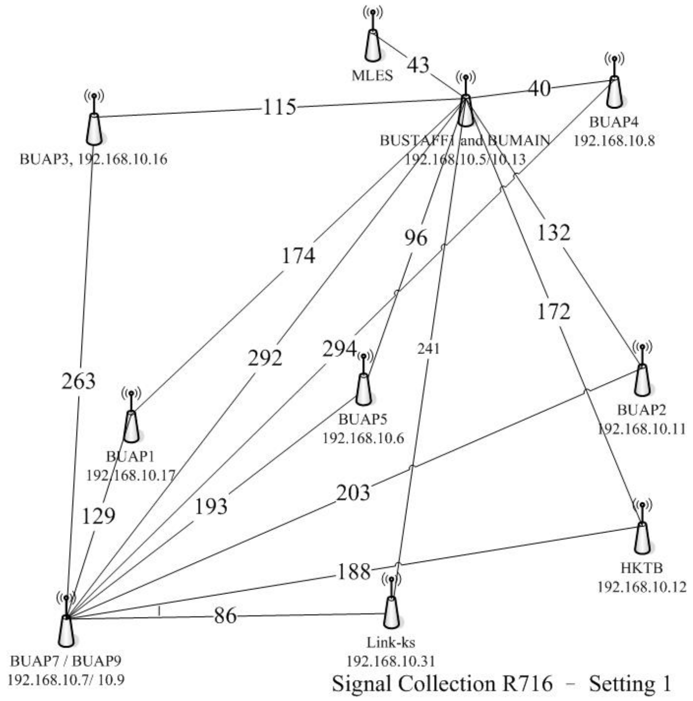


Figure 1. Testbed Preparation 1

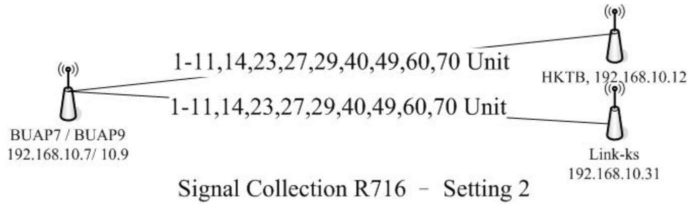

1-11,14,23,27,29,40,49,60,70 Unit
HKTB, 192.168.10.12
BUAP7 / BUAP9
192.168.10.7/ 10.9
1-11,14,23,27,29,40,49,60,70 Unit
Link-ks
192.168.10.31
Signal Collection R716 – Setting 2

Figure 2. Testbed Preparation 2

4.3 Experimental Testbed Preparation 3

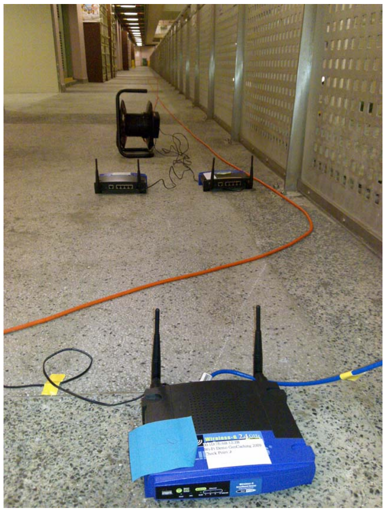


Figure 3. Testbed Preparation 3

Figure 3 shows our Experimental Test bed without obstacles in the outside environment, only 4 APs is detected, about 100 samples was taken for more than 10 distances, distance unit marked has been taken a sample for analysis.[5]

5 Test Result and Analysis

5.1 Result and analysis - Testbed 1

Samples was taken for BUAP1, we recorded -31,-33,-45 RSSI level most frequency. The distance between BUAP7 and BUAP1 is 129 Units(i.e. 10.75ft). Figure 4 and 5 shows the graph for RSSI level and its frequency, sorted by RSSI and its frequency.
In Figure 4, the result is sorted by RSSI, three peak is observable, RSSI -31 has most frequency of 11092, the second peak is -45, but the frequency is not higher then RSSI -33, and the third peak is -63. From the experiment, we can see the there are multi-path, construction interference and de-constructive interference may occur inside the test bed. This raise a importance information for the future works, as those peaks RSSI are evenly distritbuted within the spectrum, some computation can be made to identify the wireless environment(i.e. indoor environment / inside the crowded room).[3]

5.2 Result and analysis - Testbed 2

Samples taken for 11ft, which show highest frequency recorded for RSSI is -58. This test bed is used to computet

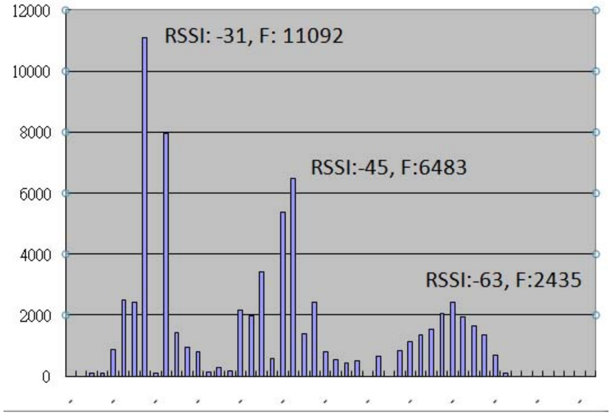


Figure 4. Testbed 1

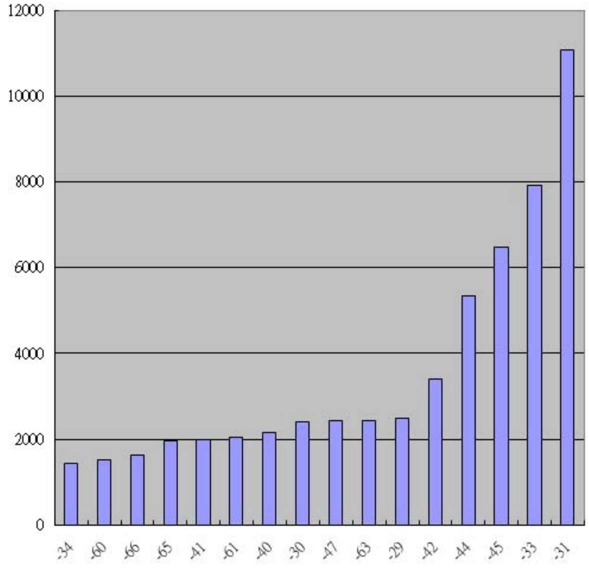


Figure 5. Testbed 1 Sorted

the function(fx) for the RF propagation Loss Model (PL). If we don't have the exact X,Y information of the AP, we can still estimate the distance between the AP, which act as a wireless surveillance monitor. In this situation, the function(fx) is
Lp(58)=20*LOG(2425)+20*LOG(3.3528)-function(fx)
i.e. function(fx)= 20.202

This function can calculate when we receiving average of RSSI -63, d=5.961m (19.55ft).
The median distance error slightly, from 1 m to 3 m.
The result is not going to be valid, for RSSI -63, the distance should around 27ft from my training data, this is the result why PL may not be accurate when using one AP for location estimation.
We should design how function(fx) work in dynamic environment in the future.

5.3 Result and analysis - Testbed 3

For outdoor environment, the PL demonstrate better performance, the RSSI spectrum is much narrow then indoor, the RSSI peak is obversely. For Figure 6, the average RSSI for 46ft, from -52 to -55, which is easier to identify, and less dynamic to the PL equation function(fx).

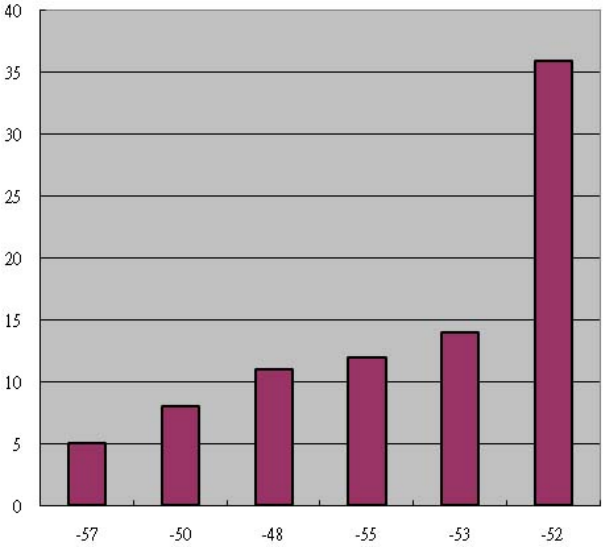


Figure 6. Testbed 3 Sorted

6 Experimental Result

In Testbed 1, there are 3 signal retrieval points, and total of 9 fixed mobile device, a total of 10 training points. The training process is placing the mobile device at a particular location, the AP is receiving RSSI and store into the database.
Referring to Figure 4, the RSSI spectrum spread across intensively from -27 to -34, -41 to -43 and -56 to 64. Three peaks are obtained from the result, the phenomenon is important to the future work, as it can be deviated and compute how the RSSI will look like in some indoor situation. For indoor situation, the RSSI spectrum is wide and fluctuated, on the other hand, for outdoor situation, RSSI spectrum is narrower and RSSI level is more stable. In order to made PL equation to be trustable, further study on the behavior of how the function(fx) varying is a must.
Multi-path fading and people's activities lead the RSSI fluctuated. According from the dataset collected from the fingerprint measurement, various signal statistical character is observed.
Figure 7, showing the multipath affecting the relationship with signal and distance. Multi-path will lead to the distance longer then the direct one and causes signal diminished. The distribution of the RSSI spectrum is varied by those kind of effects.

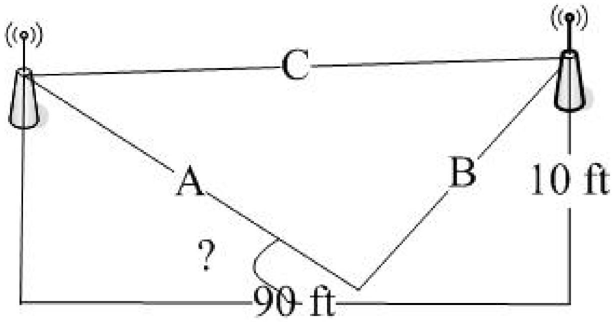


Figure 7. Multipath

7 Conclusion and future work

Enhance the WLAN Location technique by observing signal statistical character[4] has been introduced. By combining the technique with FP and PL, location estimation can be applied and evaluated. The future work is to expose more factor affecting the RSSI, combine the factor to enhance the location estimation.
Difference device and wireless electronic component release difference power of signal, a calibration system for the fingerprint is proposed to enhance the presence positioning method. Afterward, a wireless network surveillance monitoring system can be proposed.

References

[1] Hermersdorf, M. Indoor positioning with a WLAN access

point list on a mobile device. In Proc. WSW 2006 at SenSys 2006 (October 2006).

[2] http://en.wikipedia.org/wiki/Linksys_WRT54G_series.

[3] Kaemarungsi, Kamol and Prashant Krishna-murthy (2004): Modeling of indoor positioning systems based on location fingerprinting IEEE INFOCOM 2004 - The Conference on Computer Communication, vol. 23, no. 1, March 2004, pp. 1013-1023.

[4] Location Estimation in ZigBee Network Based on Fingerprinting, Qingming Yao, Fei-Yue Wang, Fellow, IEEE, Hui Gao, Kunfeng Wang and Hongxia Zhao.

[5] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg, COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses, in Proc. The First International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization(WiNTECH 06) , pp. 34-40, Sep. 2006.

[6] C. di Flora and Hermersdorf. M. a practical implementation of indoor location-based services using simple wifi. *Journal of Location Based Services 2,*, pages 87–111, June 2008.

Concept Discovery in Youtube.com using Factorization Method

Janice Kwan-Wai Leung

Abstract

Social media are not limited to text but also multimedia. Dailymotion, YouTube, and MySpace are examples of successful sites which allow users to share videos and interact among themselves. Due to the huge amount of videos, categorizing videos with similar contents can help users to search videos more efficiently. Unlike the traditional approach to group videos into some predefined categories, we propose to facilitate video searching with clustering from comment-based matrix factorization and to improve indexing via the generation of new concept words. Factorized component entropies are introduced for handling the difficult problem of vocabulary construction for concept discovery in social media. Since the categorization is learnt from users feedback, it can accurately represent the user sentiment on the videos. Experiments conducted by using empirical data collected from YouTube shows the effectiveness of our proposed methodologies.

1 Introduction

In the Web 2.0 era, people can interact effectively in the Internet instead of just retrieve data. Social networks such as forums, blogs, video sharing sites are examples of applications. Social networks like Facebook [3], Bebo[1], Flickr [4] are blooming with user generated contents which can be in forms outside text such as images or videos.

Recent years online video sharing systems are burgeoning. In video sharing sites, users can upload and share videos with other users. YouTube [6] is one of the most successful and fast-growing systems. In YouTube, users can share their videos in various categories. Among these video categories, music is one of the most popular one and the number of music videos overly excess that of other categories [9] [19]. Users are not only allowed to upload videos but tag videos but leave comments on them as well. With more than 65,000 new videos being uploaded every day and 100 million video views daily, YouTube becomes a representative community among video sharing sites [7].

Due to the incredible growth of video sharing sites, video searching is no longer a easy task and more effort should be paid by users to search their desire videos from the entire video collection. To address this problem, grouping videos with similar contents together and indexing are necessary. As such, information about video content is needed for the objective mentioned above. However, it is an even challenging problem to find out accurate information about the uploaded videos. Currently, videos on YouTube are only coarsely grouped into some predefined high level categories (e.g. music, entertainment, sports, etc) in which category of a video is just decided by a single user who put up the video. Under this policy of predefining categories, videos in a single category still span through a wide range of varieties. For example, in the music category, we may find music from various countries or with different musical styles. Though some other video sharing sites, such as DailyMotion [2] and MySpace [5], have a lower level of category for music videos, the categorization just follow the basic music genre. However, people attentions to music are not limited to these simple genre. Furthermore, the predefined categories maybe too subjective to capture the real attracted issue of singers to the majority of users since they are only defined by a small group of people. Finally, the current categories on YouTube are fixed and it is hard to add/remove categories too often. As time goes by, some categories may become obsolete and some new topics may be missing from the categories.

These observations motivate us to explore a new way of video categorization for facilitating video search. In this work, we propose a novel commentary-based clustering technique by utilizing user comments for achieving this goal. Unlike the traditional approaches of predefining some categories by human, our categorization is learnt from user comments. The advantage of our proposed approach is three-fold. First, our approach can capture public attentions more accurately and fairly than that of the predefined categories approach as we have taken the user opinions into consideration. In other words, the resulting categories are contributed by public users rather than a small group of people; Second, since user attentions can be changed from time to time, the categories of our method can be changed dynamically according to the recent comments by users; Finally, as users comments are in the form of natural language, users can describe their opinions in details with rich text. Therefore, by commentary-based clustering, we can obtain clusters which represent fine-grained level ideas of videos.

In the literature, various clustering techniques have been proposed for video categorization [16] [26]. However, this type of techniques did not take user opinion into consideration and thus the clustering results do not capture public interested issues.

Apart from predefined categories, YouTube also provides tagging to assist video searching service. Indexing videos with some words describing the videos should theoretically be helpful in the context of video understanding while users would not have any idea before viewing a video. Nevertheless, the tags are usually too loose and not structural which are hardly to give enough description of videos.

Some researchers have proposed to use the user tags on videos for clustering [17] [15]. Though user tags can somehow reflect user feelings on videos, tags are, in many cases, too brief to represent the complex ideas of users and thus the resulting clusters may only carry high-level concepts. Another stream of works which use commonly fetched objects of users for clustering [11] suffer similar shortcoming of neglecting object content. In [25], they proposed to adopt a multi-modal approach for video categorization. However, their work required lots of human efforts to first identified different categories from a large amount of videos.

We want to remark that although commentary-based clustering can theoretically obtain more fine-grained level clusters, it is much more technically challenging than that of tag-based clustering. The reason is that user comments are usually in the form of natural language and thus preprocessing is necessary for us to clean up the noisy data before using them for clustering.

The rest of the paper is organized as follows. Section 2 discusses previous works in the context of social network mining. Section 3 explains our proposed approach for video categorization in video sharing sites. Section 4 briefly introduces our web crawler. Section 5 presents the details of pre-processing of the raw data grabbed by our crawler. Section 6 describes our video clustering algorithm. Section 7 presents and discusses our experimental results. Section 8 concludes the paper.

2 Related Works

Since the late eighties, data mining has became a hot research field. Due to the advancing development of technologies, there is an increasing number of applications involving large amount of multimedia. For this reason, researches in the field of data mining are not limited to text mining but multimedia mining. Qsmar R. Zaine et al. [26] developed a multimedia data mining system prototype, MulitMediaMiner, for analyzing multimedia data. They proposed modules to classify and cluster images and videos based on the multimedia features, Internet domain of pares referencing the image of video, and HTML tags in the web pages. The multimedia features used include size of image or videos, width and height of frams, date on which the image or video was created, etc. S. Kotsiantis et al. [16] presented a work to discover relationships between multimedia objects based on the features of a multimedia document. In their work, features of videos such as color or grayscale histograms, pixel information, are used for mining the content of videos.

Motivated by the bloom of social networks, plenty of works have been done involving the study or analysis of online social networks. Different approaches are proposed to discover user interests and communities in social networks. Tag-based approach is one of the invented methods. In [17], Xin Li et al. developed a system to found common user interests, and clustered users and their saved URLs by different interest topics. They used the dataset from a URLs bookmarking and sharing site, del.icio.us. User interests discovery, and user and URLs clustering were done by using the tags users used to annotate the content of URLs. Another approach introduced to study user interests is user-centric which detects user interests based on the social connection among users. M. F. Schwartz et al. [20] discover people's interests and expertise by analyzing the social connections between people. A system, Vizster [14], was designed and developed to visualize online social networks. The job of clustering networks into communities was included in the system. For this task, Jeffrey and Danah identified group structures based on linkage. Except the use of sole tag-based or user-centric approaches, there are works done with a hybrid approach by combing the two methods. In [15], user interests in del.icio.us are modeled using the hybrid approach. Users are able to make friends with others to form social ties in the URLs sharing network. Julia Stoyanovich et al. examined user interests by utilizing both the social ties and tags users used to annotate content of URLs. Some researchers proposed the object-centric approach for social interests detection. In this approach, user interests are determined by the analysis of commonly fetched objects in social communities. Figuring out common interests is also a useful task in peer-to-peer networks since shareds interests facilitate the content locating of desire objects. Guo et al. [11] and K. Sripanidkulchai [22] presented in their works the algorithms of examining shared interests based on the common objects which users requested and fetched in peer-to-peer systems.

3 Public Attention Based Video Concept Discovery and Categorization for Video Searching

With the ceaseless growth of media content, it is increasingly a tense problem for video searching. It is usual that users hardly find their desire videos from the immense amount of videos. There are two main directions to ease the process of video searching, one is enhancing the text-based search engine whilst the other one is designing a better directory. In this paper, we focus on the former approach.

Though many video sharing sites allowed tagging function for users to use tags to annotate videos during the upload process, it is very common for user to tag videos by some high level wordings. As such, tags are usually too brief for other users to locate the videos by using the text-based search engine. In our method, as user comments usually describe the videos in details, we can use them for video clustering to obtain fine-grained categories. By identifying the concept words for each categories, we can use them as latent tags for the corresponding categories in order to facilitate the video searching process.

In music domain, music videos in sharing systems are always categorized according to their types of musical sounds (e.g. pop, metal, country, etc.) under the music genre. However, except music styles, people may have many different attitudes and preferences (e.g. appearance of singers, event of performance, age of songs, etc) towards music in different regions. Therefore, to categorize music based on publicly interested issues, music genre is not a good categorical construct for video searching.

Our aim is to find a categorization where videos in each video group are representing a popular topic of interest and improve index with the in-depth concept of videos. In our algorithm, public attentions are modeled and video concepts are discovered by clustering videos into groups with the utilization of user-left comments.

Previously, computer scientists have tried many ways to find user interests. Tags are very popular to help in this context [17]. However, in a previous study of tagging in Youtube, it has been observed that many tags could not enhance the description of video as a result of system constraints [10].

Several disadvantages would be raised in this manner. Tags on a video are manually given by the one who uploads the video, thus the tags are just expressing a single user's feeling about the video. A study of content interactions in Youtube shows that tagging is unreliable as a result of self-promotion, anti-social behavior as well as other forms of content pollution [8]. Therefore, tags on a video would have a strong bias and are not fair enough to exactly describe what the video is actually about. Furthermore, single-user given tags are definitely not representative of public feelings about the video. To address the sparsity and ambiguity of tagging, folksonomy search has been suggested [18] to improve existing tags in video. However, such systems still depends on a set of content category tag which is self found in youtube.com.

Moreover, videos are often tagged with a small number of words. As such, often fails to give enough description on the video. Though there is a previous work classifying videos from youtube.com by using the tags, the reported average number of tags per video is just 8 to 9 which is far fewer than the amount of comments per video [21]. Therefore, tags are insufficient to provide detailed information about videos. Another study of tagging across four major social media websites has shown that only 0.58% of tags in youtube.com belongs to the content category. Such percentage is the lowest among the four major social media websites of study [13]. In order words, only a very small amount of tag can identify the content category of the video in youtube.com. Since comments can be given by any users on any videos as feedbacks, they express different users thoughts about a video. Thus, containing more in-depth information about the videos. Also, by allowing every user to leave feedbacks, the number of comments on a video are usually much more than that of tags. Hence, utilizing comments instead of tags to find out the attracted issues can solve the above difficulties.

In a study of video search in youtube.com, it is found that search services are critical to social video websites but users often cannot contribute to the search service [12]. In our proposed work, such problem can be addressed by involving the user-left comments to enhance video searching.

Mentioned above, though tagging is popular be used as an assistant in video sharing sites, it is yet far from perfect for video searching. Our proposed work is aimed to supplement the tagging technique to achieve the goal of providing a better video searching service for users.

Beside tag-based, some researchers proposed the content-based approach to categorize videos [23]. Using video content as categorizing materials can group similar videos together according to their actual content. Nevertheless, video content itself only provide objective information about the videos but nothing about users' idea. Consequently, this approach fails to group videos according to public attentions. In contrast, user-left comments include users' view about the videos. Therefore, comments can, undoubtedly, be used to categorize videos based on public attentions.

Video features can also be used to achieve the goal of videos clustering [16]. Video features, however, are hard to be extracted automatically. Due to the limitation of human resources, automatic information retrieval from mass amount of data is preferred. Also, using video features to cluster videos suffers the same shortcomings of content-based as well. Because of information retrieval dealing with text is much easier than video features extraction, and comments, in addition to video content, provide users' views on videos, user-left comments are significant for clustering videos.

4 Dataset collection

YouTube is a video sharing platform on which users can upload their own videos for sharing purpose. Along with each video, a short description can be entered by the uploading user and tags as well. Apart from the video uploading user, other registered users can also contribute to the video surrounding text by leaving comments on the video. In this paper, we focused on the user comments of videos of Hong Kong singers in YouTube and did a comparison between comments and tags.

We first defined a set of 102 Hong Kong singer/group names. Given the set of singer/group names, we developed a crawler to firstly visit the YouTube web site and automatically searches from the site the related videos based on video titles and video descriptions. From the resulting videos, the crawler saves the URL of each videos for further process. For the convenience of gathering user comments, the crawler transforms the fetched URLs to links which link to the pages of "all comments" mode of corresponding videos. With all the transformed video URLs, for each link, the crawler is able to scrape the video web page and grab the video title, all the user comments and the user names of who left comments on the video.

In the data set acquired by our crawler, 19305 videos are grabbed with 102 singers and 7271 users involved.

5 Data Pre-processing

To ease the process of video searching by discovering the public attentions and categorizing videos, larger amount of data is required from video sharing sites. However, just the large-sized collection of text-formatted raw data is not applicable for further processing. Large-sized dataset always need to undergo data pre-processing in the field of data mining. Here is no exception in our algorithm. After crawling YouTube, the mass data need to be pre-processed before performing video clustering.

Here are two steps of data pre-processing involved in our introduced algorithm,

1) Data Cleaning

2) Text Matrix Generation

5.1 Data Cleaning

As the comments left on YouTube videos are written in natural languages which consist lots of non-informative words, such as "thank" , "you", etc, text processing with such materials must be caution. To avoid resulting a poor clustering, data cleaning is necessary for handling the noisy words.

In natural languages, there are many words that are not informative for clustering. These words would make the entire dataset very noisy. Applying a stoplist is one of the ways to clean up these words. Since some words are obviously not informative, it is easy to define a stoplist of noise. With a predefined stoplist, non-informative or distractive words can be strained from the dataset. After removing all the useless words by the stoplist, the dataset is then passed to the process of matrix generation.

5.2 Text Matrix Generation

Text-formatted data is not easy for further processing, it is more convenient to transform the data from text to matrix representation beforehand.

For example, the dataset can be represented by matrix A of size $n \times m$ where n is number of videos in the dataset and m equals to number of unique case-insensitive words in the dataset. In A, each row is a vector of video words and element $a_{i,j}$ is the frequency count of word j occurs in comments left on video i.

To transform the textual data into a more easy-computed text matrix, a dictionary is firstly built with the case-insensitive words in all the comments in the dataset. As comments are all in texts, linguistically, there exist many meaningless words in comments. These meaningless words, e.g. "is", "am", "the", "a", always occur in an extremely high frequency. Therefore, words occur in frequency exceeding a threshold should be discarded. On the other hand, words that seldom occur are probably not the important ones, so words with few occurrence should also be neglected. Therefore, we set an upper bound and a lower bound for word occurring frequency. All the words with frequency less than the lower bound or larger than the upper bound are filtered out. After filtering all the meaningless words, dictionary can then be built and matrix can be generated as well.

6 Video Processing via Clustering

In order to facilitate the video searching process, finding fine-grained video concepts and constructing a video category based on public attentions are crucial as there is no way to match a video with the desired ones without a deep understand of video content and people do searching with their interests in the usual practice.

As video comments left by users provide opinions about the video or singers in the video, some words in the comments are actually describing the fine-grained level concept of videos. Therefore we can find video concepts analyzing the video comments. With the concept words discovered from comments, video indexing can be improved by incorporating those concept words. Hence, facilitating video searching and make it be done in a more accurate manner.

With the reason that public attentions are reflected from the comments users left on videos, grouping similarly commented videos together is a possible way to provide a good video categorization. Since the objective of clustering is to distinguish substantial amount of data and group similar objects together, clustering is an adequate algorithm for constructing a video category that can guide user to his/her desire videos.

Figure 1 shows the procedures of finding video concepts, discovering public attentions to Hong Kong singers and categorizing Hong Kong singer videos from YouTube.

6.1 Video Clustering and Concept Discovery

For our purpose of building a good video category and learn the video concept for easier video searching, Non-negative Matrix Factorization (NMF) is the chosen clustering algorithm [24]. We propose to apply NMF for clustering based on three reasons. First of all, NMF is a bi-clustering method. With a bi-clustering algorithm, comment words and videos can be clustered simultaneously. Thus, the main characteristics of video groups can be drawn while grouping videos with similar user views together. Additionally, NMF does not provide an absolute assignment of videos to groups. Absolute assignment clustering algorithms are not suitable for singer video clustering. In practice, a video can belongs to multiple groups. For example, a classic music video can be performed by a singer who is passed away. The video is said to be in both ”classic” group and ”died singer” group. As NMF calculates possibility coefficients of each video to different groups, a single video can be assigned videos to multiple groups. Finally, NMF is effective for clustering. Since we need to cluster a large amount of data, effectiveness is one of the concerns. An effective low-dimensional linear factor model is desired.

Comments on a video often capture users feelings about the video or describe the video. Videos are clustered into the same group if they bear comments with similar contents. Similar videos, therefore, can be grouped together and with their characteristics be revealed as publicly attracted ones.

Let A be the $n \times m$ video-word matrix generated in the process of data pre-processing, where n and m are the number videos and number of words in dictionary respectively. As all the elements in A are the occurrence counts of words in documents, they are greater or equal to zero. This makes matrix A a non-negative matrix.

Since the importance of a term to a document can be reflected by it’s number of appearance, the well-known keyword measure in Information Retrieval $tf - idf$ is adopted for extracting important words. Within the dataset, all the comments of a video is aggregated and considered as a document. Importance of term i in document j is $w_{i,j}$ which is computed by using $tf_{i,j}$ (term frequency of term i in document j) and idf_i (inverse document frequency of term i). Terms that are important to a document are expected to appear many times in the document. For this reason, the term frequency is used to measure the normalized frequency of a term in a document. Suppose there are t distinct terms in document j, $tf_{i,j}$ can be computed as,

$$tf_{i,j} = \frac{f_{i,j}}{\sqrt{\sum_{k=1}^{t} f_{k,j}^2}} \tag{1}$$

where $f_{i,j}$ is the number of times that term i appears in document j. As words appear in many documents are not useful for distinguishing documents, a measure idf is used to scale down the importance of these widely-used terms. The inverse document frequency of term i is defined as,

$$idf_i = log\frac{N}{n_i} \tag{2}$$

where N is the total number documents in the dataset, and n_i is number of documents that containing term i.

After computing the term frequency and inverse document frequency, the importance weight of a term i in document j is defined as the combination of $tf_{i,j}$ and idf_i,

$$w_{i,j} = tf_{i,j} \times idf_i \tag{3}$$

The greater the weighting, the more the importance is the term to the respecting document.

From matrix A, a non-negative matrix X can be produced by calculating the importance weights. Each element in X is defined as,

$$x_{j,i} = w_{i,j} = \frac{a_{i,j}}{\sqrt{\sum_{k=1}^{t} a_{k,j}^2}} \times log\frac{N}{n_i} \tag{4}$$

By fitting a k-factor model to matrix X, where k equals to number of groups to be obtained, X is decomposed into two non-negative matrices W and H, such that $X = WH + U$. After matrix decomposition, W is in size of $n \times k$ and H is in size of $k \times m$.

Our objective is to find W and H such that $X \approx WH$. By iteratively updating W and H, we can obtain W and H by minimizing the following function,

$$F(W, H) = ||X - WH||^2 \tag{5}$$

with respect to W and H and subject to constraints that $W, H \geq 0$.

Figure 2 shows the decomposition of video dataset matrix. From the resulting matrices, relationships between

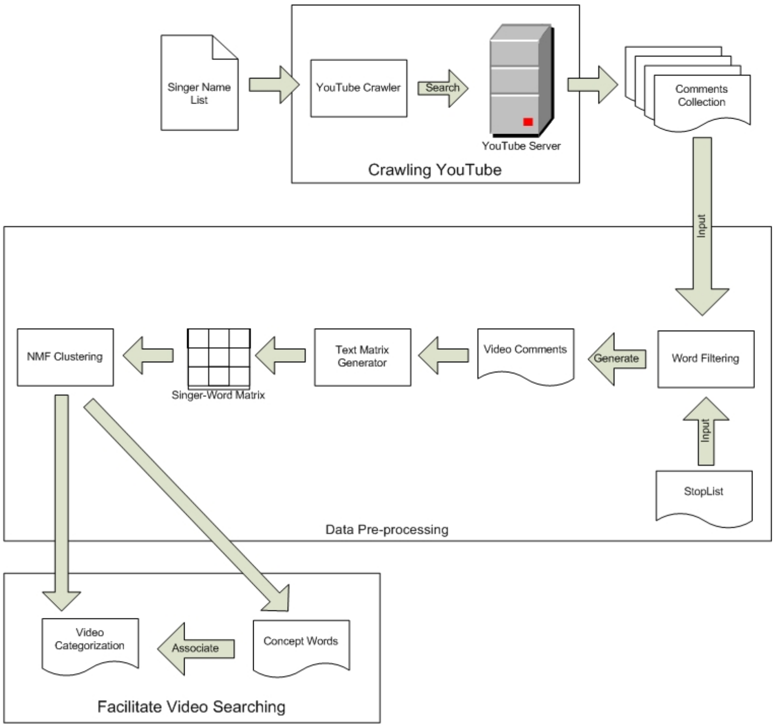


Figure 1. Video concept discovery and video categorization of Hong Kong singer videos in YouTube .

words, videos and clusters are revealed. Matrix W shows the relationships between videos and different clusters, whilst H clarifies the relationships between words and clusters. In W, value held in $w_{n,k}$ is the coefficient indicated how likely video n belongs to cluster k. To fit the purpose of our research, we have refined the method of group assigning in NMF. The original application of NMF algorithm assigns an object to a group in a maximum coefficient approach. However, in our method, video n is treated to be in group k if $w_{n,k}$ has the a value greater than a threshold β_k within vector n in W, where the value of threshold β_k is data dependent. The threshold should be chosen in a coefficient distribution depending manner. Videos can then be grouped into clusters based on their similarities. We define the set of clusters for video V_n that it belongs to as,

$$C_n = \{k \in K \mid \forall\, W_{n,k} > \beta_k\} \tag{6}$$

where K is set of all clusters.

Matrix H provides the information about the characteristics of the video groups. Concept words of a cluster can be found with H as $h_{k,m}$ is the coefficient of the term m belongs to cluster k. For each cluster, the top 10 words, with respect to the term-cluster coefficient, are considered to be the concept words for the cluster. Which the words states the properties of a group of videos and gives an in-depth description for the videos. Enhancing video index by incorporating the discovered concept words can consequently improve users video searching experience.

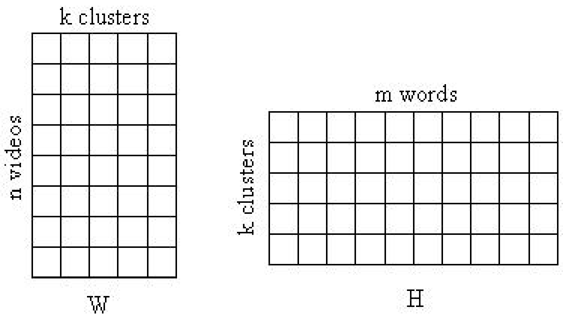


Figure 2. NMF decomposition for video clustering.

6.2 Factorized Component Entropy Measures for Vocabulary Construction

While matrix factorization methods and latent Dirichlet methods have often been successful applied to process news articles and technical papers, applications of such algo-

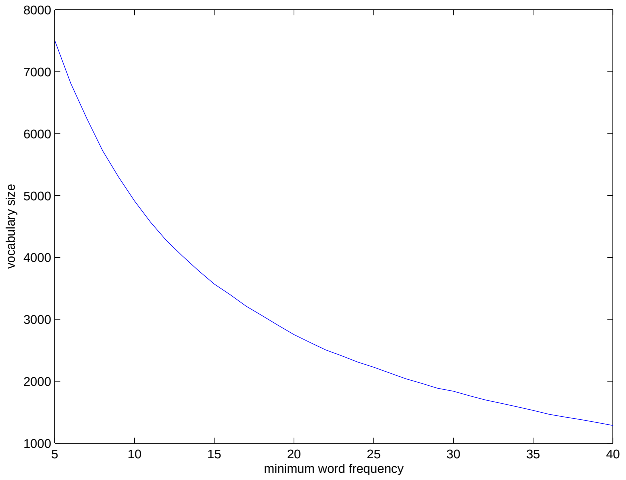


Figure 3. Vocabulary size.

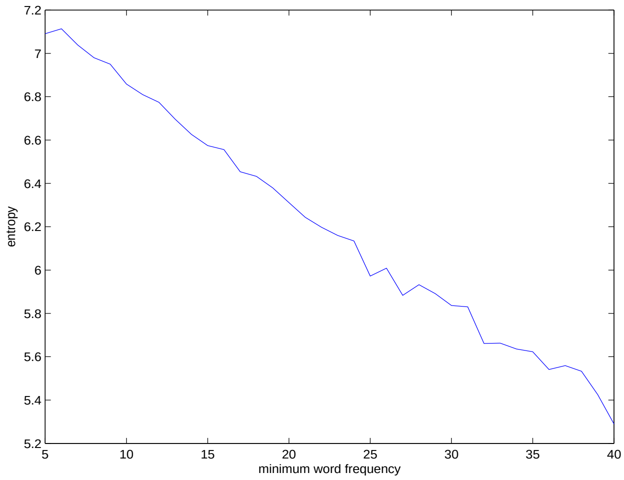


Figure 4. Word-concept entropy.

rithms to short and terse statements in commentary pose significant difficulties. Misspellings and the very short length of the commentary are often the norms in comments in youtube.com. We propose the use of factorized component entropy as a measure to construct good vocabulary for analyzing noisy commentary.

Figure 3 shows size of the vocabulary as a function of the global minimum word frequency where we can see a sharp drop in the size of vocabulary when the global word frequency is increased.

The two matrices W and H generated from factorization have the effect of indicating the cluster membership. The cluster membership c_i of the $i-$th concept is simply given by

$$c_i = \arg\max_j W_{ij},$$

where j is the concept label. To evaluate how the words are distributed among the different concepts, we can compute the word-concept entropy of the j-th concept using the following formula,

$$Ef_j = -\sum_i (H_{ij}/\sum_i H_{ij}) \log(H_{ij}/\sum_i H_{ij}). \quad (7)$$

A smaller word-concept entropy implies that the words in the features have coefficients in H that is distributed across a smaller number of features and is thus more favorable. A large concept entropy implies that the words have coefficients evenly distributed across the different concepts and thus cannot be clearly differentiated.

Figure 4 shows the word-concept entropy as a function of the global word frequency. As the global word frequency increase, and the size of vocabulary decreases which leads to a reduction in word-concept entropy.

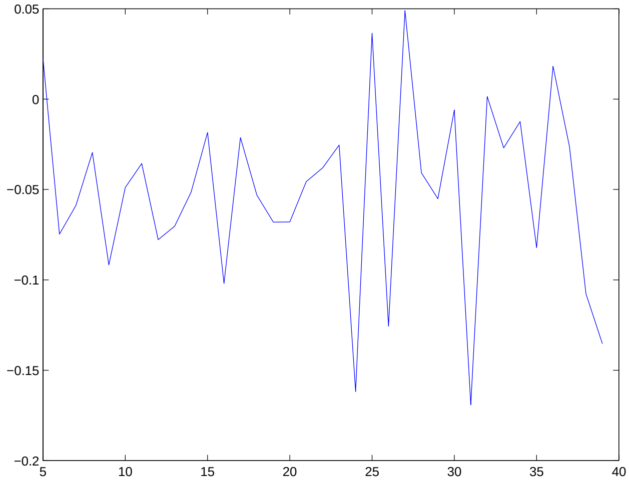


Figure 5. Derivative of word-concept entropy.

By taking the discrete derivative of the entropy, we can measure the change in entropy where the large drop in entropy represents the suitable size for vocabulary construction. Figure 5 shows the derivative of the word-concept entropy.

Similarly, we can also define the video-concept entropy which represents how well video commentary are grouped together using the following video-concept entropy formula,

$$Es_i = -\sum_j (W_{ij}/\sum_j W_{ij}) \log(W_{ij}/\sum_j W_{ij}), \quad (8)$$

where Es_i is the video-concept entropy of the j-th video commentary.

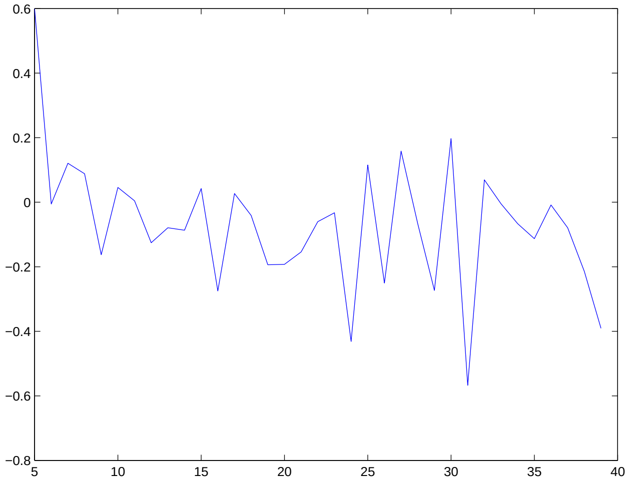


Figure 6. Derivative of joint entropy.

In the end, the joint entropy can be obtained by multiplying the video-concept entropy, word-concept entropy and the entropies similarly obtained by taking the transpose of W and H. The derivative of the joint entropy is shown in Figure 6, where the noises are further suppressed.

7 Experimental Evaluation

A proof-of-concept experiment was done to with videos in Hong Kong regional music domain. An Intel(R) Core(TM)2 Quad 2.40GHz PC with 4GB RAM was used to conduct our experiment. Our web crawler was implemented in VC++ and the core algorithm was implemented in Matlab.

7.1 Empirical Setting

As the videos were grabbed by searching from the YouTube site with predefined list of singer names, there are possibilities that some videos are grabbed more than one time. For those videos preformed by more than one singer, as long as there are more than one singer names annotated in the video title, the video will be collected in times equals to the number of hits the predefined singer name hits the video title. To achieve a more accurate clustering result, duplicated videos are removed from the dataset.

In comments, users are used to mention the singer names when they are commenting on him/her. This will make the singer names dominate in every group of concept words. However, it is not conspicuous enough to reveal detailed concept of videos by singer names. Therefore, in our experiment, we add singer names to the stoplist as well.

Furthermore, some videos are less popular or just been uploaded for a short time that only have a few comments. These videos which have relatively few words are non-informative for video clustering. Videos with commentary words less than the threshold discovered in earlier section are removed.

The videos are clustered into k groups with the clustering algorithm discussed in section 7, where k is experimentally set as 20. The experiment was done twice, once with threshold β_i regarding cluster i to be mean coefficient of all videos,

$$\beta_i = meanCoef_i = \frac{\sum_{j=1}^{n} w_{j,i}}{n} \tag{9}$$

To compensate the poor performance caused by the extremely uneven distribution of coefficient, we chose the threshold to be mean coefficient plus standard deviation of all videos for the second experiment. β_i regarding cluster i is defined as,

$$\begin{aligned} \beta_i &= meanSdCoef_i \\ &= \frac{\sum_{j=1}^{n} w_{j,i}}{n} + \sqrt{\frac{1}{n}\sum_{j=1}^{n}(w_{j,i} - \frac{\sum_{j=1}^{n} w_{j,i}}{n})^2} \end{aligned} \tag{10}$$

where n is total number of videos being clustered.

7.2 Video Categories and Concepts

Since video clustering is a complete clustering analysis, publicly attracted music categories in Hong Kong can be found by clustering the videos. We deployed NMF as our clustering method. Applied the clustering algorithm to the video dataset in the way discussed in Section 6.1, with the experimentally chosen number of cluster of 20, videos were clustered into groups based on the words in their comments. The mean coefficient of videos to a cluster is set as the threshold. Videos with coefficient higher than the threshold of a cluster are said to be in that cluster. Under this strategy, videos can belong to several clusters as they may have multiple characteristics. Table 1 shows the discovered categories and concepts from our dataset.

Unlike the generic music video categorization of some famous video sharing sites, such as DailyMotion divides music videos into eight classes (Pop, Rock, Rap, R&B, Jazz, Metal, Covers, and Electros), we categorized videos of local singers into twenty classes which are far more specific.

From our clustering result, we noticed that videos of singers are not only limited to general music videos, but also funny clips, award presentations, commercial advertisements as well as event promotion clips. Looking at the music videos alone, by clustering users' comments, we

Group	Concept Words
1	beautiful lyrics melody
2	female makeup dress
3	cute pretty handsome
4	sex photos scandal
5	funny hilarious laughing
6	rap raps hip
7	movie film story
8	cantonese mandarin language
9	commercial pepsi coke
10	piano piece ear grade
11	japanese japan korean
12	china olympic games
13	old classic memories
14	dance dancer moves
15	guitar band rock
16	award tvb gold
17	english chinese accent
18	sad legend died
19	together couple two
20	voice pretty talent

Table 1. Latent video categories discovered in Hong Kong music video domain from YouTube

Group	Top three cluster representative words
1	sin story her
2	ltd invisible target
3	木紋 如沾 數著
4	special 鐘泊桐 characters
5	我的第 hkpca awards
6	andrew yun fung
7	family food sheh
8	actors chin stephen
9	quot buenos zero
10	惺士山工 小南版 repeat
11	始終有 這地球 寺唱
12	bigboy2000 blogspot search
13	chi stephen derek
14	莫文蔚來囉 lollipop terry
15	label gold koon
16	takes goes 戀情告急
17	xuite daily blog
18	bird carina kar
19	lap jennifer wealthy
20	mahjong tak spirit

Table 2. Cluster representative words extracted from video meta data

found that people's attitude towards Hong Kong music are not only target on the music styles. There are also other features of music which people are interested in, like languages, age of music, music instruments, type of singers, singer's voice, composition goodness, etc.

Furthermore, categorize singer videos with the proposed clustering algorithm, people can identify dance-oriented videos (Group 14), cross-culture produced music (Group 11) or even movie theme songs (Group 7) easily. Other than simply categorizing singer video clips, some up-to-date news in the local music circle, like scandals (Group 4), can also be found.

Tags are popularly investigated in the contest of topic detection. To compare the effect of concept finding by comment with tags, an experiment was conducted with the same setting but video meta data as the dataset. Video meta data are all the video surrounding tests including title, tags and description. Contrastingly, representative words of clusters extracted from video meta data cannot bring any idea of the video groups. Table 2 lists the top three representative words of each resulting video group derived from meta data. From the words listed in the table, nothing about the video

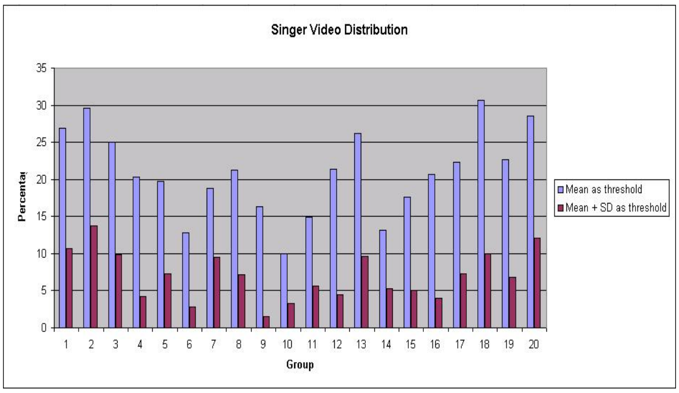


Figure 7. Singer video distribution in YouTube.

		Concept Words from Group			
		A	B	C	D
Percentage of Videos from Group	A	34.04%	4.02%	0%	2.13%
	B	0%	15.79%	0%	0%
	C	0.84%	0.84%	7.58%	0.84%
	D	0%	10.17%	3.39%	5.26%

Table 5. Percentage of videos with tags covering concept words across groups

		Precision	
Group	Concept Words	Mean as Threshold	Mean + SD as Threshold
A	sex photos scandal	21.64%	81.58%
B	old classic memories	61.04%	78.16%
C	sad legend died	35.86%	60.34%
D	together couple two	64.44%	79.82%
Average		45.75%	74.96%

Table 3. Precision of objective clusters

concept of the groups can be told. With this table and table 1, we can easily compare the effectiveness of comment and meta data for concept finding. The tables show that tags, titles or short descriptions are not sufficient for concept discovery of online videos.

Figure 7 illustrates the distribution of Hong Kong singer videos in YouTube according to the proposed algorithm using mean and mean + sd as thresholds. From the figure, we can see that the distribution of videos diverse over different threshold values. With the mean coefficient as the threshold, compared to the video groups resulted from the algorithm with mean + SD coefficient as threshold, larger groups of videos can be obtained. In the other words, algorithm associated with a smaller group assigning threshold would result heavier overlapped video groups.

The video clustering results are evaluated by human experts. To make the evaluation less controvertible, we only show the precisions of objective video groups in Table 3 where groups A, B, C, D are cluster 4, 13, 18, 19 respectively in our clustering. In the table, we noticed that assigning videos to groups with a smaller threshold may sometimes lower the precision. This will be caused in the groups which are very distinct to others. As a video group is too specific, the video-group coefficients to the group hold the extreme values. Also, closely related videos to the distinct group is always much fewer than videos which do not. Hence, videos are condense at the lower extreme side regarding the coefficients distribution. As a result, lowered the mean coefficient and caused the poor precision. On the other hand, we can see that the algorithm which assigns videos into groups with a larger threshold yields far better precisions. The average precision of the larger-threshold clustering groups in the table is 74.96% whilst that of the lower-threshold clustering is just 45.75%. The difference between the precisions resulted from clustering with the two different thresholds reflects the degree of extraordinary of the video group. The larger the difference, the more the special the group is. For example, in group 4, the two precisions differ from each other by a large percentage at about 60%, and from the concept words we can know that this group is about scandal of singers involving their sex photos. This is obviously an extremely distinct group.

7.3 User Comments vs User Tags

As tags are believed to be an accurate description of an object and have been widely used for finding user interests and grouping objects, it is necessary to examine the virtues of user comments over tags before utilizing comments to capture public attentions and categorize videos to facilitate the video search in video sharing sites. One important ob-

Cluster I	Top 10 concept words in user comments	old classic memories drama childhood love 80s memory loved san
	Top 10 frequent user tags	chinese chan mv cheung wong love music mtv top anita
Cluster II	Top 10 concept words in user comments	sad legend two died missed heaven star superstar crying talented
	Top 10 frequent user tags	cheung chan leslie anita mui chinese mv danny hong wong
Cluster III	Top 10 concept words in user comments	guitar solo band rock cover drummer chords intro crap violin
	Top 10 frequent user tags	chinese beyond wong kong cheung ka kui hong nicholas paul
Cluster IV	Top 10 concept words in user comments	sex photos stupid fake victims private innocent scandal girls stop
	Top 10 frequent user tags	gillian chung sex photo edison chen gill cheung cecilia chan

Table 4. Examples of concept words from user comments and user tags in four video clusters

servation from our experimental results is that user comments usually contains more in-depth information than that of user tags. Table 4 shows both the top 10 concept words found from user comments and the top 10 user tags of four clustered groups. From the concept words in the user comments, we can make a reasonable prediction that cluster I is about some music videos of some old songs. From the user tags, however, we can only find some singer names or some high-level descriptions (e.g. music, mv, mtv). Same as cluster II, from the concept words, this cluster is probably talking about some superstars who are already died. Nevertheless, the most frequent tags are only names of those dead superstars which do not reveal the low-level description of the group. Cluster III is the similar case as the above two clusters. Concept words from user comments state that this group is about the band sound and rock music but the tags only list out the name of a local popular band, ”Beyond”, and some of the band members. Tags of the other clusters suffer the similar problem as the above mentioned clusters. From the table, we can see that the user tags actually agree with our discovered concept words though the tags just exhibit the high-level sketch of the groups. In the other words, our algorithm gives an in-depth characterization of the videos with the concept words which the characterization cannot be exposed by the user tags, and in the mean time, the concept words achieve a strong agreement with the tags.

From this observation, we can conclude that if we want to obtain clustering results in a more fine-grained level, using commentary-based clustering technique is more suitable. For the purpose of facilitating video search, it is beyond doubt that result of fine-grained level clustering involving user points of attention is more desirable.

To give a more in-depth analysis of comments and tags, we have compared concept words against tags in different clusters. Table 5 records the portion of videos whose tags cover the concept words of different groups and there are two major observations from the table. First, we can see that there are at least 65% of videos whose tags cannot cover the concept words of the group they belongs to. This implies tag-based clustering cannot completely capture user opinions and video content. Second, we can see that the concept words of each group are mostly covered by tags of its own group. This once again verify the accuracy of our proposed method.

8 Conclusion and Future Work

In this paper, we have proposed a novel commentary-based matrix factorization technique to cluster videos to facilitate searching and generate concept words to improve indexing. We propose the use of factorized component entropy as a measure to construct good vocabulary for analyzing sparse and noisy social media data. Experimental results showed that our commentary-based clustering yields better performance than that of tag-based approach which was proposed previously in the literature. On the other hand, we have successfully discovered some non-trivial categories among the videos of Hong Kong singers. Since our categorization is learnt from user feedbacks, it can provide an easy way for users to reach their desired videos via our list of categories.

In our future work, we plan to extend the commentary-based technique from video clustering to user and singer clustering. After we have obtained the three types of clusters, we can acquire the relationships among different videos, singers and users by analyzing the inter-cluster similarity. As such, social culture can be studied by combining and analyzing the discovered relationships. With the video-video, singer-singer, user-singer, and user-user relationships found by clustering, we can know the changes in music styles and singer styles over the ages, the trend of music, the ways people appreciate music, and even the special relationships of singers reflected by news, and more. Relationships observed by clustering are not only useful for social scientists to study social culture, but also beneficial for businesses, entertainment companies, fans clubs, social network systems and system users. With the help of examined user-user relationships, businesses can be profited from reducing advertising costs by advertise only to the potential customer groups. User-signer relationships define user-idol groups, entertainment companies can effectively promote to the target groups. Determining the user-singer relationships, in addition to profits for entertainment companies, fans groups can easily be managed. Other than the advantages for some specific parties, general users are also benefited. Well-clustered groups of videos and singers equipped with a batch of concept words leads to a effort saving video searching for users. Also, social network systems are able to detect and refine incorrect tags with the concept words resulted from clustering. As a result, description of videos are more precise and thus improves the video searching function.

References

[1] http://www.bebo.com.

[2] http://www.dailymotion.com.

[3] http://www.facebook.com.

[4] http://www.flickr.com.

[5] http://www.myspace.com.

[6] http://www.youtube.com.

[7] Usa today. youtube serves up 100 million videos a day online.

[8] F. Benevenuto, F. Duarte, T. Rodrigues, V. A. Almeida, J. M. Almeida, and K. W. Ross. Understanding video interactions in youtube. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 761–764, New York, NY, USA, 2008. ACM.

[9] X. Cheng, C. Dale, and J. Liu. Understanding the characteristics of internet short video sharing: Youtube as a case study. *In CoRR abs*, Jul 2007.

[10] G. Geisler and S. Burns. Tagging video: conventions and strategies of the youtube community. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 480–480, New York, NY, USA, 2007. ACM.

[11] L. Guo, S. Jiang, L. Xiao, and X. Zhang. Fast and low-cost search schemes by exploiting localities in p2p networks. *J. Parallel Distrib. Comput.*, 65(6):729–742, 2005.

[12] M. J. Halvey and M. T. Keane. Exploring social dynamics in online media sharing. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1273–1274, New York, NY, USA, 2007. ACM.

[13] M. Heckner, T. Neubauer, and C. Wolff. Tree, funny, to read, google: what are tags supposed to achieve? a comparative analysis of user keywords for different digital resource types. In *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*, pages 3–10, New York, NY, USA, 2008. ACM.

[14] J. Heer and D. Boyd. Vizster: Visualizing online social networks. *IEEE Symposium on Information Visualization, 2005*, 2005.

[15] C. M. C. Y. Julia Stoyanovich, Sihem Amer-Yahia. Leveraging tagging to model user interests in del.icio.us. In *AAAI '08: Proceedings of the 2008 AAAI Social Information Spring Symposium*. AAAI, 2008.

[16] P. P. Kotsiantis S., Kanellopoulos D. Multimedia mining. In *WSEAS Transactions on Systems, Issue 10, Volume 3*, pages 3263–3268, December 2004.

[17] X. Li, L. Guo, and Y. E. Zhao. Tag-based social interest discovery. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2008. ACM.

[18] J. Z. Pan, S. Taylor, and E. Thomas. Reducing ambiguity in tagging systems with folksonomy search expansion. In *ESWC 2009 Heraklion: Proceedings of the 6th European Semantic Web Conference on The Semantic Web*, pages 669–683, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] C. G. R. A. A. F. L. Rodrygo L. T. Santos, Bruno P. S. Rocha. Characterizing the youtube video-sharing community. 2007.

[20] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Commun. ACM*, 36(8):78–89, 1993.

[21] A. S. Sharma and M. Elidrisi. Classification of multimedia content (video's on youtube) using tags and focal points. Working paper.

[22] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, volume 3, pages 2166–2176 vol.3, 2003.

[23] S. Tsekeridou and I. Pitas. Content-based video parsing and indexing based on audio-visual interaction, 2001.

[24] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, New York, NY, USA, 2003. ACM.

[25] L. Yang, J. Liu, X. Yang, and X.-S. Hua. Multimodality web video categorization. In *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 265–274, New York, NY, USA, 2007. ACM.

[26] O. R. Zaïane, J. Han, Z.-N. Li, S. H. Chee, and J. Y. Chiang. Multimediaminer: a system prototype for multimedia data mining. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 581–583, New York, NY, USA, 1998. ACM.

Measurements, Analysis and Modeling of Private Trackers

Xiaowei Chen
Computer Science Department
Hong Kong Baptist University
xwchen@comp.hkbu.edu.hk

Xiaowen Chu
Computer Science Department
Hong Kong Baptist University
chxw@comp.hkbu.edu.hk

Abstract—**BitTorrent plays a very important role in the current Internet content distribution. The enormous impact of public and private trackers should not be overlooked. Public trackers are suffering from free-riding problem, but private trackers work very well because of effective Share Ratio Enforcement (SRE) which is an auxiliary incentive mechanism. Therefore, understanding the characteristics of private trackers is essential to BitTorrent content distribution and further development.**

We have crawled and traced fifteen trackers with nearly one million torrents for four months. In this paper, we first provide taxonomy of private trackers, and then present in breadth and depth measurement study on the characteristics of private trackers. We have found that private trackers have apparently different features and statistics from public trackers, ranging from the user viscosity, torrents evolution, user behaviors, content distribution and other metrics to measure the quality of service in private trackers. There are some new features that have not been examined in private trackers by previous measurement studies. Furthermore, we use game theory to study the effectiveness of SRE mechanism. We model the mechanism and propose an improved SRE mechanism to further incent the users and enhance the performance of private trackers.

Keywords-private tracker, incentive mechanism, BitTorrent, content distribution, peer-to-peer networks

I. Introduction

BitTorrent is currently the dominating P2P file sharing protocol. As one of the core components in BitTorrent protocol, trackers play an irreplaceable role during the distribution process which periodically provides updated peer lists to connected clients. BitTorrent trackers can be divided into two categories, public trackers and private trackers [1].

Public trackers (also known as open trackers) can be used by anyone by adding some tracker addresses to an existing torrent. There are a lot of public trackers such as the Pirate Bay [2], Mininova [3], ISOHunt [4], etc. Although BitTorrent has implemented the Tit-for-Tat (TFT) algorithm as an incentive mechanism, public trackers are still suffering from free-riding problem: first of all, a peer may stop uploading immediately after it finishes the download task; secondly, a peer usually sets a limit on the total upload bandwidth.

In recent years, private trackers become more and more popular and generate a huge amount of Internet traffic [5]. E.g., Torrents.ru alone has more than 1000TB of content and generates more than 46GB/s of Internet traffic. Users in private trackers can usually achieve much faster download speed than users in public trackers can. Based on strict member controlling policy, private trackers adopt Share Ratio Enforcement (SRE) as an auxiliary incentive mechanism to overcome the free-riding issue. SRE forces registered users to maintain a ***share ratio*** (i.e., upload-to-download ratio). In general, a registered user will be banned from private tracker community if his share ratio is lower than a threshold (e.g., 0.5). On the other hand, a user with higher share ratio can receive more benefits. There are two ways to achieve a high share ratio: one is to provide a high upload bandwidth; another one is to prolong the seeding time.

The main differences between public trackers and private trackers are summarized as follows:

- Users & Torrents Scale: Public trackers are open to everyone, but private trackers are available for registered users only. It is remarkable that the number of torrents in private trackers collectively will be much more than that in public trackers in existence [6].
- Incentive Mechanism: Public trackers rely on the BitTorrent TFT algorithm. Private trackers implement an additional SRE mechanism to incent the users to contribute as much as possible.
- Traffic Counting: Public trackers do not count each user's traffic during content distribution. Private trackers accurately record each user's total amount of uploaded data (T_u) and downloaded data (T_d). A user's ***Share Ratio*** is then calculated as T_u/T_d. If a user's Share Ratio is lower than a predefined threshold, he will be banned from this private tracker community.
- Download Performance: The download performance of private trackers are usually much better than public trackers due to the SRE mechanism. This is because a user in private tracker is incented to provide a high upload bandwidth and a long seeding time.

Given the importance and popularity of private trackers, it is essential to understand their characteristics so as to help us design better mechanism and build better sustainable environment. To this end, we conduct a measurement study and provide a thorough analysis on the collected datasets. Our main contributions are summarized as follows:

- We have crawled 13 private trackers and 2 public trackers from September 28, 2009 to February 10, 2009, and have obtained 31 datasets that cover nearly 2.5 million torrents.
- By analyzing the collected datasets, we find that private trackers have apparently different features and statistics

from public trackers, ranging from the user viscosity, torrents evolution, user behaviors, content distribution, et al. We show that SRE is an effective mechanism to incent users to seed as much as possible.

- Furthermore, we use game theory to study the effectiveness of SRE mechanism. We model the mechanism and propose an improved SRE mechanism to further incent the users and enhance the performance of private trackers.

The rest of our paper is organized as follows. Section II provides an overview of private trackers. Section III describes our measurement methodology and the collected datasets. Section IV presents detailed analysis on the measured datasets. The modeling of SRE mechanism based on game theory is presented in section V. Related work is presented in section VI, following by our conclusions in Section VII.

II. Overview of Private Trackers

In this section, we provide a taxonomy of private trackers, define certain terms which are widely used in this field, and briefly introduce the operation principle of private trackers.

A. Taxonomy Overview

We begin our taxonomy by classifying the roles and operation processes of private trackers. We break down the roles in private tracker into the two components shown in Table I. In general, private tracker's sysops establish and publish private tracker by using open source codebase, and then operate it by attracting users with high ranking [7]. Users sign up in private trackers by using some approaches [8]. After that, registered users select accepted clients to seed or leech contents above minimum share ratio, and then do contributions if possible (e.g. upload contents, donation, etc.). For each role and operation process we give related terms and definitions in part B.

TABLE I. Breakdown of Private Trackers

Private Trackers		
Role	***Operation Process***	***Related Terms***
Tracker User	Registration	Invitation Code
	Sharing (Seeding & Leeching)	Share Ratio, Passkey, Slot, Snatched, Freeleech, HnR, Point System
	Contribution	User Class
Tracker Owner	Building & Publishing	Codebase, Seedbox
	Ranking Improving	Scene Release, Pre/Pre'd, Pre-time

Related Terms do not cover all components to operation processes.

B. Terminologies and Definitions

The terms used in the private trackers are not standardized. For the sake of clarity, we define the terms shown in Table I, which are widely used in private trackers. Note that the terms list here are non-exhaustive.

Invitation Code: It is a typical registration way to become a member of closed private tracker communities. Contributing members meet specific requirements are eligible to invite their friends to join this private tracker by sending an invitation code.

Share Ratio: The private tracker calculates the share ratio for each user, which is the total amount of data the user has uploaded, divided by the total amount it has downloaded.

Passkey: It is a unique identity that private trackers assign each registered user. Passkey is usually a hexadecimal string. It is appended to the announce URL in the .torrent file which is dynamically generated for each member. This is to prevent private torrents from being uploaded to public websites. Well behaved users should not leak the passkey and announce URL to other members or public trackers; otherwise, they will be banned from the private trackers.

Slot: The slot system is used to limit the concurrent downloads for members that have ratio below a minimum value. If all the download slots are filled, the system will deny any connection before validating.

Snatched: It indicates that how many times a torrent file is completely downloaded.

Freeleech: When a torrent is flagged with freeleech, it means leeching that torrent will neither affect member's download amount nor decrease his share ratio. Meanwhile seeding that torrent will continue to increase upload amount.

HnR: It refers to the behavior that a peer stops uploading as soon as it completes the downloading. HnRs are highly prohibited, as they are counter-productive to sharing and the health of the torrent. Different trackers have different rules about HnR.

Point System: Many private trackers incorporate a "point/credit system" with SRE mechanism. Points can be earned maintaining a good share ratio; uploading torrents, etc. Points can be spent to improve member's account authority.

User Class: Most private trackers deploy a ranking system to categorize their users based on each user's contribution to the community. For example, the users in CHDBits are categorized as shown in Table II [9].

Codebase: Typically private trackers do not develop themselves, but instead use or modify many open source codebases, such as TBDev [10], TS SE [11] , etc.

Seedbox: A seedbox is a private dedicated server used explicitly for torrent transfers or seeding at high rate.

Scene Release: Basically it means that a DVD or software which has been cracked by someone or some groups.

Pre/Pre'd: When a release group pres a release (distribution content), it will be available for other people and the distribution will start.

Pre-time: Since proper Scene Releases are not directly pre'd on trackers, normally they are done on IRC channel, there exists a waiting period before it arrives the private trackers.

C. Operation Principle of Private Tracker

Private trackers implement a strict set of rules to control member eligibility and content quality. Users sign up to be a member of private tracker through an invitation system. Under a passkey system, each member is given a unique announce URL to perform content distribution. In a predefined User Class system which adopts SRE mechanism, members can be automatically or manually promoted or demoted. The typical operation principle of private tracker is illustrated in Fig. 1.

TABLE II. USER CLASS IN CHDBITS TRACKER

Different User Classes		
User Class	***Explanation***	***How to Work***
Peasant	Demoted users who must improve their ratio within 20 days or they will be banned.	Demoted to this class if: Downloaded>10GB & ratio<0.3 Downloaded>50GB & ratio<0.4 Downloaded>100GB & ratio<0.5 Downloaded>200GB & ratio<0.5 Downloaded>400GB & ratio<0.7
User	Default class of new members. Can upload subtitles and delete subtitles. Cannot view NFO file.	

Power User	Can view NFO file.
	Can view user list,
	Ask for reseed,
	Send invitation,
	View Top 10 and
	View other's torrents
		Be a member ⩾ 5 weeks;
		Downloaded ⩾ 50GB
		And ratio ⩾ 1.05;
		Auto demoted if ratio < 0.95.

Elite User	Elite User or above would never be deleted if parked.	Be a member ⩾ 10 weeks;
		Downloaded ⩾ 120GB
		And ratio⩾ 1.55;
		Auto demoted if ratio < 1.45.

Crazy User	Can view other users' history of comments and forum posts.
	Can be anonymous when seeding/leeching.
		Be a member ⩾ 15 weeks;
		Downloaded ⩾ 300GB
		And ratio⩾ 2.05;
		Auto demoted if ratio < 1.95.

Insane User	Can upload torrents without going through the Offer section.	Be a member ⩾ 20 weeks;
		Downloaded ⩾ 500GB
		And ratio⩾ 2.55;
		Auto demoted if ratio < 2.45.

Veteran User	Veteran User or above would never be deleted whether parked or not.	Be a member ⩾ 25 weeks;
		Downloaded ⩾ 750GB
		And ratio⩾ 3.05;
		Auto demoted if ratio < 2.95.

Extreme User	Can update outdated external information.	Be a member ⩾ 25 weeks;
		Downloaded ⩾ 1TB
		And ratio⩾ 3.55;
		Auto demoted if ratio < 3.45.

Ultimate User		Be a member ⩾ 30 weeks;
		Downloaded ⩾ 1.5TB
		And ratio⩾ 4.55;
		Auto demoted if ratio < 3.95.

Nexus Master		Be a member ⩾ 30 weeks;
		Downloaded ⩾ 3TB
		And ratio⩾ 4.55;
		Auto demoted if ratio < 4.45.

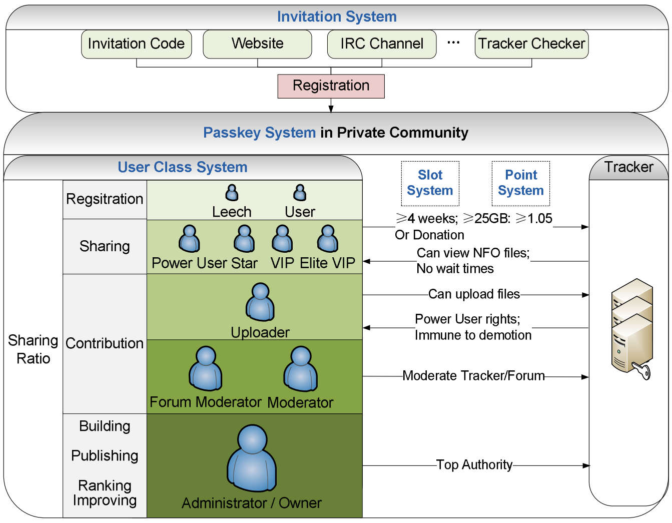


Figure 1. The Typical Operation Principle of Private Tracker

III. MEASUREMENT SETUP

In this section, we present the details of our measurement setup, methodology, and an overview of our collected datasets.

A. *Selection of Private Trackers*

There are a huge amount of private trackers on the Internet [12]. Most of them are closed systems, and it is difficult for an outsider to receive an invitation code. Nevertheless, based on the list and categories of [7, 12], we successfully joined 13 representative private trackers to perform our measurement study. For comparison purpose, we also crawled 2 well-known public trackers. We summarize all these trackers in Table III.

TABLE III. TRACKER LIST

Category		**Tracker Information**		
		Name	***# of Active Torrents***	***# of Registered Users***
Public	General	thePirateBay	19,931,758	4,196,330
		TorrentPortal	2,345,113	1,197,133
Private	General	TorrentLeech	24,347	N/A
		RevolutionTT	29,760	N/A
		Demonoid	236,983	N/A
		Bitsoup	13,474	N/A
		ILoveTorrents	9,313	N/A
	HD (High Definition)	CHDBits	12,093	18,939
		HDStar	6,805	8.397
		HD-Torrents	8,356	35,924
	Foreign	Torrents.ru	723,798	3,804,600
	Music	DimeaDozen	36,615	109,990
	TV	TheBox.bz	51,999	N/A
	DVD	AsianDVDClub	26,144	43,043
	Adult	PureTNA	66,883	1,306,501

Data updated until. 16:00, February 4, 2010. N/A indicates the tracker does not provide it.

B. *Data Crawling and Collection*

We use passive method to collect private tracker traces by using a crawler developed by ourselves. Data collection is recorded periodically by crawling web pages provided by each private tracker. These datasets are divided into three aspects, as shown in Table IV. "Tracker statistics" covers the information of registered users, torrents, seeders, leechers, and seeder-to-leecher ratio, etc. "Top 10" includes information about top members based on traffic, speed, share ratio, etc. "Torrent list" contains detailed torrent information, such as torrent type, name, size, added time, snatched times, seeders, leechers, etc. Notice that trace duration includes tracker maintenance periods during which our crawling cannot be done.

TABLE IV. CRAWLING DATASETS

Tracker Name	**Tracker Statistics**	**Top 10**	**Torrent List**	**Trace Duration (days, mm/dd/yy)**
thePirateBay	√	N/A	(Part)	**31**, 12/29/09 – 01/28/10
TorrentPortal	√	N/A	√	9, 02/02/10 – 02/10/10
TorrentLeech	N/A	√	√	27, 01/15/10 – 02/10/10
RevolutionTT	N/A	N/A	√	**37**, 12/26/09 – 02/10/10
Demonoid	N/A	N/A	√	10, 01/17/10 – 01/26/10
Bitsoup	N/A	N/A	√	**103**, 10/16/09 – 01/26/10
ILoveTorrents	N/A	N/A	√	**119**, 09/28/09 – 01/26/10
CHDBits	√	Top 250	√	**119**, 09/28/09 – 01/26/10
HDStar	√	√	√	**112**, 10/05/09 – 01/26/10
HD-Torrents	√	√	√	19, 01/23/10 – 02/10/10
Torrents.ru	√	N/A	N/A	**55**, 10/01/09 – 11/25/09
DimeaDozen	√	√	√	9, 02/02/10 – 02/10/10
TheBox.bz	N/A	N/A	√	9, 02/02/10 – 02/10/10
AsianDVDClub	√	√	√	9, 02/02/10 – 02/10/10
PureTNA	√	√	√	9, 02/02/10 – 02/10/10

N/A indicates the tracker does not provide it or we cannot acquire it.

There are three challenging issues we need to address: incomplete torrent list of public trackers, information loss caused by crawling time intervals, and limited trace durations.

ThePirateBay only allows displaying recent 50 pages for each torrent category and 300 pages for all torrents, so we cannot depict the whole picture of thePirateBay, but we have crawled as many torrents as we can. On the other side, we have succesfully crawled the complete torrent lists from all private trackers.

The frequency of data collection should be carefully tailored to achieve a smooth crawling. Many trackers enforce a minimum time interval between two page visits. To solve this issue, we estimate the relation between information loss ratio with time interval. We configure our crawlers to execute every 4 hours for tracker statistics and torrent list, every day for Top 10 because its change is not frequent. Complete torrent list can reflect the overall population of torrents in the tracker. Furthermore, we have crawled two popular torrents every 10 minutes to study torrent life span.

Another issue is the limited trace duration because we try to draw unbiased conclusion from these complete torrent lists. To avoid bias when studying complete torrent lists, we refer to the method in [13]. Those torrent lists whose trace durations are longer than 30 days will be studied in our paper. We also take other samples into study but they can be viewed as preliminary results reference.

IV. Measurement and Analysis

In this section, we present our detailed measurement results and analysis. We discuss the user viscosity, single torrent evolution, user behaviors, distribution of private trackers, and the comparison with public trackers.

A. User Viscosity of Private Trackers

According to the traffic ranking (updated data on Feb. 6, 2010) from Alexa [14], we select 6 popular public trackers (thePirateBay.org, Torrentz.com, ISOHunt.com, BTJunkie.org, TorrentReactor.net, Mininova.org) and 6 most famous private trackers (Torrents.ru, Demonoid.com, TorrentLeech.org, PureTNA.com, Bitsoup.org, TheBox.bz) from our crawling list.

Page views measure the number of pages viewed by site visitors. Multiple page views of the same page made by the same user on the same day are counted only once. Page views per user is the average number of unique pages viewed per user per day. Fig. 2 shows the current page views per user of each tracker. It is obvious that private trackers have higher page views per user than public trackers. This implies that users in private trackers have more interest than public trackers. User viscosity of private trackers is better than public trackers.

Next, we take a close look at this viscosity of users by analyzing the number of users in each class of CHDBits. Fig. 3 shows our measured data of CHDBits in a 3-month period[1]. The total number of users keeps relatively stable[2]. In general, stability of total number of users is applicable to other private trackers. It is mainly dependent on two factors: the capacity of the private tracker and the maintenance of the tracker's administrator.

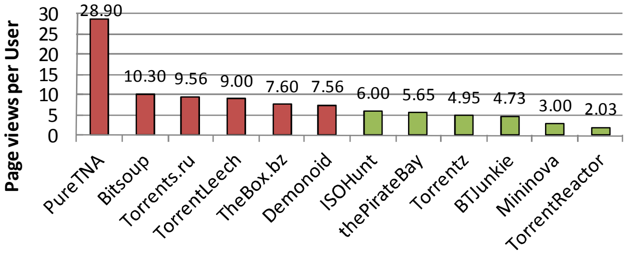


Figure 2. Page views per User in Popular Trackers

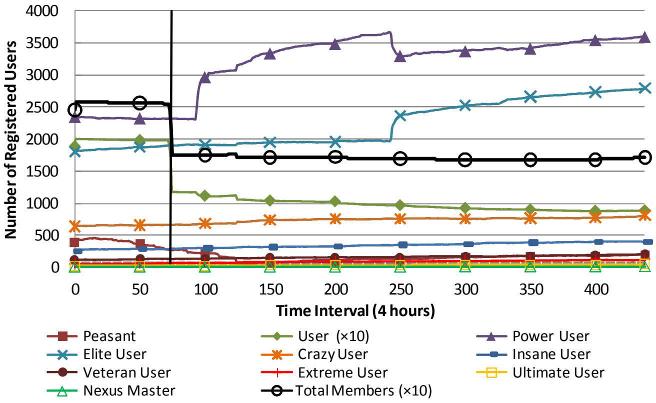


Figure 3. Number of Registered Users of each User Class in CHDBits

Different private trackers adopt different user class policy, but their internal structures are very similar. In Fig. 3, most of the users belong to the User class (i.e., share ratio is lower than 0.9). We observe that at the number of users in User class in decreasing with time, while the numbers of users in Power User and Elite User classes are increasing. It shows the effectiveness of the ranking system in private trackers as users are incented to promote themselves to a higher user class.

B. Torrents of Private Trackers

In this subsection we investigate the behavior of a single torrent and compare with a torrent in public tracker TorrentPortal. Fig. 4 presents the evolution of the number of seeders, leechers, snatched times and the sum of the former three of a single popular torrent in CHDBits. The main finding is that the number of seeders is significantly larger than the number of lechers most of the time. It will be further discussed in subsection C.

Fig. 5 shows the number of peers in an active torrent (i.e., a torrent with at least one seeder) in DimeaDozen and TorrentPortal. Torrents are ordered from the largest to smallest based on the number of peers joined in each torrent. It is denoted as Torrent Rank in the label of X-axis. In Fig. 5, we can see there exist "giant" torrents not only in public trackers, but in private trackers which contain more than 1,600 seeders. Besides, we can see there are more peers in a torrent in private tracker than in public tracker. That guarantees the high download speed in private tracker. Of course, due to the limitation of server capacity, the "giant" scale in private tracker cannot reach the level of public tracker. E.g. in TorrentPortal, there are ten torrents with 8,388,607 seeders,

[1] Notice that the unit of curves "User" and "Total Members" is 10.

[2] The reason of the big drop of total members and User class around time 70 is because the administrator deleted many inactive members.

and some of them even have 8,388,607 leechers. These torrents are usually popular TV series and movies.

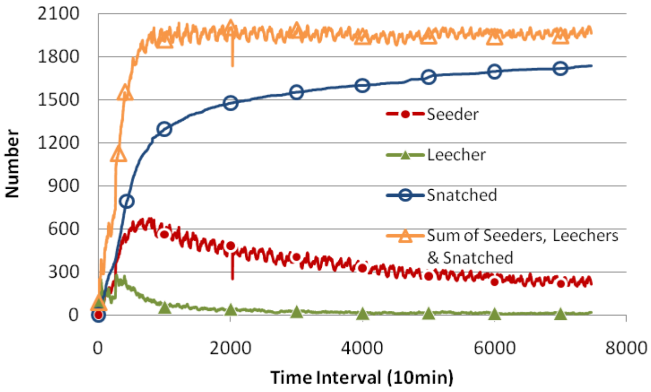


Figure 4. Seeder, Leecher, Snatched & Sum of Single Torrent in CHDBits

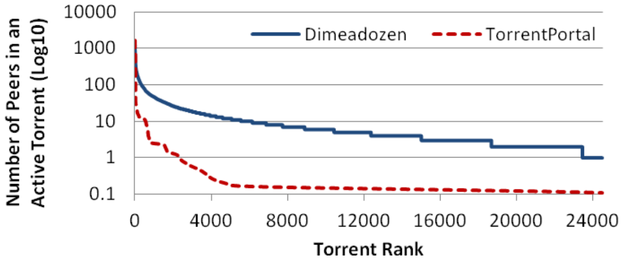


Figure 5. Number of Peers in Active Torrents

Fig. 6 and Fig. 7 show the age distribution and transfer speed of torrents respectively in DimeaDozen (music private tracker). In Fig. 6, about 40% torrents are uploaded in 3 months and 82% of torrents are uploaded within one year. Surprising observation is that there are torrents nearly 3 years that are still active. In Fig. 7, though there are a lot of long live torrents with seeders in private tracker, more than 98% torrents with small leechers have no more than 100KBps in this music private tracker, only those popular torrents attract many leechers and have very high speed. It implies private tracker has a starvation condition that the number of seeders is significantly larger than the number of leechers. Furthermore, because the supplies from seeders exceed the demands of leechers, so the leechers' download speed can be very high in private trackers.

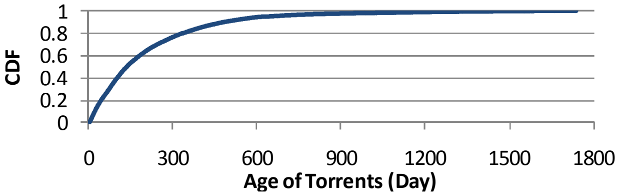


Figure 6. Age Distribution of Torrents in DimeaDozen

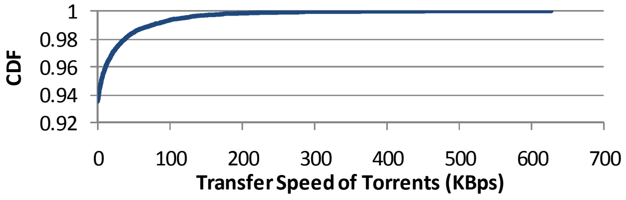


Figure 7. Transfer Speed Distribution of Torrents in DimeaDozen

C. *User Behaviors of Private Trackers*

We have shown that private trackers have better user viscosity, longer seeding time and more active seeding peers, etc. In some degree, they reflect the activity of members in private tracker communities. This subsection presents our findings about user behaviors in private trackers.

1) *Activity of Users*

Active user is a set which includes two parts: active tracker user who is actually participating the content distribution, and active browsing user who is browsing the private tracker web site. Fig. 8 shows the active user ratio (active users to total users) statistics per day and per week in CHDBits and HDStar. In Fig. 8, though these two trackers have different number of users, their active user ratios are surprisingly very close to each other: the per day ratio keeps around 50% and per week ratio keeps around 80%. From Fig. 8 we can see that, compared with active user ratio per week, there still exists nearly 30% members who are inactive per day.

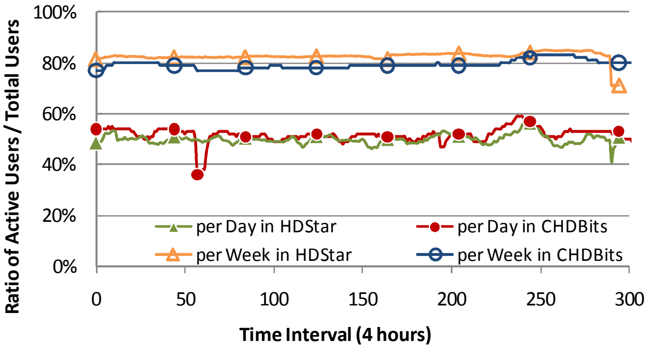


Figure 8. Ratio of Active Users/Total Users per Day & per Week

Then we select one week period (Oct. 13, 2009 to Oct. 20, 2009) in CHDBits and HDStar to compare the change of active browsing user and active users connecting to the tracker (for short, active tracker users). In Fig. 9, the active browsing users in two trackers present the same trend with the change of time. The wave peak happens at 22:00PM every day and the sub-peak is at 10:00AM every day, which means that there are most of browsing users around these two time points. The wave hollow is located at 6:00AM every day.

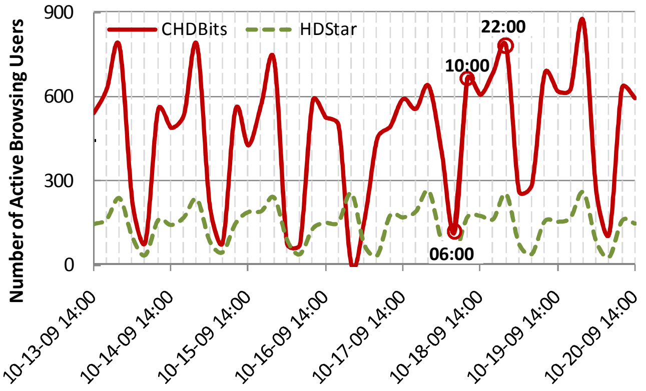


Figure 9. Number of Active Browsing Users

When adding the factor of active tracker users, the result of ratio (active tracker users to active browsing users) in Fig. 10 shows that the wave peak becomes 6:00AM, while the wave hollows are at 10:00AM and 22:00PM. That means most users in private trackers keep staying in the BitTorrent swarm. For example, at 6:00AM, most users are not browsing, but their clients keep connecting to the tracker, so the ratio of active tracker users to active browsing users presents peak status. The reason of the CHDBits irregular wave changes (around 22:00PM, Oct. 17, 2009 to 10:00AM, Oct. 18, 2009) in Fig. 10 is that the tracker suffered non-periodical DDoS attack, which also affects the tracker in the following 2 days, so members' clients have difficulty in connecting to the tracker.

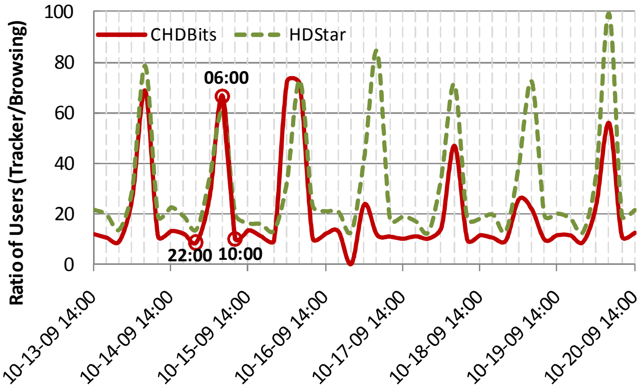


Figure 10. Ratio of Tracker Active Users/Active Browsing Users

2) *Activity of Peers*

In order to further investigate users' behaviors in private trackers, we track the ratio of seeding peers to leeching peers and show the result in Fig. 11. Generally, the number of seeding peers in private trackers is 5 to 15 times of the number of leeching peers[3]. It indicates that users in private trackers are seeding more than leeching. The wave properties of three trackers also follow the interchange pattern of daytime and nighttime.

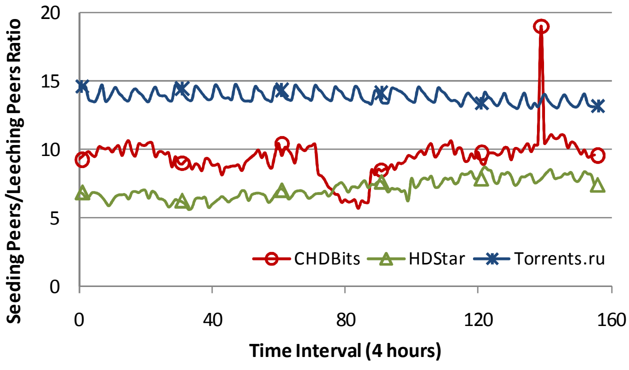


Figure 11. Ratio of Seeding Peers/Leeching Peers in three Pirvate Trackers

3) *Traffic of Private Trackers*

This subsection shows that statistics of the traffic of private trackers. Traffic here indicates the sum of uploaded and downloaded traffic. In Fig. 12, on one hand, we can see that the total traffics per user of CHDBits and HDStar trackers keep increasing from 430TB, 496TB to around 955TB, 851TB, respectively. Fig. 13 shows that total traffic in Torrents.ru is surprising 15-35GBps with nearly 700,000 living torrents. The tremendous traffic may bring heavy burden to our ISPs. On the other hand, the ratios (total uploaded to total downloaded) of two trackers maintain the same value (around 5). This further proved that users in private trackers are doing more seeding than leeching. It reflects a kind of imbalance: supply is more than demand, which also proves that the high download speed in private tracker is partly because of the enormous seeders in the swarm.

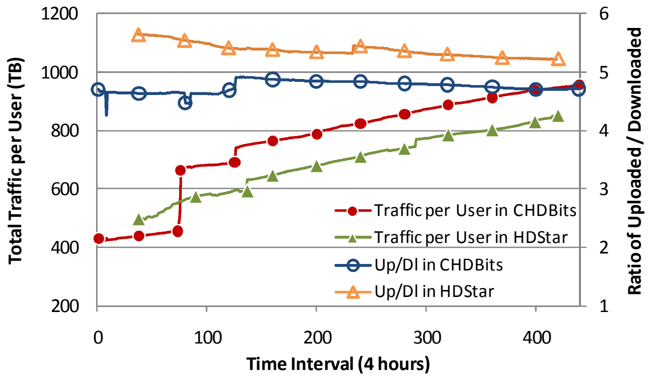


Figure 12. Total Traffic/User & Ratio of Total Uploaded/ Downloaded

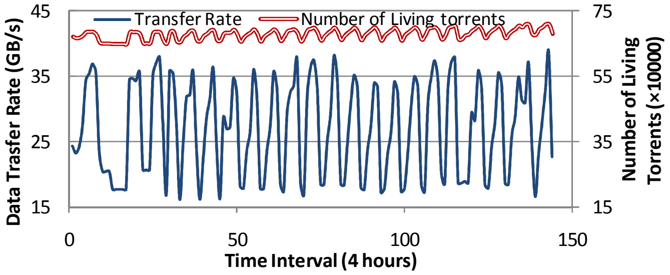


Figure 13. Data Transfer Rate in Torrents.ru

4) *Top 250 of Private Tracker*

CHDBits provides us lists of Top 250 from many aspects, such as best downloaders, best uploader, best share ratios, etc. Fig. 14 shows the change of top 250 members with best share ratios. Though CHDBits have over 15,000 members, there are no more than 250 members who have share ratios of over 10. Actually, most of members only have the share ratio of no larger than 1. E.g., in current DimeaDozen, there are 39% members whose share ratios are lower than 0.25, 75% members whose share ratios are lower than 1.

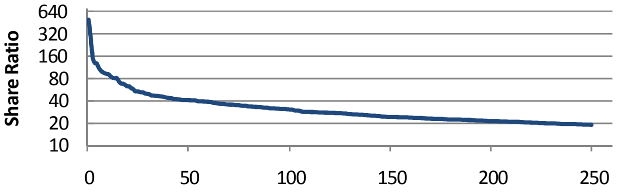


Figure 14. Top 250 Members with Best Share Ratio in CHDBits

Fig. 15 presents the fastest uploading and downloading speed in top 250 members. Combined the results of subsection B and C, we may conclude that some members with better share ratios are not very active. Besides, there are many seeding peers in private trackers, so members can easily download with high speed. This will induce low share ratios,

[3] The abnormal change in CHDBits curve is because the relocation of the server at that time, so the users can hardly connect to the tracker.

and keeping with low share ratios is definitely not safe in private trackers. Therefore, active members have to seed for a long time (shown in Fig. 6), and hungrily aim for safe share ratio to avoid being banned from the trackers.

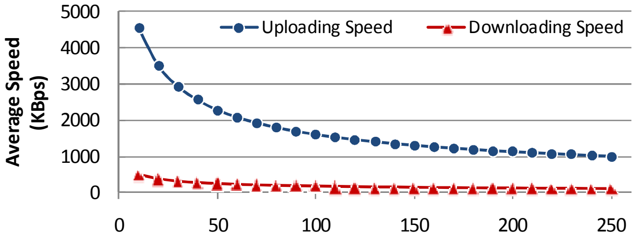


Figure 15. Top 250 Members with Fastest Transfer Speed in CHDBits

5) *Active Torrents Rate of Private Trackers*

Active torrents rate is the ratio of the number of torrents which have at least one seed to the total number of torrents in the tracker. Along with the increasing torrents scale, maintaining the active torrents rate is one of KPIs (Key Important Factor) to evaluate the active level and ranking of trackers. Fig. 16 compares the active seeder rate of 14 trackers. Apparently, most private trackers are far better than public trackers.

We use $N(s)$ indicates the number of seeders in a active torrent, and $N(l)$ indicates the number of leechers in a active torrent. $N(s\_l)$ is the sum of torrents which $N(s) > N(l)$, and $N(l\_s)$ is the sum of torrents which $N(l) > N(s)$. Fig. 17 compares the ratio of $N(s\_l)$ to $N(l\_s)$. Except for AsianDVDClub, all private trackers have more $N(s\_l)$, which means they are more active than public trackers.

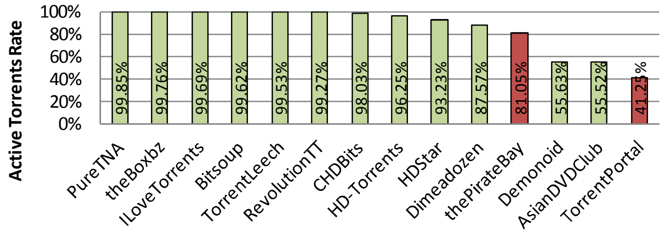


Figure 16. Rank of Active Torrents Rate in Trackers

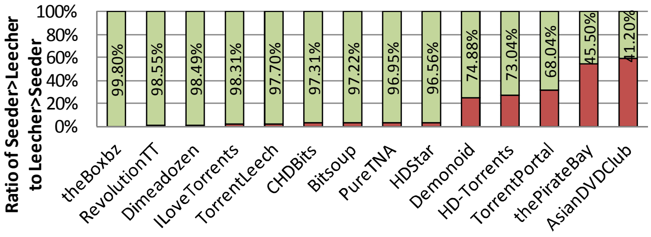


Figure 17. Rank of Seeder > Leecher / Leecher>Seeder Ratio

D. *Content Distribution of Private Trackers*

This subsection shows the content size distribution and category distribution in private trackers. In order to be representative, we choose to show results of one music tracker (DimeaDozen), one HD tracker (CHDBits) and one general tracker (RevolutionTT).

1) *Content Size Distribution*

Fig. 18 shows the classification of content size distribution base on the number of contents in two private trackers, respectively. We can see the most number of contents are small contents with 0-1GB, followed by the contents which are large than 100GB.

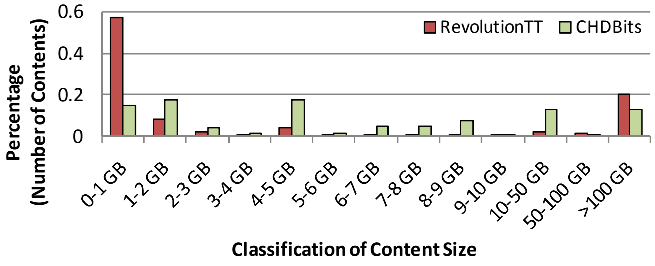


Figure 18. Distribtuion of the Number of Contents

2) *Category Distribution*

Fig. 19 and Fig. 20 show the category of content distribution based on the number of contents in two private trackers, compared with peer distribution respectively. We can see that popularity of content category basically matches the peer distribution accordingly. The DimeaDozen provides detailed categories of music, that is, 61 categories, which shows a long tail distribution. We list the top 5 category percentage in Table V. Fig. 5 shows that TV series, movie and adult are the leading types which occupy 81% share in private tracker. There is a great diversity in the content distribution based on category.

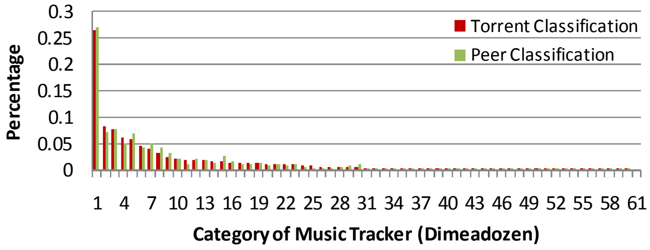


Figure 19. Classification of Peers and Torrents in Dimeadozen

TABLE V. TOP 5 CATEGORIES IN DIMEADOZEN

Category	Rock	Alternate	Jazz	Singer or Songwriter	Progressive Rock
Percentage	27.03%	7.17%	7.63%	5.15%	6.92%

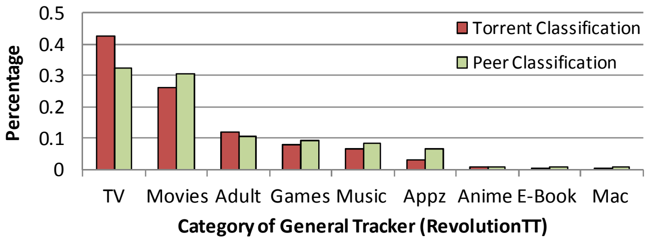


Figure 20. Classification of Peers and Torrents in RevolutionTT

Except for the above measurement, there are other metrics which can measure the performance of private trackers, such as pre-time, firewall rate, freeleech, download slots, etc. We will give more detailed in our technical report.

V. Analysis and Improvement of SRE Mechanism

In this section, we will analyze the effectiveness of private tracker's SRE mechanism based on game theory. Furthermore, we model the mechanism and propose an improved SRE mechanism to incent the users' activity and enhance the quality of service of private trackers.

A. Analysis of SRE Mechanism based on Game Theory

From a macroscopic view, the total download bandwidth is equal to the total upload bandwidth in a BitTorrent swarm. The reason of free-riding problem is that some users contribute a lot but receive unfair return. BitTorrent adopts TFT algorithm as a kind of incentive mechanism, but it has been shown that increased upload contribution only marginally improves download rates [15]. Peers lack the motivation to seed and have no reason to contribute once they have satisfied their immediate demands. In private trackers, we have previously shown that SRE (Share Ratio Enforcement) is a very effective auxiliary incentive mechanism through our measurement study. Here we use a general game-theoretic framework to analyze SRE and answer the following question: Why does the TFT mechanism alone fail to achieve the same level of performance as private trackers that use SRE?

In order to simplify the game, we assume that each peer has two strategy sets, upload (*UL*) and download (*DL*). *N* is the set of natural number.

$UL = \{(ul_1, ul_2, ..., ul_p) \mid p \in N\}$, denotes *p* different upload levels sorted from lowest to highest;

$DL = \{(dl_1, dl_2, ..., dl_q) \mid q \in N\}$, denotes *q* different download levels sorted from lowest to highest.

Each peer selects a pair of (ul_i, dl_j) from *UL* and *DL*, and builds its own strategy space S_k:

$$S_k = \{(ul_i, dl_j) \mid i, j, k \in N,\ 1 \le i \le p,\ 1 \le j \le q\}$$

Then *n* peers in P2P system construct a strategy space set *S*:

$$S = (S_1, S_2, ..., S_n), \quad n \in N$$

In general, peer wants to download by fulfilling its bandwidth configuration or maximum bandwidth. But it is difficult to download as pre-selected in *DL* set. In P2P content distribution network, the actual download performance is decided by the number of active peers joined in the swarm. Here we use the actual download performance of peer *i* to define the utility u_i according to its own strategy s_i. The set of *u* is defined as:

$$u = (u_1, u_2, ..., u_n), \quad n \in N$$

Each peer has an optimal strategy s_i^* from its strategy space S_k. We define the optimal strategy set as s^*:

$$s^* = (s_1^*, s_2^*, ..., s_n^*), \quad n \in N$$

Then the Nash equilibrium can be defined as:

$$u_i(s_1^*, s_2^*, ..., s_i^*, ..., s_n^*) \ge u_i(s_1^*, s_2^*, ..., s_i^{'}, ..., s_n^*),$$
$$\forall s_i^{'} \in S_i,\ s_i^{'} \ne s_i^*,\ \forall i \in N$$

If peers are rational, they will choose to upload with minimum speed and download with maximum speed (ul_1, dl_q). If all peers upload with maximum speed, then peers' utility will become high. If peer *i* chooses $(ul_1^{\,i}, dl_q^{\,i})$ and peer *j* chooses $(ul_p^{\,j}, dl_q^{\,j})$, then peer *i*'s utility will become intermediate, peer *j*'s utility will become low. Therefore, two peers' utilities can be represented in Table VI.

TABLE VI. Utility Function

	Peer *j* (ul_{high})	**Peer *j*** (ul_{low})
Peer *i* (ul_{high})	$(\underline{u_{high}}, \underline{u_{high}})$	$(u_{low}, u_{intermmediate})$
Peer *i* (ul_{low})	$(u_{intermmediate}, u_{low})$	$(\underline{u_{low}}, \underline{u_{low}})$

Based on game theory, we know there are two Nash equilibriums in Table VI. (u_{low}, u_{low}) is strict Nash equilibrium, and (u_{high}, u_{high}) is weak Nash equilibrium. Because peers are rational, so both sides will choose upload with minimum speed to reach the strict Nash equilibrium, and hence Tragedy of Common will occur. Though BitTorrent adopts TFT as a conflict resolution scheme, notice that TFT is based on repeated game and it can only take effect in infinite repeated game. However, there does not exist actual infinite repeated game but only finite repeated game in P2P content distribution. We will get the same result in finite repeated game and in one time game [16]. So the final result of using TFT algorithm induce the strict Nash equilibrium (u_{low}, u_{low}).

But if we introduce SRE mechanism in P2P content distribution system, peers cannot just be rational, and they are enforced to upload/seed to reach certain share ratio under a giving SRE policy. Even if a peer still chooses to upload with minimum speed, it must reach the required minimum share ratio by seeding for a long time. Therefore, the strict Nash equilibrium is broken, and the equilibrium will transit to the weak Nash equilibrium (u_{high}, u_{high}). This is an optimal equilibrium. All peers download and upload with high speed. This will best utilize the potential of all peers and achieve the maximum efficiency of content distribution.

B. Modeling of SRE Mechanism

Here we choose one general tracker (RevolutionTT) and one HD tracker (CHDBits) to model their SRE mechanisms. Their SRE mechanism is defined as Table VII.

TABLE VII. SRE Mechanism

	Downloaded Data	**Share Ratio (no less than)**
RevolutionTT	3.0GB	0.3
	6.0GB	0.50
	10.0GB	0.80
	>10.0GB	>0.8
CHDBits	10GB	0.3
	50GB	0.4
	100GB	0.5
	200GB	0.6
	400GB	0.7
	>400GB	0.95

In general, SRE mechanism is related with two variables: share ratio (*r*) and the volume of downloaded data (*d*). RevolutionTT and CHDBits have different SRE mechanisms. According to Table VII, we found that SRE in both trackers can be modeled as S-Curve: it first starts with a rapid growth stage and then enters into a stable and smooth stage. We model the S-Curve by the following equation:

$$\ln(r) = b_0 - \frac{b_1}{d}, \quad r \geq 0, d > 0, r, d \in R \qquad (1)$$

Through our regression analysis, b_0 is 0.179 and b_1 is -4.326 in RevolutionTT. In CHDBits, b_0 is 0.023 and b_1 is -11.81.

C. Improvement of SRE Mechanism

Though the SRE mechanism achieves a big success in private trackers, we still notice that there are a lot of inactive users. Although administrators can delete inactive users periodically, it is better to improve the activity of users. We will consider the following factors:

- *R*: It is the SRE mechanism used in private trackers.
- *A*: It denotes the Activity of user in terms of seeding time ($\Omega(t_s)$) and the number of seeding torrents ($\Psi(n_i)$).
- *P*: It denotes the Period of time to the last seeding ($\mathrm{P}(t_l)$), the longer to the present, the larger the *P* is.

Our improved SRE mechanism (RAP) is shown as Eq. (2) to calculate the final ratio:

$$Ratio_{final} = r - \mathrm{P}(t_l) + \sum_{i=1}^{n} \Omega(t_s) \times \Psi(n_i), \qquad (2)$$
$$r, t_l, t_s, n_i \geq 0, r, t_l, t_s \in R, n_i \in Z$$

Eq. (2) enforces that a user can not be inactive for a long time; otherwise his final ratio will be decreased by the inactive time period. The final ratio can also be increased by the seeding time and number of seeding torrents.

VI. Related Work

To the best of our knowledge, there are little studies based on measurements, analysis and modeling with BitTorrent private trackers. In 2005, Nazareno et al firstly studied the Share Ratio Enforcement (SRE) in private trackers [17]. They used the same cooperation metrics to compare several BitTorrent communities, including one small scale private tracker (easytree.org). They viewed the SRE as extrinsic gifting which is giving not motivated by a direct, immediate benefit [18]. Then Nazareno et al deeply studied another private tracker (bitsoup.org) with other two BitTorrent communities based on resource demand, supply and their relationship [13]. It seems that private trackers do not attract enough researchers' attention at that time. Until 2009, Tribler team proposed that if a peer uploads more than it downloads in a swarm means the peer should have more credit. Private trackers possibly exist "credit squeeze" which indicates that the lacking of credit leads to significantly reducing of system efficiency [19].

Though the previous investigations revealed some characteristics of private trackers and obtained some primitive research results to SRE mechanism, the noticeable limitation is samples they studied are one or two private trackers, which provides a limited view of current various private trackers. It makes some ideas they proposed stayed in hypothesis stage but not conclusive. Though Chao et al. investigated 800+ private trackers, depicted a broad and clear picture of private trackers landscape [1], there is no clear picture of private trackers inside the box. They did not focus on SRE mechanism, and lacked in-depth studies on some metrics to measure the quality of service provided in a relatively wide range of private trackers.

VII. Conclusion

In this paper, we provide a useful taxonomy of private trackers. In breadth, we have crawled and compared system behaviors among 13 private trackers and 2 public trackers for a 4-month period. We find that private trackers have apparently different features and statistics compared to public trackers, ranging from the user viscosity, single torrent evolution, user behaviors, content distribution and other metrics to measure the quality of service in private trackers. In depth, we study the effectiveness of SRE mechanism based on game theory. Besides, we model the mechanism and proposed an improved SRE mechanism to further incentive the activity of users and enhance the quality of service of private trackers.

References

[1] C. Zhang, P. Dhungel, Di Wu, Z. Liu,and K.W. Ross, "BitTorrent Darknets," Infocom 2010, San Diego, 2010.

[2] "The Pirate Bay," http://thepiratebay.org/.

[3] "Mininova," http://mininova.org/.

[4] "isoHunt," http://isohunt.com/.

[5] "The 100 Most Popular Private Trackers Of 2008," http://filesharefreak.com/2008/12/31/the-100-most-popular-private-trackers-of-2008/, December 31, 2008.

[6] "Private Trackers with the Most torrents," http://filesharefreak.com/2009/09/19/private-trackers-with-the-most-torrents/, September 19, 2009.

[7] "Torrentking.org," http://www.torrentking.org/?order=rating.

[8] "The Essential Guide to Getting Into Private Trackers," http://filesharefreak.com/2009/06/18/the-essential-guide-to-getting-into-private-trackers/.

[9] "CHDBits," http://chdbits.org/faq.php.

[10] "TBDEV.net,"http://sourceforge.net/projects/tbdevnet/.

[11] "TS SE," http://templateshares.net/.

[12] "Trackers List," http://filesharefreak.com/trackers-list/.

[13] Nazareno Andrade, Elizeu Santos-Neto, Francisco Brasileiro, et al. "Resource demand and supply in BitTorrent content-sharing communities," Computer Networks Journal, December 2008.

[14] "Alexa," http://www.alexa.com/.

[15] M. Piatek, T. Isdal, A. Krishnamurthy, T. Anderson, "One hop reputations for peer to peer file sharing workloads," in NSDI08.

[16] Gibbons, R, "A Primer in Game Theory," Harvester, New York, 1992.

[17] Nazareno Andrade, Miranda Mowbray, Aliandro Lima, et al. "Influences on cooperation in BitTorrent communities." Third Workshop on the Economics of Peer-to-peer Systems, SIGCOMM 2005, August 2005.

[18] Matei Ripeanu, Miranda Mowbray, Nazareno Andrade, and Aliandro Lima, "Gifting Technologies: A BitTorrent Case Study," First Monday, vol. 11(11), November 2006.

[19] D Hales, R Rahman, B Zhang, M Meulpolder, et al. "BitTorrent or BitCrunch: Evidence of a credit squeeze in BitTorrent?" In the Proceedings of the 5th Collaborative Peer-to-Peer Systems (COPS) Workshop, June 29 - July 1, 2009.

Understanding the Cultural Influence on Tagging Pattern
- The First Step to Design User-Centric Recommender System

Ho Keung Tsoi

Abstract

To design a user-centric recommender system, the first and foremost task we need to do is understanding the users. In this regard, this paper starts by evaluating tagging behavior from different cultures, and introduce our future works.

Tagging behavior in social network has been studied in great detail by peer researchers, and various metrics are proposed to evaluate such patterns. However, not many studies have dealt with the important cultural dimension associated with tagging manners. To this end, this study investigates the tagging behaviors in SongTaste and Last.FM, two tagging-enabled music social networks targeted at Eastern Asian users and European users respectively. We examined tag agreements among friends and members, what kinds of tags are favored by different cultures, tag discrimination and the non-obviousness of tag. Our results suggest that the cultural dimension significantly matters the tagging behavior.

We conclude with a discussion on the potential impacts of our findings and on-going works.

1 Introduction

User-centric design of recommendation system is to develop a recommendation system addressing the users' needs. Unlike the existing systems, they were developed in a way regardless of the target subject. Their approach is good for the generic purpose, but not sufficient to meet the individual needs or neglected the importance of contextual information, such as the cultural originality of user. To this end, our works attempt to understand and evaluate the fundamental needs of users, and hence we investigate the tagging behavior in our first step.

Collaborative tagging is the process of describing a resource by user-created annotation. It could be in the form of short-phrases or keywords, in which adding new contextual dimensions(metatag) to resources by mean of the resources' type, discipline, content or strings that are meaningful to certain users [1, 5]. This has been getting more and more common as many popular social websites, such as Flickr, Last.FM, del.icio.us and Digg, deploying this mechanism.

The phenomenon of adopting tags into recommender systems in order to better calculate user-user similarity in collaborative filtering, given that tags may infer users detailed preferences on item features, are well studied by a number of researchers [3]. However, most of their works are focusing on research citation sites (e.g. CiteULike), or webpage bookmarking sites (Delicious) [9, 10]. In this lack of detailed investigation of tag analysis in other domains, we are particularly interested in studying social music sites, one of primarily targeted product domains in social recommender systems.

On the other hand, it is observed that the cultural dimension would matter human behavior. Take the example of social greeting, western people will hug each other for politeness's sake, whereas eastern asian won't contact so closely in this way. With this observation, we speculate the tagging behavior will also be impacted the cultural origin of users. Therefore, driven by the existing limitations and observations, our work reveal the differences of tagging behavior of users from different cultural backgrounds when they are actively interacting with a music sharing sites. We expect that the analysis results can be suggestive to related works on recommenders and social search, so as to best leverage the gap between cultural originality and tagging patterns. To the best of our knowledge, we are the first to probe into the cultural effect on tagging behavior.

In particular, this paper addresses the following questions:

RQ1: What is the tag agreement among friends in both cultures?

RQ2: What is the tag agreement among members in both cultures?

RQ3: What is the tag non-obviousness index in oriental users compare with western user?

RQ4: How is the tag discrimination value change from Eastern Asian to European?

RQ5: How the tags classes distribution diverse from oriental users to western user?

And these in turn will be further explained in subsequent sections.

2 Related Works

If a tag is not appearing in item's content, the item's description for instance, this tag will be valuable since it adds extra information to that item. Furthermore, the ability of a tag distinguishing between this collection of resources and other resources can also be measured. These two evaluations are modeled by tag non-obviousness and tag discrimination [4]. Tags can be classified into different categories. Golder and Huberman [5] present a classification scheme in which put the tags into seven categories. Sen et al. [11] further refine the seven categories into three more general classes. One can better understand the tagging behavior at the level of categories of tags.

In social network, one of the characteristic is the collaboration between users. Users can explicitly state the friendship with others. They may be real world friend, or they just share common interest in a virtual world and thus become a virtual friend. This kind of relationship always bidirectional, that is, user has to confirm the friend request before the relationship established. But this depends on individual system design. Studies [3, 8, 10] show that the presence of friends will affect user's tagging pattern in certain extent. Such as re-applying the tag his/her friends used before.

Additional to friendship, various social ties can also be found in social network [9, 10, 11]. People join the same discussion group or participate in a common event are some of the examples. Users join the same group implied that they share some common interests. This connection could be shown in their tagging patterns.

Another important aspect is the cultural origin of tagger. It is observed that the cultural dimension would matter human behavior [6]. Take the example of greeting, European like to have a close hug, while Chinese reserve some distance to others. This cultural divergence in human behavior also found in the preference of product recommendation interfaces and attention on object [2, 7]. Oriental users are found to be holistic, while western are more analytic.

3 Experimental Materials

3.1 Dataset

To emphasis the cultural dissimilarity, our experimental data was crawled from two popular music social websites: SongTaste[1] and Last.FM[2]. They have more than 2.3M and 30M registered users resp. And they also share similar features, like allowing users to listen to the song, leave comments on it, and different rankings are available. And more importantly, they have different target user groups, namely Chinese and European, which fit our investigation.

The dataset consists of 200 popular songs each from SongTast and Last.FM (cut-off date is 6t December, 2009). Users commented on these songs and their friends were considered, also for the tags applied by them. In summary, 6,500 users (each applied at least one tag) from the two websites were selected for analysis throughout this paper,the average number of tags applied are 10.3 (SD 74.47) and 62.1 (SD 36.34) in SongTaste and Last.FM respectively.

4 Metrics

In the following analysis, unless otherwise specified, t-test assuming unequal variances with a risk level of 0.05 is used. Table 1 summarized our results.

4.1 Tag Agreement among Friends & among Members

RQ1: What is the tag agreement among friends in both cultures?
RQ2: What is the tag agreement among members in both cultures?

To measure how the tags agree with each other, we use the Symmetric Jaccard Coefficient [3, 10], which is the fraction of tags common in both users.

$$\frac{|T_{user} \bigcap T_{friend}|}{|T_{user} \bigcup T_{friend}|} \tag{1}$$

The tag agreement among friends are found to be 0.004 and 0.086 from SongTaste and Last.FM respectively (*t*= 1.96, $p < 0.05$). This indicates that the friendship agreement in western users is significantly stronger than oriental users.

We then measure the tag agreement among members in a similar manner. To achieve so, we need to find out which discussion group(s) the subject user has joined, and then all the members belong to these groups. The next step was to find all the tags used by the subject user and members.

Aftermath, we can apply equation 1 to obtain the value. The coefficients of SongTaste and Last.FM are 0.001 and 0.071 respectively (t = 1.96, $p < 0.05$), which indicates that the agreement among members in western.

We also compare the different tag agreements within the system, which is the cultural independent difference of tag agreements. In the Chinese groups, the agreements among friends is 0.001 and that in members is 0.002 (by t-test: paired two-sample for means, *t*=1.96, $p < 0.00$), while the values in European groups are 0.111 and 0.097 (by t-test:

[1] http://www.songtaste.com, Chinese as primary user

[2] http://last.fm, European as primary user

Table 1. T-test results for different comparisons

	Friends Agreement		Members Agreement		Within Domain				non-obviousness	
	SongTaste	Last.FM	SongTaste	Last.FM	ST_{frd}	ST_{mem}	FM_{frd}	FM_{mem}	SongTaste	Last.FM
Mean	0.0006328	0.1105970	0.002105	0.0973138	0.00063	0.00210	0.11059	0.09731	0.93023	0.95873
Variance	0.00002998	0.005808	0.000486	0.00520	0.00002	0.00048	0.00580	0.00520	0.01778	0.00215
P value	0		0		8.6×10^{-8}		4.47 $\times 10^{-35}$		0.00477	
t Critical two-tail	1.960325		1.960271		1.96032		1.96032		2.59604	

Table 2. Examples of tags in the three categories in [11]

Class	Definition	Examples
Personal	They are often used to organize a user's own resources	ok computer, new prog, always in my mind
Subjective	Express people opinions related to a web resource	amazing, awesome, favorites
Factual	Identify facts about the described web resource	beatles, pop, electronic

Table 3. Examples of tags in the three categories in [11]

Class	Definition	Examples
Cat. 1	Identifying What (or Who) it is About.	john lennon, male vocalists, hard rock
Cat .2	Identifying What it is.	singer-songwriter,original, experimental
Cat .3	Identifying Who Owns It.	beatles, pop, electronic
Cat .4	Refining Categories.	60s, uk, orchestral
Cat .5	Identifying Qualities or Characteristics.	good, relax, perfect
Cat .6	Self Reference.	jumping, memories, god
Cat .7	Task Organizing.	hand claps, dance, party

paired two-sample for means, *t*=1.96, $p < 0.00$). Details result is shown in Table 4.

These results are in line with our observations. The social affiliation in European is stronger than in Eastern Asian. People in the former group share more with others, they always have gathering with friends for instance. Consequently, the values of tag agreement in both friendship and membership are larger comparing with the Chinese one. The larger value in the friendship agreement in the European implied that the relationship of friend embedded the interest sharing information. People have friendship may somehow due to the sharing of common interest.

4.2 Tag non-obviousness

RQ3: What is the tag non-obviousness index in oriental users compare with western user?

Tag non-obviousness is the ratio of tags not appear in the content to the total number of tags of that item (200 songs in our case). To analyze the tag non-obviousness in both systems, we additionally crawled the items' content, and compare against the tags designated to that item.

$$\frac{|T \notin Content|}{|T \in song|} \times 100\% \tag{2}$$

93% and 95% of the tags in SongTaste and Last.FM respectively are found to be non-obvious (*t*=2.60, *p* = 0.004). This suggests users in both cultures tend to apply tags that put intellectual value to the item. In other words, people in both cultures prefer to associate tag that is not appear in the content (for example, words appearing in the title). They might think that applying obvious tag is redundant, and no information can be gained in this way. Instead, they apply tags that hold extra information and can help themselves or others to more understand the items.

Also from the t-test analysis, though the result is shown to be significant, we can see that the difference in tag agreements in the two cultures is more manifest. Consequently, we draw the conclusion that dependence of cultural origin and 'obvious' of tag is not as important as those in tag agreement and cultural origin.

4.3 Tag Discrimination

RQ4: How is the tag discrimination value change from Eastern Asian to European?

Based on our 400 popular songs, we can measure the tag-discrimination values using the formula proposed in [4], with the distinct papers substituted by distinct songs in our case.

$$\frac{\sum(\#\_of\_distinct\_songs\_for\_each\_tag)}{\#\_of\_tags} \tag{3}$$

To the two extreme cases, when each song is just tagged one, we have the lower bound of 1.0 songs/tag; alternatively, when we have the upper bound of 200 songs/tag when each song is associated with every other tag. And therefore the tag discrimination value should lie between the lower and upper bond inclusively. With these considerations, we calculate the tag discrimination values of SongTaste and Last.FM to be 2.16 songs/tag and 53.18 songs/tag respectively.

From the information theory perspective, the information gain that a tag provides in Last.FM is much larger than that in SongTaste. And the tags in the former system are well discriminating the resources generally. This result is expectable, since the number of tags in Last.FM is more than in SongTaste, and we didn't process the semantically related tags, and hence the result is reasonable.

4.4 Tag Classes Distribution

RQ5: How the tags classes distribution diverse from oriental users to western user?
Due to only a small portion of overlap in the 400 songs mentioned above, another twenty songs common in both systems and their tags were extracted to perform the tag classes' analysis. We then manually identify the 1,313 tags into the seven categories in [5] and also the three categories in [11]. This is to give reader a full picture of how the classes defined and emerge, and also avoid the deficiency of a particular scheme. Here we give some examples of what tags belong to which class in Table 2 and Table 3. The 7-class classification scheme is the first to put tags into different categories in del.icio.us, a collaborative tagging system for web bookmark, and therefore some of the categories are specific for this domain. As for the 3-class classification scheme, it is based on the former one, and is designed for generic purpose.

Figure 1 and Figure 2 show the tag classes distribution under the two classification schemes. We can see that the Factual class or Cat. 1 take a major proportion of tags in both cultures whereas the difference lie in the rest of the classes. The European tags span wider range of variety compare with the Chinese one. This also inherent the cultural characteristic that European would like to express them more as reflected in their tagging pattern. As in the case of other areas, Chinese prefer the one with higher utility value. Readers should note that there are no tags fall in Cat. 3 under the classification scheme of [5] in both websites, this is because Cat. 3 is not applicable to our domain.

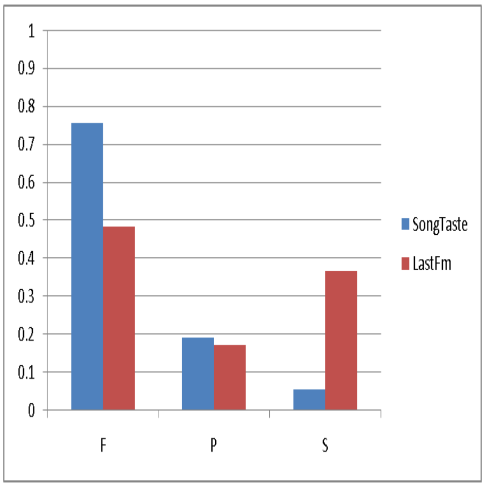


Figure 1. 3-class classification

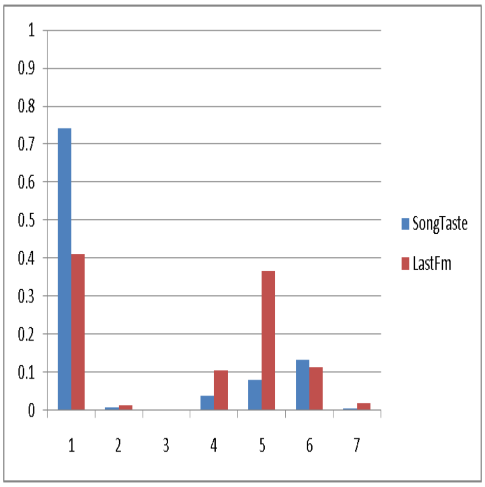


Figure 2. 7-class classification

5 Discussion

The results presented in the previous section suggest that oriental users and western users share some common characteristics in tagging behavior, as well as some dissimilarity.

Similarity:

S1: The agreement among friends is not significant in both systems. Only a small portion of user's tags is overlapped with the tags applied by his/her friends. (Friends agreement)

S2: Most of the tags assigned by users, regardless of cultural origin, are belonged to the Factual class. (Tag classes distribution)

S3: The tags assigned by both cultures give additional information to the item. (Tag non-obviousness)

Dissimilarity:

D1: The agreement among users who join the same discussion group have more tags in common in the case of European. (Member agreement)

D2: European would like to express their feeling towards the song to a higher degree. (Tag classes distribution)

D3: Tags applied by the western group are more valuable in identifying items. (Tag discrimination)

It seems that the overall social affiliation is stronger in western users than oriental users. Western users are willing to share more with each others, while the individualism is stronger in oriental users. This may inherit from the traditional cultures. Besides, the popularity of web 2.0 technology in oriental users is yet as good as that in western user, it can be revealed from the participation rate in both systems.

Furthermore, the low overlapping in Within Domain comparison found in SongTaste, could be explained by the goal of personal information management [9]. The objective of using tags for Chinese users might due to the personal information management purpose.

Based on these findings on dissimilarity in tagging behavior, we suggest designers can embed this information in their system design. One can also develop a cultural-specific tag recommendation algorithm. The performance of such algorithm should outperform those traditional algorithms for generic purpose.

5.1 Limitation

There are far more aspects of tagging patterns can be analysis, such as re-tagging, tagging trend, so on and so forth. However, we got limited access to the real world data. We have discovered many Chinese websites that are similar to those popular in the western during the data collection stage, but many of them do not provide API nor enough information such as the tagging timestamp, who has tagged what item, for analysis. To better study the cultural difference in tagging behavior, we suggest researchers to create a experimental system to collect the data, and hence have more control and data available.

6 Conclusion and On-Going Progress

Apart from the tagging behavior analysis, we also conducted an online survey[3] and implemented the traditional cosine similarity based CF recommender system [4]. The purpose of this online survey is to collect contextual information, such as age group, operating system using, for later investigation on how these factors can infer user preference in choosing movies.We might train a Bayesian Network to predict user preference given the OS they are using.

As for the recommender system, various recommendation algorithms, and tag-recommendation approaches will be available in the near future, and this will act as a baseline comparison of our recommendation algorithm.

6.1 Proposed Recommendation Algorithm

Taking the cultural dimension into consideration, we design a recommendation algorithm in such a way taking care of this information.

Our approach is to differentiate users into two groups. One group has a particular interest or personality, and the another one doesn't possess any interest at the initial stage. This fit our observation that the cultural originality affect user behavior, and therefore the interest and personality. The neutral group, or the later one, simply represent those who do not affect by the cultural dimension.

We propose that when an individual from the neutral group situated in a group with certain interest. A person who don't read too much but inside a book fair for example, he or she will somehow interested in reading or purchasing some books. So in our model, the neutral one will be influenced by the surrounding information (i.e. groups with certain interest.).

This phenomenon is similar to the case of electrical charge. In the study of electromagnetism, there exist substance with potential difference (individual with certain interest), as well as substance which is neutral in nature (individual who is neutral). And the neutral substance will have induced p.d. when there is charged object nearby. Physicists use the inverse-square law of distance to model the aforementioned physical property in object. We propose to borrow this idea to our case.

The present of others will determine the current user preference, and this influence will be decay as two individuals are further apart. There are also many models in the physical science can be referred to. In the future, we will choose a model from them and modify it to suit our case, so as to address the contextual dimension and give solution to the cold-start problem in recommending items.

[3] http://www.comp.hkbu.edu.hk/ hktsoi/exp2

[4] http://158.182.9.112/rec.php

References

[1] H. S. Al-Khalifa and H. C. Davis. Towards better understanding of folksonomic patterns. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 163–166, New York, NY, USA, 2007. ACM.

[2] L. Chen and P. Pu. A cross-cultural user evaluation of product recommender interfaces. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 75–82, New York, NY, USA, 2008. ACM.

[3] J. S. et al. Leveraging tagging to model user interests in del.icio.us. In *Proceeding of the AAAI Spring Symposium on Social Information Processing*, 2008.

[4] U. Farooq, T. G. Kannampallil, Y. Song, C. H. Ganoe, J. M. Carroll, and L. Giles. Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 351–360, New York, NY, USA, 2007. ACM.

[5] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, 2006.

[6] I. Lee, G. W. Choi, J. Kim, S. Kim, K. Lee, D. Kim, M. Han, S. Y. Park, and Y. An. Cultural dimensions for user experience: cross-country and cross-product analysis of users' cultural characteristics. In *BCS-HCI '08: Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008*, pages 3–12, Swinton, UK, UK, 2008. British Computer Society.

[7] K. C. I. N. A. Nisbett, Richard E.; Peng. Culture and systems of thought: Holistic versus analytic cognition. In *Psych. Review, Vol.108*, pages 291 – 310, 2001.

[8] O. Nov, M. Naaman, and C. Ye. What drives content tagging: the case of photos on flickr. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1097–1100, New York, NY, USA, 2008. ACM.

[9] E. Rader and R. Wash. Influences on tag choices in del.icio.us. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 239–248, New York, NY, USA, 2008. ACM.

[10] E. Santos-Neto, D. Condon, N. Andrade, A. Iamnitchi, and M. Ripeanu. Individual and social behavior in tagging systems. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 183–192, New York, NY, USA, 2009. ACM.

[11] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, New York, NY, USA, 2006. ACM.

Vaccination Deployment in Protection against Influenza A (H1N1) Infection

Shang XIA

Abstract

Vaccination is an effective way to control epidemic spreading by adjusting the composite structure of susceptible, infected and vaccinated populations. Three vaccine deployment factors can affect the H1N1 infection dynamics, they are: (1) total amount of vaccine, (2) vaccine releasing time, and (3) vaccine distribution method. Yet the impact of these deployment factors still remain to be systematically understood. In our study we develop an SIV (susceptible, infected and vaccinated) model that incorporates 6 age-grouped populations, and a survey-based contact matrix. We parameterize this model with current H1N1 influenza parameters. The developed SIV infection equations for each group enable us to simulate both within- and between-group epidemic spreading dynamics. Under different vaccination deployment settings, we observe that enough vaccine availability can lower the final percentage of infected population. Releasing vaccine before infection stage transition can improve the dynamical infection process. Vaccination by infection force can improve the efficacy of vaccine deployment.

1 Introduction

The outbreak of swine-origin influenza A (H1N1) all over the world led the World Health Organization (WHO) to declare an influenza pandemic on June 11, 2009 [3]. Vaccination is a critical way to control epidemic spreadings by adjusting the composite structure of susceptible, infected and vaccinated populations [12]. However, in real practice, the effective deployment of vaccines for a pandemic virus might be limited by several factors, they are (1) how many doses of vaccine are required; (2) when vaccines are available to start to be released; (3) what is the distribution priority for individuals with different demographical features. In addition, prior to and during the vaccine deployment process, the composite structure of infected population are dynamically evolved as a result of virus spreading. These vaccine deployment factors and infection dynamics will change the landscape of epidemic spreading and should be considered in order to schedule an effective vaccine deployment plan.

The same as the previous influenza pandemic experience, current H1N1 infection is characterized by age distributions that a higher attack rates and increased proportionate mortality are associated with younger population[9] [5]. This heterogeneity of infection vulnerability by age will have important implications for optimal vaccination distribution strategies.

To control the H1N1 infection spreadings in US, the Centers for Disease Control and Prevention (CDC) projected a vaccination plan that a total of 45 million doses would be available by mid-October, followed by 20 million doses every week thereafter [1]. Under this plan H1N1 vaccine was distributed mainly to health care personnel and young children [2] with the concern that it is more efficient to protect those who are more vulnerable to be infected.

Recently many discussion have been proposed to debate whether a higher priority for vaccination distribution method based on individuals' infection vulnerability is really effective[14][15]. It has been suggested that an vaccine distribution strategy based on individuals' transmissibility would have a greater impact on reducing overall infection than the current practice of focusing vaccination efforts on infection vulnerable populations [7] [10]. In particular, the potential benefit of preferentially vaccinating school-aged children has been discussed, since this age group is disproportionately responsible for influenza transmission [13][6][4].

However all of these discussions mentioned above ignore a critical aspect of vaccination deployment in real practice – the availability of vaccine, such as the total amount of vaccine doses and the available time for first batch of vaccine releasing. Given limitations of our knowledge of a newly emerged virus and constraints of production and logistical ability, the time of vaccine releasing is always delayed than infection dynamics and the amount of available vaccine are always a fractal of current needs. Based on these practical limitations, how to optimally distribute vaccine requires a deep understanding of relative impacts of vaccine deployment factors on disease infection dynamics.

To address this question mentioned above, we developed an age-structured infection model to describe the H1N1 virus spreading during 2009. We use this model to observe the infection dynamics under different vaccination settings, such as the amount of available vaccine, vaccine releasing time and the vaccine distribution methods. Based on the results of our simulations, we analyze the impact of each vaccination deployment factors on infection dynamics, which can provide an solid foundation for design an optimal vaccination deployment plan in real practice.

The remainder of this paper is organized as follows: Section 2 presents detailed epidemic infection model. Section 3 is about the vaccination deployment strategies. Section 4 provides simulation experiment and results analysis. Section 5 conclude the whole paper and highlight the major contribution of this paper.

2 SIV Infection Model

In this section, we will present the detailed formulations of SIV virus infection model, introducing the age structure of host population and the contact relationship among them.

2.1 Model Formulation

In our model, each individual's health status will be labeled as S (susceptible), I (infected) or V (vaccinated) during virus infection process, and the composite structure of these three compartmental population is a representation of infection dynamics. We represent the heterogeneity of virus infection by dividing population into different age groups, which have different infection parameters, and depict the heterogeneity of virus spreading based on population's contact structure both within and cross each age groups. Finally, We use a set of difference equations together with virus infection and spreading parameters of current H1N1 virus to model the dynamics of H1N1 epidemic spreading process. More details will be presented in the following subsections.

2.2 Population Age Structure

During virus infection process, the response to an infection exposure are quite different for people with different demographical features. For example, the probability of a successful infection might be totally different in terms of individuals' virus resistibility. These difference can be generalized as the heterogeneity of virus infection which is viewed as the result of diversities in individuals' demographical features, such as, age, gender, ethnicity and so on. With the aim of analyzing the impact of the heterogeneity of population's infection vulnerability, we introduce population's age structures that each individual will be divided into groups in terms of their age and each age groups will have a set of parameters of virus infection rate and recovery rate to represent the diversities in individuals' vulnerability.

In this model, the whole population are divided into 6 age groups, which contain A_1(0-4), A_2(5-14), A_3(15-24), A_4(25-44), A_5(45-64), A_6(64+). The number of population in each age groups are parameterized by the demographical statistics of United Kingdom in 2007 by Office for National Statistics of UK [16].

Table 1: Age Structure of Population in UK (2007)

Age Groups	Population (Million)	Percentage (%)
A_1 (0-4)	3.446	5.7
A_2 (5-14)	7.380	12.2
A_3 (15-24)	7.841	13.0
A_4 (25-44)	17.156	28.5
A_5 (45-64)	14.738	24.5
A_6 (65+)	9.699	16.1
All	60.26	100

Based on the above mentioned population age structure, the epidemiological parameters of each age group are listed in Table 2. Infection rate β is the probability of an successful infection when an susceptible individual is exposed in an infectious contact. We propose that the heterogeneity of individuals' infection rate is the result of the natural immunization ability of each age groups. Based on Miller's cross-sectional serological survey on H1N1 immunization in different age groups [8], we can parameterize the diversity of individuals' infection vulnerability.

Recovery rate σ describes the percentage of infected individuals that will be recovered in a computing time unit, which is the reverse of the infection periods. On the basis of data regarding viral shedding from studies of seasonal influenza, most patients with flu infection might shed virus from 1 day before the onset of symptoms through 5 to 7 days after the onset of symptoms or until symptoms resolve; in young children or severely ill patients, the infectious period might be longer. Because of the limited knowledge of infection period, in our model the recovery rate for each age groups are homogeneously model as 0.143, which means the infection period is 7 days.

2.3 Contact Matrix based on Age Groups

In our study, we assume that virus infection spreading is the result of contact activities between the susceptible individuals and the infectious ones. To study the virus spreading dynamics, first, we should provide a foundation of individual's contact relationship. In this paper, we use a contact matrix $C = \{c_{i,j} | i, j \in (1, 2, ..., N_g)\}$, N_g is the number of age groups, here $N_g = 6$, to characterize contact frequency within and between each age groups. $c_{i,j}$ means the average times of contact for an individual in age group i with individuals in age group j. We parameterize the ma-

Table 2: H1N1 Epidemiological Parameters based on Age Groups

	A_1(0-4)	A_2(5-14)	A_3(15-24)	A_4(25-44)	A_5(45-64)	A_6(64+)
Infection Rate β_i	0.213	0.420	0.206	0.206	0.15	0.313
Recovery Rate σ_i	0.143	0.143	0.143	0.143	0.143	0.143

trix elements $c_{i,j}$ by using the results of a study in European countries which keep a record of participants' daily report about their contact activities [11].

Based on the basic contact frequency matrix in Mossong's results, which provide a contact matrix of all reported contacts consisting of the average number of contact persons recorded per day per participant in his survey, we can get an estimation of the number of total contacts, that is $M_{i,j}$, between age group A_i and group A_j.

Thus, the total number of contacts from group A_i to group A_j should be equal to that of contacts from group A_j to group A_i. So we have

$$c_{ij} \cdot P_i = M_{ij} = c_{ji} \cdot P_j \tag{1}$$

P_i, P_j are the total number of population in age group A_i and A_j. So the element c_{ij} of contact matrix C denotes the average contacts between an individual in age group i with individuals in age group j.

The value of contact matrix $C = \{c_{i,j} | i, j \in (1, 2, ..., N_g)\}$ is presented in table **??**.

Table 3: Contact Matrix

	A_1	A_2	A_3	A_4	A_5	A_6
A_1	1.49	1.165	0.658	2.396	0.938	0.303
A_2	0.613	4.130	0.734	2.276	0.805	0.403
A_3	0.300	0.634	2.794	1.218	0.927	0.488
A_4	0.528	0.953	0.590	1.634	0.842	0.408
A_5	0.218	0.356	0.474	0.888	1.049	0.589
A_6	0.109	0.276	0.387	0.669	0.914	1.484

2.4 SIV Model

In this section, we propose a modified population based virus infection model, SIV model, to simulate the dynamics of virus infection process. In this model, we introduce three status to label individuals' state transition, which include susceptible state **(S)**, infected state **(I)** and vaccinated state **(V)**. Individuals' status can transit from one to other based on the result of whether they are infected, recovered or vaccinated by exterior interventions in current computing unit. As mentioned in the previous sections, the population in our model are divided into 6 age groups, which is aimed at representing the heterogeneity of virus infection in terms of their age distribution. Based on this age structure, we assume that individuals within an age group are homogenous, which means they have the same parameters of infection rate and recovery rate and the identical probability to contact with individuals of outside age groups. We also propose that the heterogeneity of virus infection and virus spreading are presented by the value of virus infection parameter sets and the matrix of cross group contact frequency.

Within age group A_i, variable $S_i(t)$, $I_i(t)$ and $V_i(t)$ can represent the relative percentage of the susceptible, infected and vaccinated population. Thus the vector $\{(S_i, I_i, V_i) | i \in (1, ..., 6)\}$, can describe infection dynamics within each age groups. The virus infection dynamics among individuals in each age group can be described by the following difference equation:

For age group A_i, $i \in (1, ..., 6)$, k is the current computing moment,

$$\begin{aligned} S_i(k+1) =& S_i(k) + \tau_i \cdot I_i(k) + (-\lambda_i(k)) \cdot \beta_i \\ & \cdot [S_i(k) - \Delta v_i(k)] + (-\Delta v_i(k)) \end{aligned} \tag{2}$$

$$\begin{aligned} I_i(k+1) =& I_i(k) + (-\tau_i) \cdot I_i(k) + \lambda_i(k) \cdot \beta_i \\ & \cdot [S_i(k) - \Delta v_i(k)] \end{aligned} \tag{3}$$

$$V_i(k+1) = V_i(k) + (-\Delta v_i(k)) \tag{4}$$

In the moment of k, for the virus infection difference equations of age group A_i, $\tau_i \cdot I_i(k)$ represents the recovered case of infections. $\lambda_i \cdot \beta_i \cdot [S_i(k) - \Delta v_i(k)]$ is the newly increased infections which is the result of infectious contact activity within and cross each age groups. $\Delta v_i(k)$ is the number of vaccine that will be released with age group A_i in current moment. Thus based on the infection dynamics of current moment k, $\{S_i(k), I_i(k), V_i(k)\}$, and virus infection parameter set $\{\tau_i, \beta_i, \Delta v_i(k), \lambda_i(k)\}$, we can get the infection dynamics in next moment $\{S_i(k+1), I_i(k+1), V_i(k+1)\}$.

For the parameter set $\{\tau_i, \beta_i, \Delta v_i(k), \lambda_i(k)\}$,

- τ_i is the recovery rate for infected individuals in age group A_i.
- β_i represents the infection rate for susceptible individuals in age group A_i.
- $\Delta v_i(k)$ is the amount of vaccine that will be released within the population of age group A_i in current moment k.

- $\lambda_i(k)$ represents the risk of infectious contact both from its own located group A_i and from other age groups.

In our model, we assume that virus infection is a result of mixing contact between susceptible individuals and infectious ones. The infectious contact risk for each susceptible individuals is determined by two factors, (1) the frequency of individual contacts with other individuals within or cross age groups, (2) the probability of the contacted individual is a infected one. We use the infectious contact rate λ_i to represent the average infection risk of susceptible individuals in age group A_i as a result of the contact relationships with infectious individuals both within its own age group and cross other groups.

By the definition of contact matrix C, $C = \{c_{i,j} | i, j \in (1, 2, ..., 6)\}$, for each individuals in group A_i, the average times of contact with individuals in group A_j is $c_{i,j}$. Thus we can get the infectious contact rate λ_i by the following equation,

$$(1-\lambda_i)^{\sum\limits_{j=1}^{6} c_{i,j}} = \prod_{j=1}^{6}(1-\frac{I_j}{P_j})^{c_{ij}} \tag{5}$$

Where the left side of the equation is the probability of not being infected through $\sum\limits_{j=1}^{6} c_{i,j}$ times of within or cross group contact with the average infectious contact rate λ_i. The right side of the equation is the probability of not being infected through the combination of $c_{i,j}$ times of contact with individuals in age group A_j, in which the infectious contact rate is $\frac{I_j}{P_j}$ for each age group A_j. We can get the average infectious contact rate $\lambda_i(k)$ as,

$$\lambda_i(k) = 1 - \left[\prod_{j=1}^{6}(1-\frac{I_j(k)}{P_j(k)})^{c_{ij}}\right]^{\frac{1}{\sum\limits_{j=1}^{6} c_{i,j}}} \tag{6}$$

Based on our SIV virus infection model, we can simulate the epidemic spreading dynamics of current H1N1 virus, in which two kinds of virus infection heterogeneities are been adopted in the model, one is the heterogeneity of population's vulnerability, the other is the heterogeneity of contact relationship within or between age groups. Thus together with this SIV model, we can evaluate the efficacy of vaccine deployment strategies which are designed with different releasing priorities, such as, vaccination based on host vulnerability and vaccination by individuals' transmissibility. In the following section, we will discuss several factors of vaccination deployment that might influence the epidemics spreading dynamics.

3 Vaccination Deployment Strategies

Epidemic interventions can be used to control epidemics spreading dynamics, and suppress or prevent the intense infection outbreaks. Vaccination is one of the most important method to reduce virus infection risk. If an individual is vaccinated, it will immunize or have a lower probability of being infected when exposed to an infectious contact.

The impact of vaccination on epidemic infection dynamics can be evaluated in the two points, the individual's status and the global spreading dynamics.

- For the vaccinated individuals, they will have a lower infection rate when have a contact with their infectious counterparts.

- For the global infection dynamics, the vaccinated subpopulation can reshape the epidemic spreading landscape, which can be designed as the intervention strategies to control the infection dynamics.

A vaccine deployment schedule contains three components: (1) total amounts of vaccine available, (2) vaccine releasing time, (3) vaccine distribution methods. In epidemic infection environment, vaccinated population can be viewed as the part of function losing agglomerate, which means the new infection landscape is the confined within the unvaccinated part. Thus we can use different kind of vaccination deployment strategies to modify the epidemic infection landscape, aiming at suppressing the epidemic infection severity.

- **Amounts of Vaccine Available**
  In our study, the amount of vaccine available means the total number of vaccine that will be released in the whole process of vaccination deployment schedule. This factor is directly determine the percentage of total vaccinated population.

- **Releasing Time**
  The factor of releasing time means when the first batch of vaccine can be adopted and how long it will be lasting. The study of vaccination released in different time aims at observing the efficacy of vaccine deployment under different virus infection severity.

- **Distribution Methods**
  Vaccine distribution method means the vaccination priority of each population groups during vaccine deployment procedure. Such as, population with higher contact frequency should be first vaccinated with concern of suppressing the virus spreading speed.

Yet the impacts of these vaccine deployment factors still remain to be systematically understood. Because of the insufficient information about an unknown infectious disease, it becomes impossible to prepare a vaccine deployment in advance. In this regard, we believe it is important as well as practically desirable to find a reasonable vaccine deployment criteria before making a effective epidemic intervention schedule. This should draw on a deep understanding of relative impacts of vaccine deployment factors on disease infection dynamics.

In the following sector, we will observe the impact of the three deployment factors with different kinds of vaccination parameter settings.

4 Simulation and Results

In this section, we use SIV model with difference equations to simulate H1N1 swine flu infection dynamics. The age groups are parameterized with the demographical statistics of United Kingdom in 2007 [16]. The contact matrix between age groups are derived from Mossong's results of contact activities in European countries [11]. The value of virus infection parameters are adopted by current H1N1 virus infection study [8].

4.1 Epidemic Dynamics with Nonvaccination

Figure 1 shows epidemic curve for 2009 H1N1 influenza pandemic under SIV model with no vaccine released. It is clear that the development of the virus infection spreading can be separated into three stages in terms of the rate of newly-increased infections. They are: incipient infection stage, infection mass spreading stage, and infection stable stage.

- **Incipient infection stage.** In this stage, infection cases increase slightly and the total number of infection is less than 1% of the whole population (Figure 1(a)). In this stage the probability of infectious contacts with infected individuals both within and cross age groups is near zero (Figure 1(c)), which means the infected cases are mainly confined in their initial groups (Figure 1(b)), and the cross group infection is scare.

- **Infection mass spreading stage.** In mass infection stage, the most obvious feature is the total number of newly infected population increase sharply (Figure 1(a)). Meanwhile the probability of infectious contacts also begin shooting up (Figure 1(c)), which is a sign that infection will spread out among each groups through their contacts activities. The cross group infectious contacts work as the positive feedback mechanism to create mass infections in all of the age groups.

- **Infection stable stage.** In this stage, the increase of total infections will be stagnant; however the total number of infections is very high. This stage is an equilibrium state for epidemic spreading, which is balanced by the recovery from infection and newly infections of contact activities.

4.2 Epidemic Dynamics with Vaccination

As discussed above, three factors will impact the efficacy of vaccination deployment, including (1) the amount of total vaccine available, (2) vaccine releasing time, (3) vaccine distribution methods. In this section, we will observe the relative influence of these three deployment factors by simulating epidemics spreading dynamics adopted with different settings of vaccination deployment.

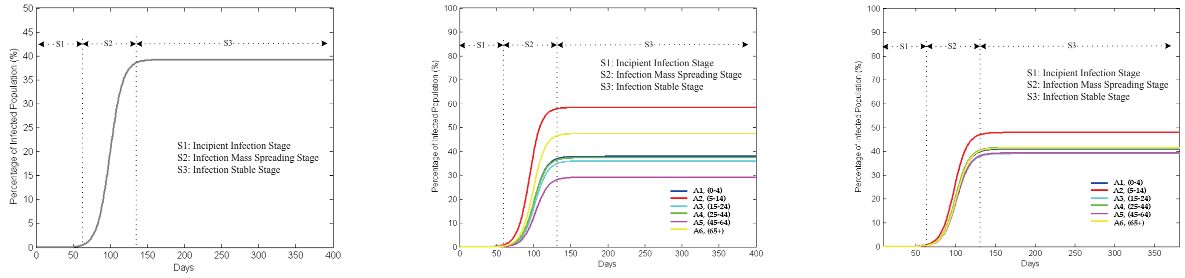
4.2.1 Vaccination Settings

In our simulations, total vaccine availability is divided into three levels, (1)low quantity (5million, 8% of total population), (2) middle quantity (10million, 16% of total population), (3) ample quantity (20million, 32% of total population). Vaccine releasing time has four choices, (1) pre-epidemic spreading ($T_1 = 0$ day), (2) incipient infection stage ($T_2 = 50$ day), (3) infection mass spreading stage ($T_3 = 100$ day), (4) infection stable stage ($T_4 = 150$ day). The vaccine will be distributed following three methods, (1) vaccination by random, in which the number of vaccine released to each age group is totally random; (2) vaccination by transmissibility, which means that two of the age groups with the highest contact frequency will be vaccinated with prime priority; (3) vaccination by vulnerability, which assign the high vaccination priority to two groups of population with higher infection rate.

Given the practical limitation of vaccine deployment, the vaccine releasing sequence is designed as followings: (1) in each time unit (day), 1 million vaccine will be released to susceptible individuals; (2) the amount of vaccine adopted to each age groups is determined by their vaccination priorities, which is defined in vaccination distribution methods.

4.2.2 Impact of Vaccination Deployment Factors

For epidemic intervention by releasing vaccine to susceptible population, the following three concerns determine the efficacy of infection dynamics control: (1) how many vaccine are available; (2) when the vaccine can be started to release; (3) how to distribute vaccine to individuals with different priority. To schedule an effective vaccination deployment plan, first, we should analysis the relative impact of three factors on each stage of epidemics spreading

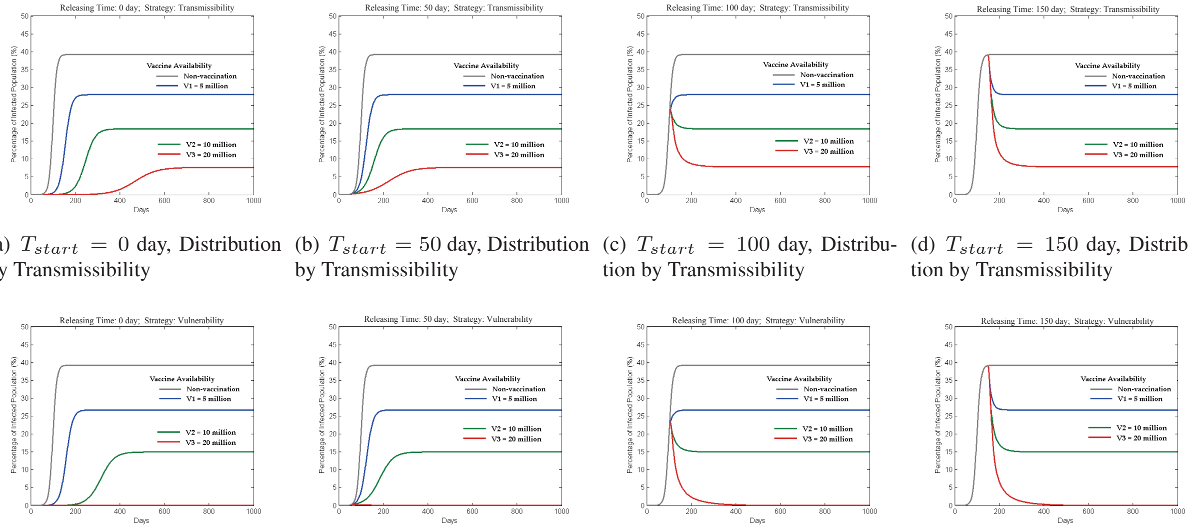


(a) Infection Dynamics of the Whole Population with Nonvaccination

(b) Infection Dynamics of Age Groups with Nonvaccination

(c) Probability of Infectious Contact within and cross Age Groups

Figure 1: Three Stages in Disease Infection Dynamics with Nonvaccination. *S1: incipient infectious stage, S2: mass infection stage, S3: stable infected stage. In incipient infectious stage, infections increase slightly and the total number of infection is relatively low compared with the whole population. In this stage infected cases are confined in initial groups. In mass infection stage, the most obvious feature is the rate of newly infected cases increase sharply. The infection will spread out among each group through their social contacts. In stable infected stage, the increase of total infections will be stagnant; however the total number of infections is very high. This stage is an equilibrium state for epidemic spreading.*



(a) T_{start} = 0 day, Distribution by Transmissibility

(b) T_{start} = 50 day, Distribution by Transmissibility

(c) T_{start} = 100 day, Distribution by Transmissibility

(d) T_{start} = 150 day, Distribution by Transmissibility

(e) T_{start} = 0 day, Distribution by Vulnerability

(f) T_{start} = 50 day, Distribution by Vulnerability

(g) T_{start} = 100 day, Distribution by Vulnerability

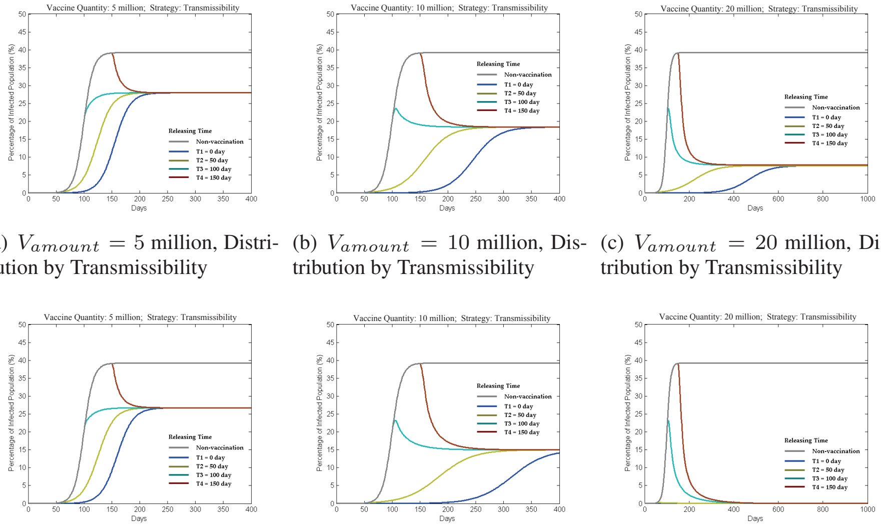
(h) T_{start} = 150 day, Distribution by Vulnerability

Figure 2: Impact of Vaccine Availability on Infection Dynamics. *Grey: contrast curve of non-vaccination; Blue: 5 million vaccine availability (8% of total population); Green: 10 million vaccine availability (16%); Red: 20 million vaccine availability (32%). The impact of vaccine availability is reflected at the percentage of infected population in infection stable stage. The simulation results show that as the total amount of available vaccine increased from 5 million (8% of total population) to 10 million (16%) and finally to 20 million (32%), the percentage of stable infected population would decrease from 28% to 18.4% and finally to 7.6% (vaccine distributed by transmissibility as shown in Figure a - d and from 26.7% to 15% and finally to infection decayed (vaccine distributed by vulnerability as shown in Figure e - f. The more vaccine released to population, the less infected population in the stable stage of epidemic spreading process.*

dynamics.

As we has discussed in the previous section, the epidemic spreading dynamics can be separated into three stage concerned with the newly increased infections.

- **Infection Stage 1: Incipient Infection Stage**
  In the first stage, incipient infection stage, the absolute number of infection are relatively low and the infection spreading are mainly confined within the initial infected population groups. Thus the infection



(a) V_{amount} = 5 million, Distribution by Transmissibility (b) V_{amount} = 10 million, Distribution by Transmissibility (c) V_{amount} = 20 million, Distribution by Transmissibility

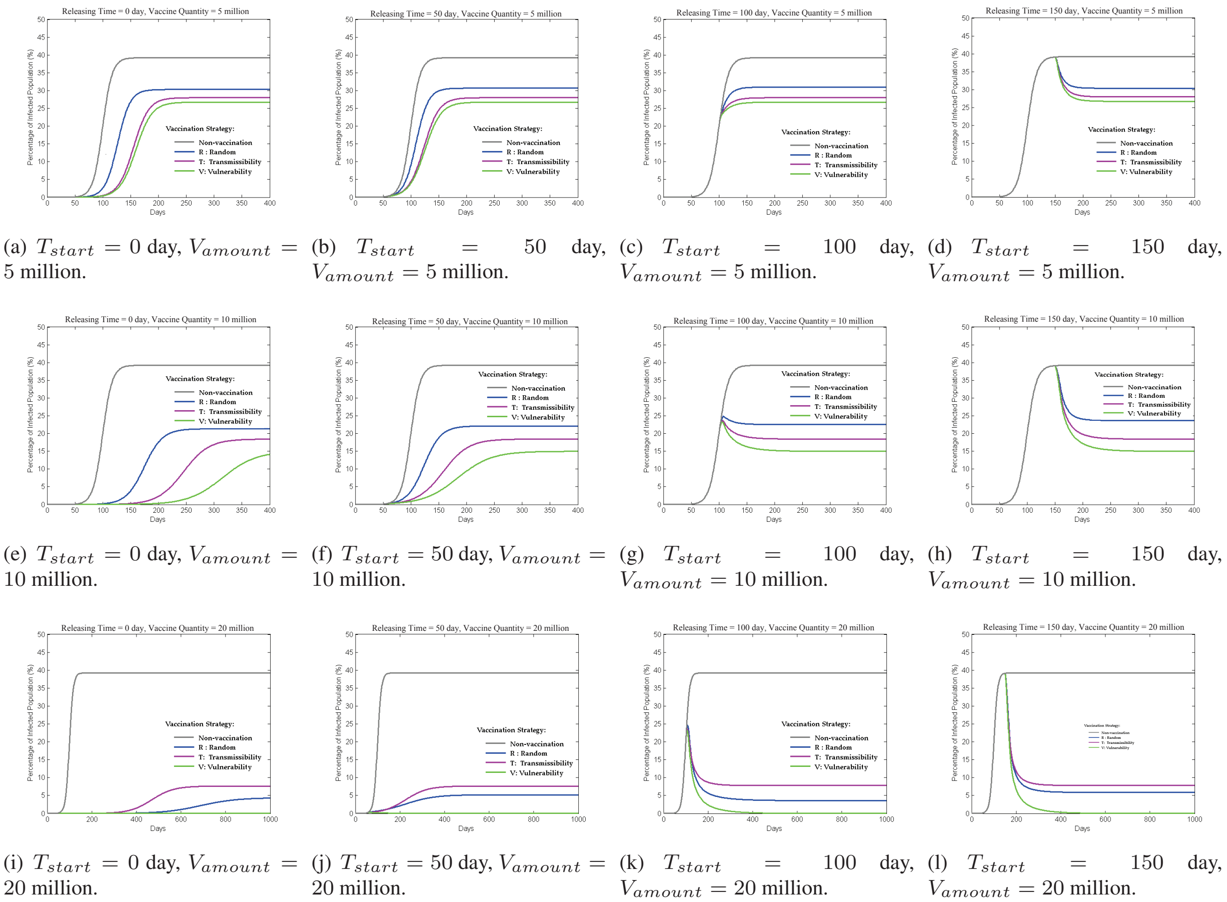
(d) V_{amount} = 5 million, Distribution by Vulnerability (e) V_{amount} = 10 million, Distribution by Vulnerability (f) V_{amount} = 20 million, Distribution by Vulnerability

Figure 3: Impact of Vaccine Releasing Time on Infection Dynamics. *Grey:contrast curve of non-vaccination; Blue: vaccination at pre-spreading of epidemics (0 day); Yellow: vaccination at incipient infection stage (50 day); Cyan: vaccination at infection mass spreading stage (100 day); Magenta Red:vaccinaiton at infection stable stage (150 day). The impact of vaccine releasing time mainly appears at infection mass spreading stage. In the simulation results, topping time of phase transition from stage 1 to stage 2 have been delayed and the rising time of infection curve have been prolonged, as shown in blue curve (pre-spreading vaccination) and yellow curve (incipient infection vaccination). The cyan curve (vaccination at mass spreading stage) and magenta red curve show that the peak value of infection will be reduced and the convergence of infection stable stage will be accelerated.*

risk imposed by cross group contact is less than 1%, shown in Figure 1(c). However the individuals with a higher infection rate are more likely been infected. Our simulation results show that if the vaccine are released in this stage and distributed by individuals' vulnerability, the lasting time of the incipient infection stage can be prolonged, as shown in Figure 4(a),4(e), 4(i).

- **Infection Stage 2: Infection Mass Spreading Stage** In the second infection stage, infection mass spreading stage, the number of infection cases in each subpopulation groups increase sharply, which is the result of positive feedback effect between rate of infectious contact and infected population. As the accumulation of infected population in each age groups, the susceptible individuals will face with an increased probability of cross group infection risk if the contact frequency with outside is higher. Our simulation results show that the vaccination strategy based on individuals' vulnerability can suppress the speed of infection spreadings, as shown in Figure 4(b),4(f), 4(j).

- **Infection Stage 3: Infection Stable Stage** In the last stage, infection stable stage, this stage is an equilibrium state for epidemic spreading, which is balanced by the recovery from the infected population and newly infections through infectious contacts. There two factors influence the results of this stage: (1) the total amount of vaccine availability, (2) how vaccine distributed among each population groups. The simulation results show that as the total amount of available vaccine increased from 5 million (8% of total population) to 10 million (16%) and finally to 20 million (32%), the percentage of stable infected population would decrease from 28% to 18.4% and finally to 7.6% (vaccine distributed by transmissibility as shown in Figure 2(a) - 2(d) and from 26.7% to 15% and finally to infection decayed (vaccine distributed by vulnerability as shown in Figure 2(e) - 2(h). The more vaccine released to population, the less infected population in the stable stage of epidemic spreading process. The distribution of vaccine reshape the epidemics spreading landscape. The vaccination method based on individuals' vulnerability can decrease the population with high infection rate and vaccination method based on individuals' transmissibility can decrease the number of individuals with a higher infectious contact activities. Thus both of these two vaccine distribution method can significantly lower the percentage of in-



(a) $T_{start} = 0$ day, $V_{amount} =$ 5 million.

(b) $T_{start} = 50$ day, $V_{amount} = 5$ million.

(c) $T_{start} = 100$ day, $V_{amount} = 5$ million.

(d) $T_{start} = 150$ day, $V_{amount} = 5$ million.

(e) $T_{start} = 0$ day, $V_{amount} =$ 10 million.

(f) $T_{start} = 50$ day, $V_{amount} =$ 10 million.

(g) $T_{start} = 100$ day, $V_{amount} = 10$ million.

(h) $T_{start} = 150$ day, $V_{amount} = 10$ million.

(i) $T_{start} = 0$ day, $V_{amount} =$ 20 million.

(j) $T_{start} = 50$ day, $V_{amount} =$ 20 million.

(k) $T_{start} = 100$ day, $V_{amount} = 20$ million.

(l) $T_{start} = 150$ day, $V_{amount} = 20$ million.

Figure 4: Impact of Vaccine Distribution Methods on Infection Dynamics. *Grey: contrast curve of non-vaccination; Blue: vaccination by random distribution; Purple: vaccination by transmissibility; Green: vaccination by vulnerability. The impact of vaccine distribution methods can be observed at infection mass spreading stage and at final infection stable stage. The results of vaccination simulations with three distribution methods, vaccination by random (blue curve), vaccination by transmissibility (purple curve) and vaccination by vulnerability (green curve), show that the percentage of final infected population in stable stage is lower, topping time of stage transition to mass infection spreading have been delayed and the rising time of stable infection stage have been prolonged respectively.*

fection in the final stable infection stage, as shown in Figure 4.

5 Conclusions

We develop a mathematical model to simulate H1N1 epidemic dynamics, which is adopted with heterogeneities individuals' vulnerability and transmissibility based on populations' age structure. In our study, we propose that the infection risk is the combination of two driven forces, one is the probability of being engaged in an infectious contact, the other one is the probability of being successful infected when exposed in a infectious contact. In our study we found that the leading force of the two are shifted from one to another during the development of epidemic spreading process.

We also focus on our analysis on the impact of deployment factors on the efficacy of vaccination distribution, the factors we concerned include total amount of vaccine, vaccine releasing time and distribution methods. By adopting different vaccination deployment settings, we found that increasing vaccine available doses can lower the percentage of final infected cases in the stable infected stage. The releasing of vaccines earlier in each infection stages can improve the dynamical process of infection dynamics by both suppressing the overshoot and accelerating the convergence of infection dynamics. Our results have also shown that, if the grouped population with a higher infection force, which characterizes the dynamical infectious contribution based on their current infection dynamics and social contact patterns, is vaccinated with a priority, both the dynamical infection process and the final stable infection state can be improved.

Our study does not consider the variability of contact pattern, such as, individuals will reduce their contact frequency with outsider as the response of epidemic outbreak. We also ignore other aspects of demographical features that might influence the infection dynamics, such as, gender, occupation and ethics. These factors will be discuss in the future works. Our current work highlights the impact of three vaccine deployment factors on each stage of epidemic infection process and based on these observations efficient vaccination deployment plan can be made to control epidemic spreading.

References

[1] US Centers for Disease Control and Prevention (Accessed 20 October, 2009) CDC joint briefing with NIH and FDA on 2009 H1N1 Influenza, August 21, 2009, 12:00pm. http://www.cdc.gov/ media/transcripts/2009/t090821.htm.

[2] US Centers for Disease Control and Prevention (Accessed 20 October,2009) 2009 H1N1 Influenza Vaccine Supply Status. http://www.cdc.gov/h1n1flu/vaccination/updates/101609.htm.

[3] World now at the start of 2009 influenza pandemic. http://www.who.int/mediacentre/news/statements/2009/h1n1_pandemic_phase6_20090611/en/.

[4] S. Cauchemez, A.-J. Valleron, P.-Y. Boelle, A. Flahault, and N. M. Ferguson1. Estimating the impact of school closure on influenza transmission from sentinel data. *Nature*, (452):750–754, April 2008.

[5] D. N. Fisman, R. Savage, J. Gubbay, C. Achonu, H. Akwar, D. J. Farrell, N. S. Crowcroft, and P. Jackson. Older age and a reduced likelihood of 2009 h1n1 virus infection. *The New England Journal of Medicine*, 36(20):2000–2001, November 2009.

[6] L. Manzoli, F. Schioppa, A. Boccia, and P. Villari. The efficacy of influenza vaccine for healthy children: A meta-analysis evaluating potential sources of variation in efficacy estimates including study quality. *The Pediatric Infectious Disease Journal*, 26(2):97–106, February 2007.

[7] J. Medlock and A. P. Galvani. Optimizing influenza vaccine distribution. *Science*, 325(5948):1705 – 1708, September 2009.

[8] E. Miller, K. Hoschler, P. Hardelid, E. Stanford, N. Andrews, and M. Zambon. Incidence of 2009 pandemic influenza a h1n1 infection in england: a cross-sectional serological study. *The Lancet*, Early Online Publication.

[9] M. A. Miller, C. Viboud, M. Balinska, and L. Simonsen. The signature features of influenza pandemics ł implications for policy. *The New England Journal of Medicine*, 360(25), 2009.

[10] G. J. Milne, J. K. Kelso, H. A. Kelly, S. T. Huband, and J. McVernon. A small community model for the transmission of infectious diseases: Comparison of school closure as an intervention in individual-based models of an influenza pandemic. *PloS One*, 3(12):1–100, December 2008.

[11] J. Mossong, N. Hens, M. Jit, P. Beutels, K. Auranen, R. Mikolajczyk, M. Massari, S. Salmaso, G. S. Tomba, J. Wallinga, J. Heijne, M. Sadkowska-Todys, M. Rosinska, and W. J. Edmunds. Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLoS Medicine*, 5(3), March 2008.

[12] NACI. Statement on influenza vaccination for the 2008-2009 season. *Can Commun Dis Rep*, 34:1–46, July 2008.

[13] E. Negri, C. Colombo, L. Giordano, N. Groth, G. Apolone, and C. L. Vecchia. The efficacy of influenza vaccine for healthy children: a meta-analysis evaluating potential sources of variation in efficacy estimates including study quality. *Vaccine*, 23:2851–2861, 2005.

[14] K. L. Nichol, J. D. Nordin, D. B. Nelson, J. P. Mullooly, and E. Hak. Effectiveness of influenza vaccine in the community-dwelling elderly. *The New England Journal of Medicine*, 357(14):1373–1381, October 2007.

[15] L. Simonsen, R. J. Taylor, C. Viboud, M. A. Miller, and L. A. Jackson. Mortality benefits of influenza vaccination in elderly people: an ongoing controversy. *The Lancet infectious diseases*, 7(10):658–666, 2007.

[16] C. Wroth and A. Wiles. Key population and vital statistics. Technical report, Office for National Statistics, 2007.

Autonomy-Oriented Mechanisms for Efficient Energy Distribution

Benyun Shi

Abstract

Due to the uneven geographical availability of energy resources and the world imbalanced economic development, it is essential for the energy suppliers and consumers in different countries or regions to most efficiently, economically, as well as reliably distribute energy resources. The general problem of energy distribution is a complex one in that many factors can be involved either endogenously or exogenously such as human activities, energy transportation efficiency, geopolitics, and so on. Traditional statistical and/or centralized models are impractical to tackle energy distribution problems that are, by nature, non-centralized and/or dynamically evolving. Although some decentralized approaches (e.g., multi-agent systems) may be adopted to solve certain types of dynamic distribution problems in a small scale, they are, at the moment, still quite limited in methodology and applications to practically address various important issues as related to the problems. In this paper, starting from a specific energy distribution problem, we present a decentralized behavior-based paradigm that draws on the methodology of self-organized computing, i.e., autonomy-oriented computing. The goal of our work is twofold: (i) to characterize the underlying mechanism of the energy distribution system, and (ii) to provide scalable solutions for efficient energy distribution. We provide simulation-based experiments to show the performances of four local behavior-based algorithms, with gradually increased behavioral complexity. The simulation results show that global objectives can be approximately reached through autonomous entities with even simple exploration and regulation behaviors. We conjecture that efficient energy trading markets can emerge from appropriate behavior-based mechanisms, which can autonomously improve energy distribution efficiency.

1 Introduction

Nowadays, the daily life of human being on the earth becomes more and more heavily depending on different kinds of energy resources (It is reported by International Energy Agency (IEA) in [14] that the total primary energy supply of the world has doubled at the end of 2007 comparing with the year 1971.). British Petroleum in [2] reports that the reserves of fossil fuels on the earth will not afford the requirement of human economic development in the very nearly future, for example, the reserves-to-production ratio of world oil (respectively, natural gas, coal) is estimated at 42 years (respectively, 60 years, 122 years), at the end 2008. In addition to the scarcity of energy resources, we are also facing another serious problem, i.e., the uneven geographical availability of energy resources and the world imbalanced energy demand. On one hand, according to the *Oil and Gas Journal* [1] and IEA [13], 56 percent of the world's proved oil reserves are located in the Middle East, and almost three-quarters of the world's natural gas reserves are located in the Middle East and Eurasia. On the other hand, North America and Europe contribute almost half the the world primary energy consumptions in 2008 [2]. Furthermore, it is predicted by IEA in [13] that, energy demand may grow rapidly in less developed regions (e.g., China, India) for the rapid development of these regions in recent years. For all above reasons, it becomes essentially important for the energy suppliers and consumers in different regions to distribute energy resources to meet their different requirements.

1.1 The Energy Distribution Problems

Distributing energy resources among energy suppliers and consumers at different regions relates to various issues in different areas (e.g., economy, geopolitics, logistics, systems, market, etc.). For example, the energy price issues; the transportation infrastructure (e.g., railway, oil or natural gas pipelines, etc.) investment issues for industries and/or governments [33]; the inventory management issues to meet short-term needs when supply of energy resources are interrupted for any reason (e.g., terrorism, severe weather events, etc.); the cascading control or congestion management issues on power grid [26][21]; and the issues about energy trading markets such as the world oil markets and oil futures markets. All these issues interrelate to form a very complex energy distribution system, which may have its own reliability [32], vulnerability [3], and system security [33]. For the reason that most of the distribution challenges arise from the fact that energy suppliers and consumers are located in different geographic regions, in the context of this paper, we mainly focus on the logistics networks of energy resources, i.e., the energy distribution management problems.

The logistics network required to supply energy resources from energy supplier to energy consumers is an integration of different distribution infrastructures (e.g., pipelines, ships, railways, etc.). Most existing studies (e.g., [20][5][24][25][26], just list a few) focus on the distribution managements under the physical constraints of the existing logistics networks. In this paper, inspired by the dynamics-driven network optimization problems proposed in [22], and biologically-inspired adaptive network proposed in [31], we will focus the formation of robust and adaptive energy flow networks or trading relationship networks in terms of distribution efficiency (e.g., minimal transportation cost).

1.2 Challenges

The reality of the energy distribution problems are more complex in terms of (i) the energy supply and/or demand may dynamically change either endogenously or exogenously, (ii) the coupling relationships between energy suppliers and consumers may not be explicitly represented by simple (linear) functions, (iii) the information may only be partially available due to private issues or competitions among energy suppliers and consumers, and (iv) in real market, entities (either energy suppliers or consumers) make decisions (e.g., where to import/export energy, how many to import, etc.) based on their own benefits rather than the global goal (e.g., to minimize the total cost of energy flows in [24][25]) of the distribution systems. Therefore, an energy distribution system can be considered to be a complex one, which is, by nature, open, highly distributed, and dynamically evolving. In this case, it is difficult for statistical models (e.g., [15][29], just list a few) and/or centralized optimization approaches (e.g., [7][24][25][20], just list a few) to solve such open, dynamic energy distribution problems.

Although some decentralized approaches (e.g., multi-agent systems [17][9]), which have been proposed to solve resource management (e.g., resource allocation [8]) problems, may also be adopted to solve certain type of dynamic energy distribution problems in a small scale, they are, at the moment, still quite limited in methodology and applications to practically address various important issues as related to the energy distribution problems, e.g., the natural mechanisms underlying an open, unpredictable energy distribution system. Most of these approaches focus on strategy design (e.g., multi-agent negotiation systems [18][16]), which belong to the problem of dynamics optimization on static networks in [22]. As far as we know, very few studies have been done to characterize the underlying mechanism of the energy distribution systems in terms of dynamics-driven network optimization [22] and/or adaptive network formation [31].

1.3 Our Considerations

In order to efficiently, economically, as well as reliably distribute energy resources in the open, dynamic environments, it is necessary for us to understand the underlying mechanisms of the distribution systems. In this paper, we present a decentralized behavior-based paradigm that draws on the methodology of self-organized computing, i.e., autonomy-oriented computing (AOC) [19]. According to AOC, the entities can spontaneously interact with each other as well as their environments, and operate based on their behavioral rules. The relationships between entities can therefore be self-organized through entities' behavioral dynamics. Global objectives can be effectively and efficiently achieved by involving positive-feedback mechanisms and collective regulation. The goal of our work is twofold: (i) to characterizing the underlying mechanisms of the energy distribution system through local interactions between energy supplier and consumers with different kinds of behavioral rules, and (ii) to provide scalable solutions for efficient energy distribution.

Before we move to study the more complex energy distribution problems, in this paper, we mainly focus on evaluating the performances of the local behavior-based paradigm for a static energy distribution problem in the first instance. The basic goals of the static energy distribution problem in this paper are (i) to distribute all energy resources from energy suppliers to energy consumers, and (ii) to minimize the total energy distribution costs. By evaluating performances of different kinds of local behavior-based algorithms, we try to answer the following questions.

- How does an optimal (i.e., minimizing the total energy distribution cost) energy flow network can emerge through local dynamic of supplier/consumer entities?
- What kind of local behaviors of supplier/consumer entities are crucial for achieving final optimal energy flow network?

The main objectives of this paper are not only to solve the energy distribution problem, but to present a natural behavior-based paradigm with respect to the energy distribution problem so that more complex energy distribution problems (i.e., open and dynamically evolving) can be studied in the future. The behavior-based paradigm may help to answer the following systematic questions of a complex energy distribution system.

- How does the energy flow network evolve in an open, unpredictable energy distribution system?

- What kind of local dynamics between supplies and consumers can improve the robustness and stability of the energy distribution system?

- What kind of energy trading mechanism (market) can be formed? What are the critical factors for the stability of the market?

The rest of this paper is organized as follows. In Section 2, we summarize related energy system models, and show their limitations in terms of the energy distribution problems discussed in this paper. In Section 3, we formulate the energy distribution problem in details. In Section 4, we present a decentralized behavior-based paradigm for the energy distribution problem, and four local behavior-based algorithms with gradually increased behavior complexity to study and solve the problem. We simulate our approaches in Section 5. Finally, we conclude our work and present some future works in Section 6.

2 Related Work

We classify the existing studies on energy systems modeling into two categories: macro- and micro-modeling. Macro-modeling aims to perform predictions (e.g., energy supply/consumption in the future) or scenarios analysis at a macroscopic level, while micro-modeling focuses on solving energy problems at specific energy domains (e.g., power grid, natural gas pipeline system). In this section, we will highlight some of the representative studies of the two categories.

2.1 Macro-modeling of Energy Systems

Traditional energy models (e.g., WORLD [12], COAL [23], FOSSIL [4], etc.) commonly use system dynamics approach, which deals with internal feedback loops and time delays that affect the behavior of an entire system through various interrelated components. System dynamics modeling has been used for strategic energy planning and policy analysis for more than three decades. The main issues the system dynamics energy models try to address include (i) understanding relationships among different components in an energy system [12][23] (e.g., the relationship between proven reserves and cumulative production in an energy discovery system), (ii) capturing the roles that an energy system plays in social, economic, and environmental systems [4][30][11], and (iii) integrating each related systems together to simulate the real world [15][29]. Most of these models are based on statistical data of population, economic growth rate, elasticity of energy substitution, and so on. Therefore, they are well suited for performing predictions or scenarios analysis at a macroscopic level.

Many in-depth work has been done to try to simulate the real world more precisely in recent years (e.g., the world energy model [15] proposed by IEA, the MIT Integrated Global System Model (IGSM) [29])). However, we are still facing the following challenges:

- To precisely simulate the real world requires significant advances in economics, the social science, and environmental science, each of which is quite complicated discipline. It is quite challenging to represent their relationships based only on statistical data.

- In reality, the energy systems are dynamically evolving (e.g., energy technology innovation). It is difficult to represent or predict this kind of dynamics.

- For such integrated systems, as reported in [28], the predictions at a global scale are considered reasonably reliable, while more work should be done to improve predictive capability at regional (i.e., microscopic) scale.

- Such simulations need exascale computing [28].

Limited by above mentioned difficulties, it is difficult for the system dynamics models to provide global/regional energy distribution solutions at specific energy domains, such as natural gas dispatch problem [20], congestion or bottlenecks management in power grid [26][6][7], cascade control [21] in power grid, etc. In the next section, we will introduce micro-modeling of energy systems which can be adopted to solve specific energy distribution problems.

2.2 Micro-modeling of Energy Systems

Micro-modeling of energy systems focuses on developing technological solutions to some specific energy problems of interest. For the energy distribution problems, most existing work are based on optimization approaches, which often combine with other techniques such as network flows models [24][25][20].

The network optimization models, which take into account the effect of spatial constraints (i.e., uneven geographical availability of energy resources), try to find optimal energy flows in a specific network [20][5][24][25]. The networks can represent either the physical energy distribution networks (i.e., the natural gas pipeline network, power grid, etc.) or the trading relationships between energy suppliers and consumers. For example, in [24][25], the authors describe U.S. integrated energy system as a network with collection of nodes and links, where energy resources may flow from one node to another under the constraints of transportation capacity and per unit energy distribution cost on each link. A constrained mathematical optimization approach is proposed to minimize the total cost of energy flows of the network. Different from the work of A. Quelhas et. al. [24][25], where energy flow in each link is independent with flows in other links, authors in [20] presents the economic dispatch in natural gas networks, where the gas flow from a node (i.e., inlet node) to another node (i.e., outlet node) in the networks is determined by the pressure at the inlet node and pressure at the outlet node. In this case, the potential flow in each pipeline is also dependent on the actual flows in other pipelines of the system (i.e., system effects in [20]). However, most existing network flow models, which are designed for optimization purposes, are still centralized approaches.

The idea of abstracting energy components into networks provides a more extensive research potential on energy system modeling. Although some decentralized approaches (e.g., [27][10]) as well as multi-agent systems (e.g., [9][8] [17]), which have been proposed to solve resource allocation problems, may also be adopted to certain dynamic energy distribution problems in a small scale. Most of these approaches focusing on strategy design (e.g., multi-agent negotiation systems [18][16]), belong to the problem of dynamics optimization on static networks in [22]. Except for the problem of dynamics optimization on static networks, the authors in [22] have also proposed the dynamics-driven network optimization problems, which include two types of dynamics. On one hand, the network structure may evolve over time to fit the dynamics (e.g., energy distribution) on the network. On the other hand, the dynamics on the network may inversely be affected by the network structure. Additionally, Tero et al. in [31] have proposed biologically-inspired approach to form adaptive networks with comparable efficiency, fault tolerance, and cost to real-world infrastructure networks (i.e., the Tokyo rail system in [31]). However, as far as we know, very few studies have been done to characterize the underlying mechanism of the energy distribution systems in terms of dynamics-driven network optimization [22] and/or adaptive network formation [31].

3 Problem Statements

As presented in Section 1, energy distribution networks (e.g., railway networks, natural gas pipeline networks) are essential for allocating energy resources under an open, dynamically evolving energy distribution system. A robust and adaptive energy distribution network plays important roles in distribution efficiency (e.g., minimizing transportation cost), and fault tolerance (e.g., transportation dysfunction during abnormal weather). In this case, designing mechanisms without centralized control [31] to form adaptive energy distribution network becomes quite significant for energy distribution management. In this paper, we first evaluate the performances of the local behavior-based paradigm (described in Section 4) for a static energy distribution problem so that the paradigm can be extended to more more complex energy distribution problems in the future.

Consider a set of n energy suppliers/consumers, we want to distribute energy resources from energy suppliers to energy consumers. As described in Section 1.1, distributing energy resources from one region to another may take various costs (e.g., capital cost associated with constructing pipelines, energy resources consumed or wasted during distribution, etc.). We abstract distribution costs among energy suppliers and consumers to be a a predefined cost matrix $CMatrix_{n\times n} = \{c_{ij} | 1 \leq i,j \leq n\}$, where c_{ij} represents the per unit energy distribution cost from one supplier/consumer i to another supplier/consumer j. In this paper, we suppose $CMatrix_{n\times n}$ is symmetric, which means that $c_{ij} = c_{ji}$, for $1 \leq i,j \leq n$. However, triangle inequality may not be correct, i.e., $c_{ij} + c_{jk}$ may not definitely greater than or equal to c_{ik}.

Definition 1. ***Energy Distribution Network*** *The predefined cost matrix $CMatrix_{n\times n}$ forms a fully-connected energy distribution network, where each node represents an energy supplier/consumer, and each link is associated with the per unit energy distribution cost.*

Suppose that initially the total energy supply of suppliers equal to the total demand of consumers, in this paper, we will study how an efficient energy flow network can emerge from the energy distribution network through local dynamics of energy suppliers/consumers.

Definition 2. ***Energy Flow Network*** *The energy flows among energy suppliers/consumers can be represented by an undirect network $G =< V, L, Q >$. The node set $V = \{v_i | 1 \leq i \leq n\}$ denotes the set of n suppliers/consumers. The link set $L = \{l_{ij} | 1 \leq i,j \leq n\}$ represents all existing energy flows (if there are energy flows between node v_i and v_j, then $l_{ij} = l_{ji} = 1$; otherwise, $l_{ij} = l_{ji} = 0$). The quantity set $Q = \{q_{ij} | 1 \leq i,j \leq n\}$ represents the volume of energy flows on each link l_{ij}.*

To evaluate the performances of different local behavior-based strategies proposed in this paper, we have two kinds of measurements for the final energy flow networks, i.e., the global cost and per unit cost of final energy flow networks.

Definition 3. ***Global Cost of Energy Flow Network*** *The global cost of energy flow network represents total distribution cost of allocating all energy supply to corresponding consumers based on the energy flow network. In this case, the total costs of all energy flows can be calculated by*

$$TC = \sum_{l_{ij} \in L} q_{ij} \cdot c_{ij} \cdot l_{ij} \tag{1}$$

Definition 4. ***Per Unit Cost of Energy Flow Network*** *The per unit cost of energy flow network represents average distribution cost of certain quantity of energy resources allocating from energy suppliers to corresponding consumers based on the energy flow network. By this definition, the per unit cost of energy flows can be calculated by*

$$PC = \frac{TC}{\sum_{l_{ij} \in L} q_{ij}} \tag{2}$$

Specific research issues to be studied include (i) *distribution rate* (i.e., can the local behavior-based strategies distribute all energy supply to consumers?), (ii) *distribution cost* (i.e., can the global cost or per unit cost of energy flow network generated by the local behavior-based strategies approach to that of the optimal solution?), and (iii) *scalability*(i.e., when the number of entities increases, can the performance of local behavior-based strategies remain efficient, i.e., higher distribution rate and lower distribution cost?). In order to answer the above questions, we will, first of all, present some detailed formulation of the local behavior-based strategies in the next section.

4 Formulations for the Behavior-based Paradigm

In this section, we will present in details the local behavior-based paradigm that draws on the methodology of autonomy-oriented computing (AOC) [19]. According to AOC, entities spontaneously interact with each other as well as their environments based on their behavioral rules to reach certain global objectives. In the context of the energy distribution problem in this paper, the global objectives are (i) to distribute all energy resources from energy suppliers to energy consumers, and (ii) to minimize the global energy distribution costs.

4.1 Entities Profile

In this paper, we have n entities $E = \{e_i | 1 \leq i \leq n\}$, each of which represents either an energy supplier or consumer. A supplier entity aims to sell its energy surplus to appropriate consumers, while a consumer entity aims to buy energy resources from appropriate suppliers to make up its energy deficit. Because the energy distribution costs are finally undertaken by both energy suppliers and consumers, in this paper, we assume that each entity prefers to energy resources with lower distribution costs.

The profile of an entity is represented as a tuple, $< id, type, volume, memory, rules >$, where id denotes the identifier of an entity. $type = \{supplier, consumer\}$ means that an entity may either be an energy supplier or consumer. $volume$ represents the amount of surplus/deficit an energy supplier/consumer has. $memory$ records information that the entity has. Since in real world no supplier/demander has complete information of the energy distribution systems, in this paper, we assume that initially, an entity only has information about its per unit distribution costs to all other suppliers/consumers on the distribution network (i.e., entity e_i only has cost information $\{c_{ij} | 1 \leq j \leq n\}$). The entity does not know other entities' $type$ and $volume$. Therefore, entities need to move on the distribution network and interact with other entities to collect more information. $rules$ determines how an entity move on the contact network and interact with other entities. In this paper, we represent two kinds of rules, i.e., behavioral rules and decision-making rules.

4.2 Behavioral Rules

The behavioral rule of an entity determines how the entity move on the distribution network to collect distribution cost information and find trading partners. In this paper, we assume that once an entity moves to a node on the distribution network, it will get the distribution cost information, as well as the energy surplus/deficit of the supplier/consumer on the node. Then the entity will determine whether or not to trade with the node based on its decision-making rules. The information of visited suppliers/consumers will be saved at the entity's $memory$. For the static energy distribution problem in this paper, entities move on the distribution network based on self-avoiding random walks, which play a central role in the modeling of the topological behavior of thread- and loop-like molecules.

Definition 5. ***Self-avoiding Random Walk*** *A self-avoiding random walk is a sequence of moves on a network that does not visit the same node more than once.*

In this paper, we will present two kinds of self-avoiding random walks to study the effects of entities $memory$ on the global performance of the mechanism. For the first kind of random walk (adopted by Algorithms 1 and 2), each entity only uses cost information of the current visited node to determine where to move for the next step, i.e., the entity does not memorize information. For the second kind of random walk (adopted by Algorithms 3 and 4), each entity will memorize all cost information of nodes that it has already visited, and integrate this information to determine its next step. The hypothesis is that by utilizing $memory$, it is much easier for an entity to find a path with small distribution cost on the static distribution network. The details of the random walks will be introduced in Section 4.4.

Remark: The self-avoiding random walk is adopted in this paper for the static energy distribution problem because the supply/demand of each entity will stay constant during the dynamic process. However, it is obviously unsuitable for the dynamic distribution problems. Here, it is necessary to emphasize that our focus is mainly on the impacts of entities $memory$ (i.e., with limited memory or with unlimited memory) on the performances of the mechanism.

4.3 Decision-making Rules

An entity makes decisions about whether or not to trade with other suppliers/consumers based on its decision-making rule. In this paper, we present three kinds of decision-making rules, i.e., first-come-first-serve, competition, and request-passing, to study the effects of different trading strategies on the global performances. Entities with first-come-first-serve decision-making rule will *passively* trade with entities by their visiting order without considering the distribution cost of energy resources; entities with competition rules prefers to trading with entities with lower distribution cost; entities with the third decision-making rule *proactively* send trading requests to a list of entities who it would like to trade with based on cost information in its memory. In other words, an entity will refuse to trade with entities who are not in its list. The hypothesis is that by proactively regulating trading partners and sending requests based on cost information in $memory$, it is more likely for an entity to find appropriate trading partners than passively trading with visitors. The details of the decision-making rules will be introduced in corresponding algorithms in Section 4.4.

4.4 Behavior-based Algorithms

To evaluate the two behavioral hypotheses, we present four behavior-based algorithms with gradually increased behavioral complexity in this section. Entities in Algorithms 1 and 2 have limited memory and adopt first-come-first-serve decision-making rule. Entities in Algorithms 3 and 4 can memorize all cost information of nodes that they have already visited. Especially, entities in Algorithm 4 can proactively send requests to potential trading partners based on information in $memory$.

Algorithm 1: *Self-avoiding Random Walk with First-come-first-serve*: At each round of this algorithm, each entity with $e_i.volume \neq 0, 1 \leq i \leq n$ (i.e., supply are not distributed or demand are not satisfied) behaves in a random order to find trading partners based on self-avoiding random walk on the distribution network. Denote $e_i.Path(t)$ as the set of entities that entity e_i has already visited up to round t, hence, the potential entities for e_i to visit at round $t+1$ is $PE = E \setminus e_i.Path(t)$. The selection probabilities are inversely proportional to the per unit energy distribution cost from current node to all other possible nodes, i.e., the probability of entity e_i selecting $e_j \in PE$ as trade partner at round $t+1$ is calculated by

$$p_{ij}(t+1) = \frac{\frac{1}{CMatrix(e_i,e_j)}}{\sum_{e_k \in PE} \frac{1}{CMatrix(e_i,e_k)}} \quad (3)$$

The trading agreement will be reached based on a first-come-first-serve rule without considering the costs of energy distribution.

```
   Input: Cost matrix $CMatrix_{n\times n}$; Volume of each entity
          $\{e_i.volume|e_i \in E\}$;
   Output: Energy flow network $RMatrix_{n\times n}$
1  Initialize $e_i.Path(1) \leftarrow e_i$ for all $1 \leq i \leq n$;
2  foreach Round $t = 1 : (n-1)$ do
3      Generate a random operation order $O(t)$ with
       $e_i.volume \neq 0$ for all $1 \leq i \leq n$ ;
4      foreach $e_j \in O(t)$ do
5          $PE = E \setminus e_j.Path(t)$ ;
6          Select $e_k \in PE$ based on Eq. 3;
7          if $e_k.volume \neq 0$ then
8              Update $e_j.volume$ and $e_k.volume$;
9              Update $RMatrix$ based on $e_j.Path(t+1)$;
10         end
11         $e_i.Path(t+1) \leftarrow e_i.Path(t) \cup e_k$;
12     end
13 end
```

Algorithm 1: Self-avoiding Random Walk with First-come-first-serve. In the algorithm, steps 5 and 6 are for the self-avoiding random walk; steps 7-10 are for the first-come-first-serve trading. We use $e_i.Path(t)$ to record the nodes that entity e_i has visited up to round t.

Algorithm 2: *Self-avoiding Random Walk with Competition*: The entities' behavioral rule in this algorithm is the same with the self-avoiding random walk in Algorithm 1. The only difference is that at each round, each entity with non-zero $volume$ will first move to the node it selected. Then, entities who selected the same node will compete for trading with the supplier/consumer at that node.

Algorithm 3: *Self-avoiding Random Walk with Information Sharing*: This algorithm is different with Algorithm 2 in terms of random walk process. In this algorithm, once an entity e_i visited a node j on the distribution network, it will memorize all cost information (i.e., $\{c_{jk}|1 \leq k \leq n\}$) of the visited node j. Then, use information in its memory to determine next step of the random walk: the entity e_i will first calculate the minimum costs $ShortestPathCost(e_i, e_k)$ to all other potential entities $e_k, k \in PE$ based on $e_i.memory$, then generate random walk probabilities based on the calculated minimum costs. The probability of entity e_i visiting $e_j \in PE$ at round $t+1$ is calculated by

$$p_{ij}(t+1) = \frac{\frac{1}{ShortestPathCost(e_i,e_j)}}{\sum_{e_k \in PE} \frac{1}{ShortestPathCost(e_i,e_k)}} \quad (4)$$

```
   Input: Cost matrix $CMatrix_{n\times n}$; Volume of each entity
          $\{e_i.volume|e_i \in E\}$;
   Output: Energy flow network $RMatrix_{n\times n}$
 1 Initialize $e_i.Path(1) \leftarrow e_i$ for all $1 \leq i \leq n$;
 2 foreach Round $t = 1 : (n-1)$ do
 3     foreach $e_j.volume \neq 0$ do
 4         $PE = E \setminus e_j.Path(t)$ ;
 5         Select $e_k \in PE$ based on Eq. 3;
 6         $e_i.Path(t+1) \leftarrow e_i.Path(t) \cup e_k$;
 7     end
 8     Generate a random operation order $O(t)$ with
       $e_i.volume \neq 0$ for all $1 \leq i \leq n$ ;
 9     foreach $e_l \in O(t)$ do
10         $V(e_l) = \{e_j|e_j.Path(t+1) = e_l\}$;
11         $S(e_l) = sort(V(e_l))$;
12         foreach $e_m \in S(e_l)$ do
13             if $e_l.volume \neq 0$ then
14                 Update $e_l.volume$ and $e_m.volume$;
15                 Update $RMatrix$ based on
                   $e_m.Path(t+1)$;
16             end
17         end
18     end
19 end
```

Algorithm 2: Self-avoiding Random Walk with Competition. In the algorithm, steps 4-5 are for the self-avoiding random walk; steps 8-18 are for the competition trading. We use $e_i.Path(t)$ to record the nodes that entity e_i has visited up to round t.

```
   Input: Cost matrix $CMatrix_{n\times n}$; Volume of each entity
          $\{e_i.volume|e_i \in E\}$
   Output: Energy flow network $RMatrix_{n\times n}$
 1 foreach $e_i, e_j \in E$ do
 2     $e_i.Path(1) \leftarrow e_i$;
 3     $ShortestPath(e_i, e_j, :) \leftarrow e_i$;
 4 end
 5 foreach Round $t = 1 : (n-1)$ do
 6     foreach $e_j.volume \neq 0$ do
 7         $PE = E \setminus e_j.Path(t)$ ;
 8         foreach $e_k \in PE$ do
 9             Calculate $ShortestPathCost(e_j, e_k)$ based
               on $e_j.memory$;
10         end
11         Select $e_k \in PE$ based on Eq. 4;
12         $e_j.Path(t+1) \leftarrow e_j.Path(t) \cup e_k$;
13         Update $e_j.memory$ based on $e_j.Path(t+1)$;
14         Update $ShortestPath(e_j, e_k, :)$;
15     end
16     Generate a random operation order $O(t)$ with
       $e_i.volume \neq 0$ for all $1 \leq i \leq n$ ;
17     foreach $e_l \in O(t)$ do
18         $V(e_l) = \{e_j|e_j.Path(t+1) = e_l\}$;
19         $S(e_l) = sort(V(e_l))$;
20         foreach $e_m \in S(e_l)$ do
21             if $e_l.volume \neq 0$ then
22                 Update $e_l.volume$ and $e_m.volume$;
                   Update $RMatrix$ based on
                   $ShortestPath(e_m, e_l, :)$;
23             end
24         end
25     end
26 end
```

Algorithm 3: Self-avoiding Random Walk with Information Sharing. In the algorithm, steps 7-11 are for the self-avoiding random walk; steps 16-25 are for the competition trading. We use $e_i.Path(t)$ to record the nodes that entity e_i has visited up to round t, and $ShortestPath(e_i, e_j, :)$ to record the dynamically changing lowest cost path from e_i to e_j.

Algorithm 4 *Self-avoiding Random Walk with Information Passing*: In this algorithm, at each round (i) each entity memorizes cost information of visited nodes to determine the next step of random walk, and (ii) entities pass trading requests to potential partners in its request list. The random walk is the same as that in Algorithm 3. For the trading part, at each round, each entity e_i will calculate the minimum costs $ShortestPathCost(e_i, :)$ to other nodes that it has not visited, and sort the entities by cost in increasing order. The request list $e_i.RL$ of e_i with size s contains entities who are top s in the sorted list of e_i. In this paper, the size of request list increases as the round t increases. Each entity will only agree to trade with another entity who is in its request list.

5 Simulations

In this section, we will describe several simulations to evaluate the performances of the local behavior-based algorithms. The four behavior-based algorithms presented in Section 4.4 are compared with the optimal solutions in terms of (i) distribution rate of energy resources, (ii) the global cost of final energy flow network, and (iii) the per unit cost of final energy flow network. The optimal solutions are calculated by a static and centralized method.

5.1 Settings

There are three inputs for the simulations: the number of entities, the per unit distribution cost matrix, and entities' $volume$.

- **The number of entities:** In reality, the energy distribution problems may have different scales, for example, the distribution network may have scale about 100 (i.e., the number of transmission transformer) at a city level, but about 1000 at a country level [25]. Hence, it

```
   Input: Cost matrix $CMatrix_{n \times n}$; Volume of each entity
          $\{e_i.volume|e_i \in E\}$
   Output: Energy flow network $RMatrix_{n \times n}$
1  foreach $e_i, e_j \in E$ do
2      $e_i.Path(1) \leftarrow e_i$;
3      $ShortestPath(e_i, e_j, :) \leftarrow e_i$;
4  end
5  foreach Round $t = 1 : (n-1)$ do
6      foreach $e_i.volume \neq 0$ do
7          $PE = E \setminus e_i.Path(t)$ ;
8          foreach $e_k \in PE$ do
9              Calculate $ShortestPathCost(e_i, e_k)$ based
               on $e_i.memory$;
10         end
11         Generate request list $e_i.RL$ with size $t$;
12     end
13     Generate a random operation order $O(t)$ with
       $e_i.volume \neq 0$ for all $1 \leq i \leq n$ ;
14     foreach $e_l \in O(t)$ do
15         foreach $e_m \in e_l.RL$ do
16             if $e_l \in e_m.RL$ then
17                 Update $e_l.volume$ and $e_m.volume$;
18                 Update $RMatrix$ based on
                   $ShortestPath(e_l, e_m, :)$;
19             end
20         end
21     end
22     foreach $e_j.volume \neq 0$ do
23         $PE = E \setminus e_j.Path(t)$ ;
24         foreach $e_k \in PE$ do
25             Calculate $ShortestPathCost(e_j, e_k)$ based
               on $e_j.memory$;
26         end
27         Select $e_k \in PE$ based on Eq. 4;
28         $e_j.Path(t+1) \leftarrow e_j.Path(t) \cup e_k$;
29         Update $e_j.memory$ based on $e_j.Path(t+1)$;
30         Update $ShortestPath(e_j, e_k, :)$;
31     end
32 end
```

Algorithm 4: Self-avoiding Random Walk with Information Passing. In the algorithm, steps 6-12 are for each entity to generate request list RL based on its $memory$; steps 13-20 are for the energy trading; steps 23-27 are for the self-avoiding random walk. We use $e_i.Path(t)$ to record the nodes that entity e_i has visited up to round t, and $ShortestPath(e_i, e_j, :)$ to record the dynamically changing lowest cost path from e_i to e_j.

is necessary to evaluate the performances of different behavior-based algorithms in different scales. In this paper, we preform the simulations for the distribution problem with $n = 10, 50, 100, 500, 1000$.

- **Per unit distribution cost matrix:** The values in the per unit distribution cost matrix are randomly generated from region $[10, 1000]$ to reflect the high cost heterogeneity between each pair of suppliers/demanders. Because we focus on the relative comparison of the behavior-based algorithms, the absolute value of the per unit distribution cost does not affect the final analysis of the algorithms.
- **Entities' *volume*:** The $volume$ of each entity is randomly generated from region $[-100, 100]$, where the global supply and demand are balanced. Similarly, the absolute value of entities $volume$ will also not affect the relative comparisons of the behavior-based algorithms.

5.2 Simulation Results and Observations

Distribution rate: The distribution rate is measured by the percentage of energy supply that has been distributed to consumers. Simulation results show that all the four behavior-based algorithms can successfully distribute all energy supply to consumers. This is because for the static energy distribution problem, the $volume$ of each entity remains constant during the dynamic process. According the the self-avoiding random walk, each entity will visit all other entities within $n-1$ rounds. Because of the simple decision-making rules each entity adopts, all supply will be distributed within $n-1$ rounds.

Distribution costs: The distribution costs are measured by global and per unit costs of final energy flow networks proposed in Section 3. Figure 1 shows the global cost comparisons of the four algorithms we proposed in Section 4.4 and the optimal solution. We can find that comparing with optimal solution, Algorithms 1 and 2 have quite worse performances. However, this is not surprising because entities in Algorithm 1 have limited memory and behave without considering cost at all. Though in Algorithm 2, entities may compete for trading when they visit the same node, this kind of competition is proved to be helpless for the minimization of global energy flow cost as shown in Figure 1. In Algorithm 3, by adopting information of visited nodes to calculate path with minimum cost and select trade partners, the entity can find appropriate walk path with much smaller energy flow cost. However, it is obvious that Algorithm 3 is still far away from optimal solution because (i) entities in this algorithm behave based only on cost information in its memory while the optimal solution is calculated by centralized algorithm based on complete information, and (ii) Algorithm 3 only uses cost information, however, there is another information, i.e., the availability of resources, which may help an entity to quickly find appropriate trade partners. We can observe from Figure 1 that Algorithm 4 achieves better performance than Algorithm 3 in terms of global cost of energy flow network. This is because in Algorithm 4, each entity calculates request list based on cost information in its memory before making decisions to trade.

The similar performance results can be observed for the per unit cost of energy flow network in Figure 2.

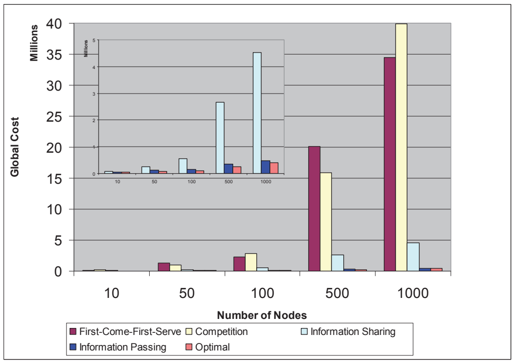


Figure 1: Global cost comparisons among four algorithms for different number of entities. The costs are calculated by Equation 1. The results show that Algorithm 4 preform better than other three Algorithms, where the global costs of Algorithm 4 are very close to the optimal solution.

Scalability: It can be observed that the per unit cost of energy flow network of Algorithm 4 approaches to optimal solution (see inset of Figure 2) as the number of nodes increases from 10 to 1000. This is because in Algorithm 4, entities collect cost information by exploring on the distribution network through more efficient random walks, and further determine potential trading partners (i.e., request list) by collected cost information. The larger the network scale, the more effective and efficient the behaviors of entities will become. Scalability is a very important characteristic for the local behavior-based paradigm to tackle huge distribution systems, where static and/or centralized algorithms cannot perform well. This evidence also shows that it is feasible and advantageous to study huge and complex systems from a bottom-up point of view.

Remark: The main purpose of this simple simulation is (i) to show the possibility that global objectives can be approximately reached through local behavior-based autonomous entities with even simple behavioral rules and decision-making rules, and (ii) to study the effects of different behavioral rules and decision-making rules on the global performances of the behavior-based paradigm. Evidences show that appropriate exploration behavior (i.e., the self-avoiding random walk in this paper) and regulation behavior (i.e., the request list generated in Algorithm 4) play important roles for the local behavior-based paradigm to achieving better global performances.

6 Conclusion and Future Work

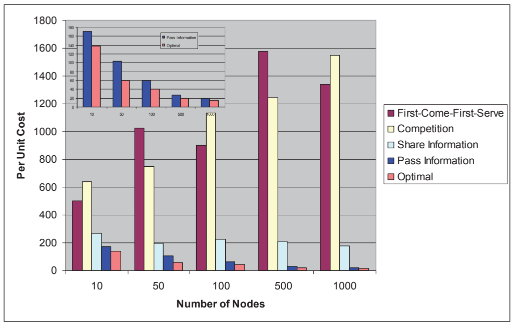


Figure 2: Per unit energy flow cost comparisons among four algorithms for different number of entities. Inset: Compare Algorithm 4 with optimal solution. The per unit energy flow costs are calculated by Equation 2. The results show that the per unit cost of energy flow network of Algorithm 4 approaches to optimal solution as the number of nodes increases from 10 to 1000.

In this paper, we present a local behavior-based paradigm that draws on the methodology of autonomy-oriented computing (AOC) for energy distribution problems, which are by nature, open, and dynamically changing over time. According to AOC, the entities in the behavior-based paradigm can spontaneously interact with each other, and operate based on their behavioral rules and decision-making rules. Simulation results on four algorithms with gradually increased behavioral complexity reveal that global objectives can be effectively and efficiently approached by involving explorations (i.e., self-avoiding random walk on the distribution network) and collective regulation (i.e., the request lists calculated by cost information in entities' memory). Furthermore, behavior-based paradigm with appropriately designed entity profiles and behavioral rules may also have scalable performance, i.e., the per unit cost of energy flow network of Algorithm 4 gradually approaches to optimal solution (see inset of Figure 2) as the number of nodes increases from 10 to 1000.

The main objectives of this paper are not only to solve the energy distribution problem, but to present a natural behavior-based paradigm with respect to the energy distribution problem so that dynamic energy distribution problems can be studied. As describe in Section 1.1, many systematic properties (i.e., vulnerability, criticality, and stability) can be involved in an open, dynamic energy distribution problem. For example, the energy supply vulnerability may relate to whether the distribution system can make sure sufficient energy supply for each energy consumer when supply of energy resources are interrupted for any reasons; the energy distribution criticality analysis may help to find out the critical suppliers/consumers in the distribution network. Understanding systematic properties of an energy distribution problem is essential for us to design robust mechanisms in the future to improve energy distribution efficiency, to study the formation of energy trading markets, to control the cascading failure of distribution networks (e.g., power grid), and so on. Since the energy distribution activities are by nature, performed by highly distributed energy suppliers and consumers, we conjecture that the local behavior-based paradigm (focusing on the local interactions of supplier/consumer entities) in this paper are more feasible than traditional centralized approaches to study the essences of the dynamic energy distribution problems.

References

[1] Worldwide look at reserves and production. *Oil and Gas Journal*, 106(48):23–24, December 2008.

[2] BP statistical review of world energy 2009. Technical report, British Petroleum, London, UK, 2009.

[3] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the north american power grid. *Physical Review E*, 69(2):025103, Feb 2004.

[4] G. A. Backus. Fossil1: Documentation. Technical report, Resource Policy Center, Dartmouth College., 1977.

[5] M. Barthélemy and A. Flammini. Optimal traffic networks. *Journal of Statistical Mechanics*, (L07002), 2006.

[6] M. Bjørdal. *Topics on Electricity Transmission Pricing*. PhD thesis, Norwegian School of Economics and Business Administration, Bergen, 2000.

[7] H. Chao and S. Peck. A market mechanism for electric power transmission. *Journal of Regulatory Economics*, 10(1):25–59, 1996.

[8] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[9] P. S. Dutta, N. R. Jennings, and L. Moreau. Adaptive distributed resource allocation and diagnostics using cooperative information-sharing strategies. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 826–833, 2006.

[10] M. Elkashlan, Z. Chen, I. B. Collings, and W. A. Krzymien. Selection based resource allocation for decentralized multiuser communications. *Physical Communication*, 1(3):194–208, 2008.

[11] T. S. Fiddaman. *Feeedback Complexity in Integrated Climate-Economy Models*. PhD thesis, the Alfred P. Sloan School of Management. Massachusetts Institute of Technology. Cambridge, MA 02139., 1997.

[12] J. Forrester. *World Dynamics*. Wright-Allen Press, 1971.

[13] IEA. International energy outlook 2009. Technical report, International Energy Agency, 2009.

[14] IEA. Key world energy statistics 2009. Technical report, International Energy Agency, 2009.

[15] IEA. World energy model - methodology and assumptions. Technical report, International Energy Agency, 2009.

[16] K. Iyer and M. N. Huhns. Negotiation criteria for multiagent resource allocation. *Knowledge Engineering Review*, 24(2):111–135, 2009.

[17] M. Jacyno, S. Bullock, M. Luck, and T. Payne. Understanding decentralised control of resource allocation in a minimal multi-agent system. In *AAMAS '07: Proceedings of the sixth international joint conference on Autonomous agents and multiagent systems*, pages 1–3, 2007.

[18] X. Li and L.-K. Soh. Hybrid negotiation for resource coordination in multiagent systems. *Web Intelligence and Agent Systems*, 3(4):231–259, 2005.

[19] J. Liu. Autonomy-oriented computing: The nature and implications of a paradigm for self-organized computing. *Keynote Talk at The* 4^{th} *International Conference on Natural Computation, and the* 5^{th} *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 3–11, 2008.

[20] K. T. Midthun, M. Bjorndal, and A. Tomasgard. Modeling optimal economic dispatch and system effects in natural gas networks. *The Energy Journal*, 30(4), 2009.

[21] A. E. Motter. Cascade control and defense in complex networks. *Physical Review Letters*, 93(9):098701, 2004.

[22] A. E. Motter and Z. Toroczkai. Introduction: Optimization in networks. *Chaos*, 17(026101), 2007.

[23] R. F. Naill. *COAL1: A Dynamic Model for the Analysis of United States Energy Policy*. PhD thesis, the Thayer School of Engineering, Dartmouth College, 1976.

[24] A. Quelhas, E. Gil, J. D. McCalley, and S. M. Ryan. A multiperiod generalized network flow model of the U.S. integrated energy system: Part I - model description. *IEEE Transaction on Power Systems*, 22(2):829–836, May 2007.

[25] A. Quelhas and J. D. McCalley. A multiperiod generalized network flow model of the U.S. integrated energy system: Part II - simulation results. *IEEE Transaction on Power Systems*, 22(2):837–844, May 2007.

[26] F. Schweppe, M. Caramanis, R. Tabors, and R. Bohn. *Spot Pricing of Electricity*. Kluwer Academic Publishers, Norwell, Massachusetts, 1988.

[27] S. Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, 2(3):271–379, 2007.

[28] H. Simon, T. Zacharia, and R. Stevens. Modeling and simulation at the exascale for energy and the environment. Technical report, Report on the Advanced Scientific Computing Research Town Hall Meetings on Simulation and Modeling at the Exascale for Energy, Ecological Sustainability and Global Security (E3), 2007.

[29] A. P. Sokolov, C. A. Schlosser, S. Dutkiewicz, S. Paltsev, D. W. Kicklighter, H. D. Jacoby, R. G. Prinn, C. E. Forest, J. Reilly, C. Wang, B. Felzer, M. C. Sarofim, J. Scott, P. H. Stone, J. M. Melillo, and J. Cohen. The mit integrated global system model (IGSM) version 2: Model description and baseline evaluation. Technical report, MIT, 2005.

[30] J. D. Sterman. *The Energy Transition and the Economy: A System Dynamics Approach.* PhD thesis, the Alfred P. Sloan School of Management. Massachusetts Institute of Technology. Cambridge, MA 02139., 1981.

[31] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebber, M. D. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki. Rules for biologically inspired adaptive network design. *Science*, 327(5964):439–442, 2010.

[32] National Energy Board. A compendium of electric reliability frameworks across canada. Technical report, The National Energy Board, Canada, 2004.

[33] Oil Division. A compendium of electric reliability frameworks across canada. Technical report, Petroleum Resources Branch, Canada, 2008.

Selectivity Estimation of XPath for Cyclic Graphs

Yun Peng Byron Choi Jianliang Xu
Department of Computer Science
Hong Kong Baptist University
{ypeng, choi, xujl}@comp.hkbu.edu.hk

ABSTRACT

Recent interests on the Semantic Web, Web ontology and XML, among other topics, have sparked a renewed interest on graph-structured databases. Path queries are often used to retrieve graph-structured data (or simply graphs) from the database. A crucial and classical query optimization has been query selectivity estimation. However, the majority of existing works on selectivity estimation focuses on relational and tree data. In this paper, we investigate selectivity estimation on path queries on possibly cyclic graph data. To facilitate selectivity estimation on cyclic graphs, we propose a matrix representation of graphs extended from prime labeling, an index for reachability queries on directed acyclic graphs. With this representation, we exploit consecutive ones property (C1P) of matrix. As a consequence, a node is mapped to a point in a 2-dimensional plane and a query is mapped into multiple points. We adopt histograms for selectivity estimation. We perform an experimental evaluation on the space and time efficiency and accuracy of the proposed technique.

1. INTRODUCTION

Graph-structured databases have a wide range of emerging applications, e.g., the Semantic Web, eXtensible Markup Language (XML), biological databases and network topologies. To-date, there has already been voluminous real-world (possibly cyclic) graph-structured data []. To retrieve subgraphs from a large graph-structured database efficiently, various query optimization techniques have been proposed. Among others, selectivity estimation of queries has been a crucial query optimization technique in databases. In a nutshell, given a query, we want to determine the number of results of the query, without invoking potentially costly query evaluation. Selectivity estimation has been built in the query optimizer of all commercial relational databases. However, the majority of previous research on selectivity estimation, with few exceptions (see Section 2), focuses on relational and tree-structured data. In this paper, we propose an accurate and efficient selectivity estimation for graph-structured databases.

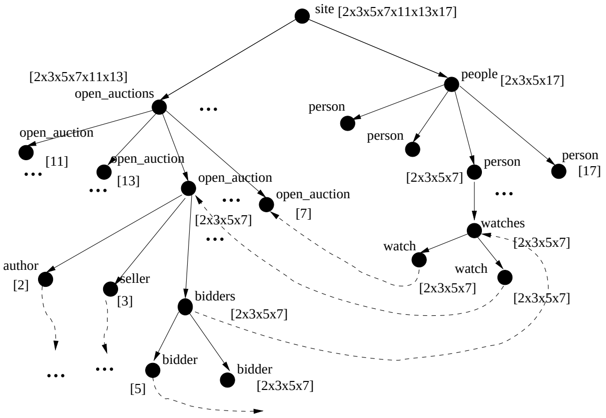


Figure 1: An example graph of auction information

Path queries have been a popular and classical tool for retrieving subgraphs from a graph-structured database. To facilitate our technical discussions, we consider a fragment of XPath query (see Section 3.2). Let us consider a simplified synthetic XMark data graph [14] shown in Figure 1, where . The graph encodes auction information, where people watch over auctions and bidders bid items. Consider an example query `//open_auction[//bidder//open_auction]`, which selects the auctions that are still open and have bidders who watch some `open_auction`s. Note that the XMark data is cyclic and the XPath query is recursive, where the query selects `open_auction`s that have `open_auction` descendants. In XMark with a scaling factor 0.1, there are 120, 708 and `bidder`s, `open_auction`s and `open_auction//bidder`s, respectively. To minimize intermediate result size, the query `//open_auction//bidder` is evaluated prior to `//bidder` from the root.

In recent years, there have been studies on selectivity estimation of XML data and XPath queries. In particular, Wu *et al.* [19] proposes to adopt histograms to selectivity estimation of XML queries. An advantage of this technique is that histograms are by far the most popular technique for query result estimation. The proposed technique relies on an interval representation of nodes, which presumes tree data. While a previous technique [2] extends the interval representation to support directed acyclic graphs, each node is potentially represented by multiple intervals. The storage requirement of multiple intervals per node can be prohibitvie.

In comparison, we represent a node with a single interval but a query with multiple intervals.

There has been a stream of work on summarizing cyclic graphs, namely XSketch [12] and TreeSketch [13]. The main idea of the work is to compute minimal bisimulation of a graph, which is used as a summary structure of the graph. However, the bisimiluation graph can still be large. Local bisimulation has been adopted to further reduce the size of the bisimulation graph. However, the estimation accuracy relies on a strong statistical assumption (uniform distribution) of data. In addition, bisimulation is not yet available in commercial databases.

In this paper, we propose selectivity estimation technique for cyclic graphs. The novelities of this paper are two. (i) We propose to adopt an index for descendant-ancestor queries that can support cyclic graphs. (ii) The second technique is to keep the index of the cyclic data simple while expanding a query for estimation. The intuitions of the two techniques can be described as follows.

We adopt prime number labeling scheme (or simply *prime labeling*) [17, 18] in this work. Prime labeling was originally proposed to determine descendant-ancestor relationship among nodes of trees. This scheme associates each node with an exclusive prime number and labels each node with the product of its parents' labels and the exclusive prime number of itself. Reachability between nodes is mapped to divisibility test of labels. In later work [17] this labeling scheme is extended to support directed acyclic graphs (DAGs). However, to the best of our knowledge, no further work has been presented to extend this idea to cyclic graphs, not mention to selectivity estimation for graph-structured data. In addition, the size of prime labeling is often smaller than that of transitive closure of a graph. We propose a binary matrix representation of prime labeling to further reduce the size of prime labeling.

Next, given a matrix representation of a graph, we transform it into a matrix with the consecutive ones property (C1P). Some columns are duplicated in the transformation. Subsequently, a node of a cyclic graph can be represented as an interval of column ID (*start*, *end*), as opposed to multiple intervals [2]. The nodes are summarized in a positional histogram. The complexity of querying is intuitively "pushed" into the mapping between duplicated column IDs. Given a query, we expand it into multiple query points using the mapping of duplicated column IDs.

The contributions of this paper are as follows.

- We propose to extend prime labeling of DAGs to cyclic graphs. We propose a binary matrix representation of prime labeling of cyclic graphs.

- We propose matrix transformation to map a node of a graph to an interval and a query to possibly multiple intervals. A two-dimensional histogram is adopted to summarize the graph. We propose an estimation procedure with the histogram.

- We performed a preliminary experimental study on the proposed techniques. It verifies that our techniques are efficient and accurate for real and synthetic data graphs.

The rest of this paper is organized as follows. Section 2 discusses related works. Section 3 presents the terminologies and notations of this work. Section 4 proposes the prime labeling and its matrix representation for cyclic graphs. Section 5 presents matrix transformation for interval representation of the graph. We present a preliminary experimental study in Section 6. We present a list of future works in Section 7.

2. RELATED WORK

There has been a host of recent works on selectivity estimation on path queries. The techniques can be roughly classified into two categories: graph-based approaches and relational-based approaches.

While graph-structured data model has its root at network data model, it was revisited in Tsimmis project, where Object Exchange Model (OEM) is proposed. DataGuide [11] is proposed to optimize query evaluation on graphs. Graphs are considered as non-deterministic automata and their DataGuide is the deterministic automata of the graph. DataGuide has been extended to support approximate query processing [8]. [6] proposes a straight-line grammar (STL), which is a special form of context-fre grammar, to summarize a data graph. To reduce the size of the grammar, [6] proposes to use a wildcard to simplify some non-terminals in a production.

Another graph-based approach [12,13] is derived from the notion of bisimulation of graphs. [12] proposes to use the minimum/minimal bisimulation as the summary structure of a data graph. To further reduce the size of bisimulation, a notion of local bisimulation [9] has been applied. To recover the path information from a local bisimilar graph, graph stability is exploited and uniform distribution of nodes are assumed.

Histograms in relational databases have been adapted to support selectivity estimation of queries on graphs. Chen *et al* [4] proposes an interval representation of echo node of a graph. The start and end position of the interval is used as the coordinates of a point in a 2-dimensional plane. Next, a positional histogram is used to summarize the point for estimation.

3. BASIC DEFINITIONS

In this section, we present the notations used.

3.1 Data Model

In this paper, we study directed node-labeled rooted data graphs, or simply *graphs* in the subsequent discussions. A graph can be denoted as $G = (V, E, r, \Sigma, \lambda, oid)$, where V is a set of nodes and E: $V \times V$ is a set of edges, $r \in V$ is a root node, Σ is a set of labels and λ:$V \rightarrow \Sigma$ is a function that returns the label of a node and *oid* is a function that returns a unique identifier of a node. For simplicity, we may denote a graph as (V,E) when other components are not relevant to the discussions.

3.2 Path Queries

Among the queries on graphs, XPath has been studied more extensively recently than others and it has been a indispensable part of eXtensible Markup Language (XML) – the *de facto* standard for electronic data exchange. Hence, we consider a fragment of XPath. The syntax is given in BNF below.

$$\begin{aligned} p &::= \epsilon \mid A \mid * \mid // \mid p/p \mid p[q], \\ q &::= p \mid q \wedge q \mid q \vee q, \end{aligned}$$

where ϵ, A, $*$ and '/' denote the *self-axis*, a label (tag), a wildcard and the *child*-axis, and '//' stands for */descendant-or-self::node()/*, respectively; and q in $p[q]$ is called a *filter*, in which s is a constant (string value), and '$\wedge$' and '$\vee$' denote conjunction and disjunction, respectively. For //, we abbreviate $p_1/$ // as $p_1//$ and // $/p_2$ as $//p_2$. We use $r[|p|]$ to denote the evaluation of the query p from the node r.

Let R be the set of nodes of the evaluation, where $R = r[|p|]$. In this paper we want to determine $|R|$ efficiently and accurately.

4. REPRESENTATION OF CYCLIC GRAPHS

To efficiently evaluate the *descendant-or-self* axis on cyclic graphs, we propose our index in this section. We extend prime labeling [18] to our problem. The benefits are twofold: (i) Prime labeling can support cyclic graphs with minor modifications and the labels are simple; and (ii) Prime labeling of a graph is often smaller than the transitive closure of the graph, in practice.

4.1 The Original Prime Labeling

Prime labeling is originally proposed for indexing trees not cyclic graphs. The main idea of prime labeling is that each node is labeled with a product of prime numbers such that the ancestor-descendent relationship between nodes could be determined by using the division of the labels. A node n_1 is an ancestor of another node n_2 if and only if the label of n_2 is *divisible* by that of n_1. In [18], a unique prime number is assigned to each leaf node. The prime label of a node is the product of the prime label of its children. Obviously, such labeling works only on trees. To extend prime labeling to support DAGs, [17, 18] requires a unique prime number per node.

4.2 Prime Labeling for Cyclic Graphs

To support cyclic graphs, the extensions of the previous work on prime labeling ([18], [17]) are two. (i) Prime labeling needs to support possibly multiple strongly connected components (SCCs) in cyclic graphs. By definition, each node in a SCC can reach any other node in the SCC. Therefore, the nodes in a SCC can be associated with the same prime label. (ii) Previous work [17] on prime labeling uses excessive prime numbers (one prime number per node). We use fewer number of (unique) prime numbers needed for labeling and therefore reduce the overall size of prime labeling.

In particular, we require a new prime number for labeling a node in one of the following situations. (i) The node is a leaf node. (ii) The node has multiple parents.

With the above background, we now give the definition of prime labeling.

Let `get_next`() is a special function which returns a prime number that it has not returned before. Assume that a cyclic graph G has been preprocessed by Tarjan's algorithm [15], where each SCC is reduced to a supernode. Denote the reduced graph (DAG) to be $G'(V', E')$. Each node n is associated with a prime label ℓ as defined in Definition 4.1.

Definition 4.1: The *prime label* ℓ of a node n of the reduced graph $G'(V', E')$ can be defined as follows.

1. If n is a leaf node, then $n.\ell$ = `get_next`().

2. If n is a non-leaf node and n has multiple parents, then $n.\ell$ = `get_next`() $\times \prod_{c \in C} c.\ell$, where C is the set of child nodes of n.

3. Otherwise, $n.\ell = \prod_{c \in C} c.\ell$.

□

```
Input: A data graph G
Output: the matrix representation of a graph M
G' = tarjan(G)
for each n in G'.V in reverse topological order
    if n is a leaf node                         /* Definition 4.1 */
        $n.\vec{\ell}$[get_next()] = 1
    else
        if |n.parent| ≥ 1
            $n.\vec{\ell}$[get_next()] = 1
        for each c in n.children
            $n.\vec{\ell} = n.\vec{\ell} \mid\mid c.\vec{\ell}$
```

Figure 2: Prime Labeling Construction `prime-construct`

The prime label can be assigned to nodes of the reduced graph G' in a reverse-topological order, i.e., a bottom-up traversal. The pseudo-code of the construction (`prime-construct`) is shown in Figure 2. The time complexity of `prime-construct` is $O(|V'| + |E'|)$.

Assume we have a set of A-nodes and B-nodes, denoted as S_A and S_B, respectively. A naive way to determine the number of B-descendants in S_B of the nodes in S_A takes $O(|S_a| \times |S_b|)$. With prime labeling, this can be done by first computing the product of the prime labels of S_A, denoted as M_A and then check the divisibility between the product M_A and the prime label of each node in S_B. This requires $O(|S_a| + |S_b|)$.

4.3 Vector Representation of Prime Labeling

A known issue of prime labeling is that it often results in very large integers. Nowadays, there has been voluminous graph data, such as protein data, social network and XML. In response to these, we propose a vector representation of prime labeling and map division of integers to logic operators of vectors.

Definition 4.2: Suppose that the prime label ℓ of a node n of a graph G is $p_{i_1} \times p_{i_2} \times ... \times p_{i_m}$, where p_{i_j} is the i_j-th prime number. ℓ is then presented by a vector $\vec{\ell}$ where $\vec{\ell}[i_j] = 1$ if and only if p_{i_j} is a factor of *ell* and 0, otherwise. The dimension of the vector is the number of prime numbers used in labeling G. □

Then, a graph can be represented as a matrix. In this work, we always discuss binary vectors and matrices. For simplicity, we may omit the term "binary".

With this representation, division and multiplication of prime labels can be mapped into logical operations on the vector representation of the prime labels. In addition, the property of prime labeling that a set of

Definition 4.3: Given two nodes n_1 and n_2, $n_1.\ell$ is divisible by $n_2.\ell$ if and only if $\neg n_1.\vec{\ell} \wedge n_2.\vec{\ell} = \vec{0}$. □

Definition 4.3 can alternatively be understood that the vector $\neg n_1.\vec{\ell}$ and $n_2.\vec{\ell}$ are *orthogonal*, where the product of the two vectors is 0.

Definition 4.4: Given a set of nodes V and n_2, $\prod_{n \in V} n.\ell$ is divisible by $n_2.\ell$ if and only if $\neg(\bigwedge_{n \in V} n.\vec{\ell}) \wedge n_2.\vec{\ell} = \vec{0}$. □

Therefore, our problem is translated into the following.

Selectivity Estimation. Assume that m prime numbers have been used in labeling and a graph with n nodes. Given a vector representation of a query $\vec{v}$ and a $n \times m$ matrix M, we want to estimate the number of rows in M that are orthogonal to $\vec{v}$. □

Hence, the remaining task is to summarize the matrix for the selectivity estimation problem.

5. AN OVERVIEW OF OUR SOLUTION

In this section, we present an overview of our solution by exploiting the matrix representing proposed in Section 4.

5.1 Consecutive One Property

Given the matrix representation discussed Section 4, we perform a few transformations on the matrix for summarization. First, we convert a matrix into a matrix with a consecutive one property (C1P).

Definition 5.1: A matrix M has the *weak Consecutive Ones Property (C1P)* if its columns can be permuted such that in each row, the ones are adjacent. A matrix M has *strong C1P* if the ones of each row are adjacent. □

Unfortunately, the conversion of a matrix into a C1P matrix is intractable. Worst still, there is no polynomial time approximation algorithm for determining C1P submatrix of a given matrix. Here, we propose a linear time algorithm to convert a matrix into a C1P matrix. Obviously, the matrix returned is not minimum. The algorithm is presented in Figure 3, namely Algorithm `convex_bipartite`.

First, we convert a matrix M into a bipartite graph $G(U, V, E)$, where U and V are two *sequences* of nodes. We create a node in U, for each row of M. Similarly, we create a node in V for each column of M. In addition, we introduce $(u, v) \in E$ if and only if $M[u][v]$ is non-zero. A *convex* bipartite graph is a bipartite graph where the neighbors of each node in U are adjacent. An example of convex bipartite graph is shown in Figure 5. It is obvious that a matrix with C1P can be derived from a convex bipartite graph. However, to obtain a C1P matrix, we will need to introduce new nodes in U. Therefore, a main task of Algorithm `convex_bipartite` is to introduce few new nodes when generating a convex bipartite graph.

Second, we discuss the details of `convex_bipartite`. In Line 01, we simply order the nodes in V by the number of neighbors. In Lines 03 - 08, we expand the bipartite graph fully. Each node has an unique id (assigned by an incremental counter) In Lines 09 - 13, we merge the bipartite graph from left to right. In Lines 14 - 19, we merge the bipartite graph from right to left. The neighbors of a node in v can then represented by the id of its first and last neighbor nodes, which forms an interval of ids. The interval of nodes is that summarized by a two dimensional histogram. In addition, after we obtained a C1P matrix, a node u in U may have been copied multiple times. Hence, we need to record the ids that a node u in U represents. In particular, we store this information in two relations f and f^{-1}, where $f(u)$ returns the ids of nodes that u represent and $f^{-1}(x)$ returns u if $x \in f(u)$.

```
Input: A bipartite graph $G$=($U$, $V$, $E$)
Output: A convex bipartite graph $G'$=($U \cup U'$, $V$, $E'$)
/* sort $v \in V$ by $v.N.length$ from smallest to largest */
01 $G$=sort($G$)
/* duplication */
02 count = 0
03 for $i$ in [1 ... $|V|$]
04   for each $u_j$ in $v_i.N$
05     construct a new node $u'$ in $U'$
06     construct a new edge $(v_i, u')$ in $E'$
07     set $u'.cid = u_j.cid$
08     set $u'.id$ = count++
/* two passes merging */
/* merging from left to right */
09 for $i$ in [1 ... $|V| - 1$]
10   for each $u_j$ in $v_i.N$
11     $p$ = find($u_j, v_{i+1}.N$)
12     if $p \neq -1$
13       mergeL2R($u_j, u_p, i$)
/* merging from right to left */
14 for $i$ in [$|V|$ ... 2]
15   for each $u_j$ in $v_i.N$
16     if $u_j.N.length == 1$
17       $p'$ = find($u_j, v_{i-1}.N$)
18       if $p' \neq -1$ and mergable($u_j, u_{p'}, i$)
19         mergeR2L($u_j, u_{p'}, i$)
20 returnG'
```

Figure 3: Transformation to a convex bipartite graph `convex_bipartite`

A property of the the intervals is that the interval of a node's descendants must be contained in the interval of the node. In the two dimensional histogram representation, the descendant point must be on the bottom right direction of the point of the node. To determine the count of the descendants of a node, we simply count the number of points in the bottom right direction of the data point of the node. Two-dimensional histogram divides the two-dimensional space (plane) into grids. Each grid stores the number of data points contained in the grid.

In all, given a matrix M, `convex_bipartite` returns a C1P M' and a map that maps a node u to the ids that u represents f and the inverse of this map f^{-1}.

5.2 Estimation Framework

We assume that the input of an estimation step is (i) a descendant step $//A$ in a path query, where A is a label, (ii) the graph represented as discussed in Section 5.1 and (iii) a set of grids in the two dimensional histogram, which contain the set of input nodes of the query $//A$.

For each grid d, we determine the A-grid on the bottom right of the grid. For a gird that is completely in the bottom right region of d, all its nodes are A descendants of the node. For the grid that is partly in the bottom right region of d, we assume the number of A descendants is proportional to the area that is in the bottom right region – we assume data

```
/* search in a list for a particular element */
find(u,list)
for $i$ in [1 ... list.length]
    if list[$i$].cid == u.cid
        return $i$
return -1

/* merge $u_j$ and $u_p$ from left to right */
mergeL2R($u_j, u_p, i$)
remove $u_j$ from $v_i.N$
add $u_p$ to $v_i.N$
move $u_p$ to left most position of $v_{i+1}.N$

/* merge $u_j$ and $u_{p'}$ from right to left */
mergeR2L($u_j, u_{p'}, i$)
remove $u_j$ from $v_i.N$
add $u_{p'}$ to $v_i.N$
move $u_{p'}$ to right most position of $v_{i-1}.N$

/* determine whether $u_j$ and $u_{p'}$ are mergable */
mergable($u_j, u_{p'}, i$)
/* recall that after mergeL2R $v_i.N$ is an ordered list */
for $k$ in [$p' + 1$ ... $v_{i-1}.N$]
    if $u_k.N.length < u_{p'}.N.length$
        return false
return true
```

Figure 4: Algorithm `merge`

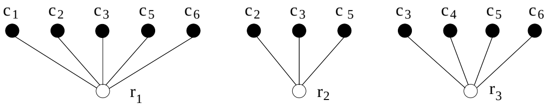


Figure 5: A simple example of convex bipartite graph

is evenly distributed in the grid. This gives us the count of $//A$ and a set of grids that contain some $//A$.

To estimate a path query $//A_1...//A_n$. Initially, we start with the grid containing the root node. We repeat the estimation described above for $//A_1$, ... and $//A_n$.

6. EXPERIMENTAL EVALUATION

In this section, we present a preliminary experimental evaluation that verifies the accuracy and efficiency of our proposed techniques.

We run our experiments on a machine with a Dual 4-core 2.93GHz with 30GB memory running Solaris OS. The implementation is written in `Java`. We use `Matrix` to refer to our technique.

We used `XMark` [14] to obtain a set of data graphs. The scaling factor was ranged from 0.1 - 0.5. We implement a query generator based on the description in Polyzotis *et al.* [12]. The average length of queries is 4. We generate 10,000 queries on the `XMark` data.

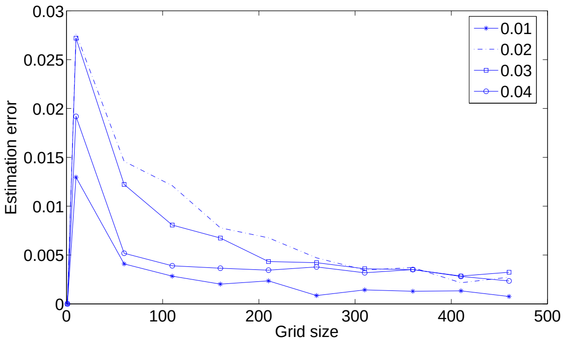


Figure 6: Estimation error

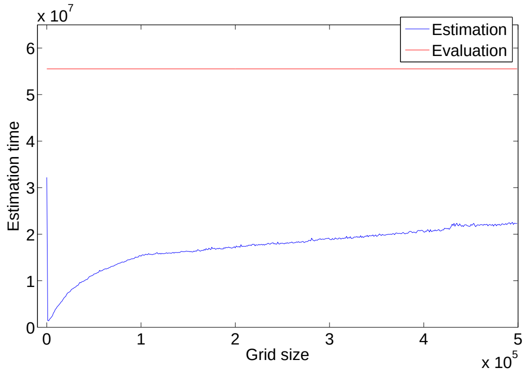


Figure 7: Estimation time

The estimation error of the queries on various `XMark` graphs is shown in Figure 6. The x-axis is the gird size of the two-dimensional histogram. The estimation error increases as the grid size increases. This is because more details of the data points have been summarized by larger grids. The estimation error drops for larger grids because we exploit an query evaluation (which produces exact answer) on the last query step. The reason is that the last step often influential to estimation. To avoid introducing large error in estimation, in the last query step, we employ a query evaluation. For large grids, many grids are involved in the last query step for query evaluation. Hence, the estimation error is low.

We compared our result with a previous work [12]. The estimation errors under these settings are ranged from 8% to 10%. In comparison, our estimation never exceeds 3%.

Next, we compare the time for query estimation and query evaluation. The result is shown in Figure 7. The figure shows query estimation is always smaller than 33% of query evaluation time.

The next experiment shows a problem that we are currently working on. While the node is neatly summarized in two-dimension history. However, it assumes the existence of two mappings f and f^{-1}. Figure 8 shows the average size of $f(u)$ under different scaling factor of `XMark`. As shown, the average size of $f(u)$ is large, at least 100. And the average size increases linearly with s.f. Therefore, we are currently working on compression techniques for f and f^{-1}.

While f and f^{-1} are large, not the entire map is used in estimating the selectivity of a query. We use a simple indexing technique on f and f^{-1} to skip lookups that would always generate empty results. Figure 9 shows the number of lookups that are skipped. This shows that the skipping is linearly to the scaling factor of `XMark`.

Next we investigated the effect of optimizations that we used to reduce the estimations. First, we used query evaluation on the last query step. We ran an experiment with

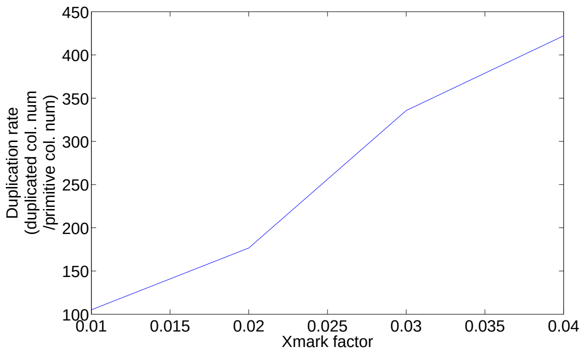


Figure 8: Number of copies of a nodes in f and f^{-1}

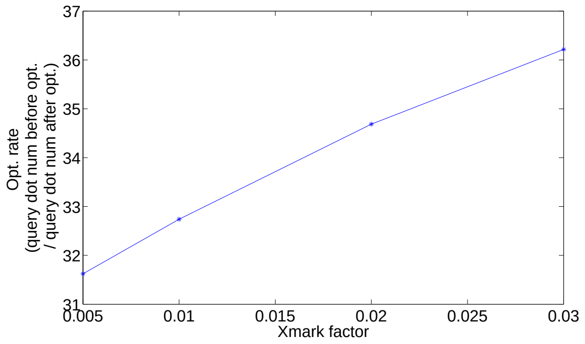


Figure 9: Performance of query dot generation opt.

and without this optimization to illustrate the effect. The result is presented in Figure 10. The figure shows that the optimization clearly reduces estimation errors.

Finally, we explore another optimization to reduce the estimation errors. The optimization is to maintain a bounding rectangle of the points in a grid. Hence, instead of selecting the entire grid or using uniform data distribution on data, we use the data points to limit the region where there are data points. The result is shown in Figure 11. It shows that this optimization clearly reduces estimation errors.

7. FUTURE WORKS

At the time this report is drafted, we are optimizing the proposed techniques. We are extending this report in a few directions. (i) We are compressing the representation of f and f^{-1}. The idea is the to detect frequent parttens in the maps and replace patterns with pattern ids. (ii) We are extending the experimental evaluation with larger datasets. The current dataset can support up to `XMark` up to a s.f. 0.5. (iii) We are deriving estimation upper and lower bounds of our method. (iv) Lastly, we are writing up the algorithms and optimizations that have been used in our experiments.

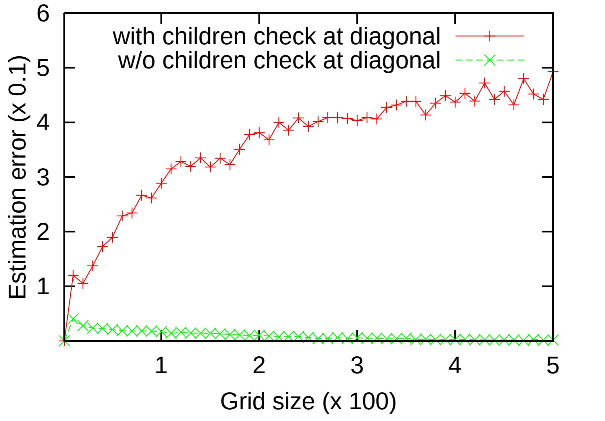


Figure 10: Performance of children check at diagonal grids at last step

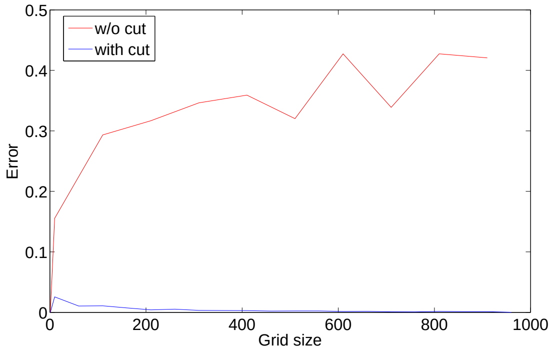


Figure 11: Performance of interval-cut operation

8. REFERENCES

[1] A. Aboulnaga, A. R. Alameldeen, and J. F. Naughton. Estimating the selectivity of xml path expressions for internet scale applications. In *VLDB*, pages 591–600, 2001.

[2] R. Agrawal, A. Borgida, and H. V. Jagadish. Efficient management of transitive relationships in large data and knowledge bases. In *SIGMOD*, pages 253–262, 1989.

[3] A. Chaudhary, D. Chen, X. Hu, M. Niemier, R. Ravichandran, and K. Whitton. Fabricatable interconnect and molecular qca circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 26(11):1978–1991, Nov. 2007.

[4] Z. Chen, H. V. Jagadish, F. Korn, N. Koudas, S. Muthukrishnan, R. Ng, and D. Srivastava. Counting twig matches in a tree. In *ICDE*, pages 595–604, 2001.

[5] D. Fisher and S. Maneth. Structural selectivity estimation for xml documents. In *ICDE*, pages 626–635, 2007.

[6] D. K. Fisher and S. Maneth. Structural selectivity estimation for xml documents. In *ICDE*, pages 626–635, 2007.

[7] J. Freire, J. R. Haritsa, M. Ramanath, P. Roy, and J. Siméon. Statix: making xml count. In *SIGMOD*, pages 181–191, 2002.

[8] R. Goldman and J. Widom. Approximate dataguides. In *In Proceedings of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats*, volume 97, pages 436–445, 1999.

[9] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. Exploiting local similarity for indexing paths in graph-structured data. In *ICDE*, page 129, 2002.

[10] L. Lim, M. Wang, S. Padmanabhan, J. S. Vitter, and R. Parr. Xpathlearner: an on-line self-tuning markov

histogram for xml path selectivity estimation. In *VLDB*, pages 442–453, 2002.

[11] J. McHugh and J. Widom. Query optimization for xml. In *VLDB*, pages 315–326, 1999.

[12] N. Polyzotis and M. Garofalakis. Xsketch synopses for xml data graphs. *ACM Trans. Database Syst.*, 31(3):1014–1063, 2006.

[13] N. Polyzotis, M. Garofalakis, and Y. Ioannidis. Approximate xml query answers. In *SIGMOD*, pages 263–274, 2004.

[14] A. Schmidt, F. Waas, M. Kersten, M. J. Carey I. Manolescu, and R. Busse. Xmark: A benchmark for xml data management. In *VLDB*, pages 974–985, 2002.

[15] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[16] W. Wang, H. Jiang, H. Lu, and J. X. Yu. Bloom histogram: Path selectivity estimation for xml data with updates. In *VLDB*, pages 240–251, 2004.

[17] G. Wu, K. Zhang, C. Liu, and J.-Z. Li. Adapting prime number labeling scheme for directed acyclic graphs. In *DASFAA*, pages 787–796, 2006.

[18] X. Wu, M. L. Lee, and W. Hsu. A prime number labeling scheme for dynamic ordered xml trees. In *ICDE*, page 66, 2004.

[19] Y. Wu, J. M. Patel, and H. V. Jagadish. Using histograms to estimate answer sizes for xml queries. *Inf. Syst.*, 28(1-2):33–59, 2003.

[20] N. Zhang, M. Ozsu, A. Aboulnaga, and I. Ilyas. Xseed: Accurate and fast cardinality estimation for xpath queries. In *ICDE*, pages 168–197, 2006.

Acknowledgements. We are grateful to Neoklis Polyzotis for providing the implementation of [12, 13]. We thank Haibo Hu and Jintian Deng for their comments on the earlier drafts.

Patient Journey Optimization Using A Multi-Agent Approach

Chung Ho CHOI

Abstract

Apart from the physical pains they suffer, most patients nowadays have to endure long waiting times during their patient journeys. While the alleviation of physical pains can mostly be done by the use of drugs, psychological stresses resulting from long waiting times are by no means trivial and always pose a threat to patients' life. Hence, an increasing attention has now focused on shortening the length of patient journey by scheduling patients in a more efficient way. In doing so, because of the decentralized structure of hospital settings, conventional centralized approaches such as operations research are hard to be applied and motivates the use of a multi-agent approach. In this paper, we propose a multi-agent framework for scheduling patients in a decentralized and efficient manner. Particularly, in order to demonstrate the effectiveness of the proposed multi-agent scheduling framework, simulations were performed based on a dataset containing about five thousand cancer patient journeys.

1 Introduction

Not to mention the physical pains they encounter, most patients nowadays also have to confront with psychological stresses resulting from long waiting times during their patient journeys. While physical pains can be alleviated by the use of drugs, psychological stresses resulting from long waiting times are by no means trivial and always pose a threat to patients' life. Since how a patient feels is well-linked to his or her medical condition [3, 8], we should not underestimate the impacts brought by such psychological stresses and thus long waiting times should not be tolerated.

In order to best utilize the existing resources to minimize undesired long waiting times, a well-designed scheduling algorithm is crucial [9]. Meanwhile, though conventional operations research methods have been found effective for centralized scheduling problems [10, 11, 5], most of them are not suitable for hospital settings where decentralized structures are found [6, 9, 2]. As a result, during the design of a patient scheduling algorithm, a multi-agent approach is proposed.

A multi-agent approach is characterized by emphasizing on local interaction and self-organization of different entities being modeled. These properties make it especially suitable for tackling complex tasks with a lot of stakeholders [13, 1]. Multi-agent methods have found applications in a variety of problem domains, such as airport resource scheduling [4], load allocation in transportation logistics [7], supply chain management [12], etc. Recently, it has also been applied to patient scheduling in [6, 9] with some initial success demonstrated. And yet, there are limitations. Paulussen *et al.*, in [6], assume that a quantified health state can be accurately derived as a utility measure for guiding the scheduling process. In [9], Vermeulen *et al.* did not consider the temporal constraints between the treatment operations during the scheduling process.

The objective of this study is to explore the extent to which a patient journey can be improved by better coordinating and mobilizing resources distributed at different medical units. In particular, we formulate the scheduling problem according to the cancer treatment practice in Hong Kong. We propose the use of a multi-agent framework in which autonomous agents interact with each other to arrive an effective overall schedule with reduced waiting times. To evaluate the effectiveness of the proposed framework, we made use of a patient identity anonymized dataset collected by Hospital Authority in Hong Kong which contains 4720 cancer patients with a diagnosis period spanning over 6 months, and have carried out simulations with the proposed approach given different settings of the environment.

The rest of the paper is organized as follows. A patient scheduling problem is formulated in Section 2. Section 3 and Section 4 present the details of the proposed agent-based scheduling algorithm. Section 5 presents some experimental results and Section 6 concludes the paper.

2 PROBLEM FORMULATION

In this section, we first briefly describe the establishment of the cancer clusters in Hong Kong. Then, we formulate the patient scheduling problem for cancer treatment as an optimization problem and explain how our proposed multi-agent framework can be adopted to address the distributed nature of the problem.

2.1 Cancer Patient Treatment - A Hong Kong Scenario

In Hong Kong, there are seven cancer clusters. Figure 1 shows the geographical distribution of the seven cancer clusters in Hong Kong. With the objective not to reveal the performance of individual clusters, we denote the set of the seven clusters as $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_7\}$. Currently, on-demand information exchange among the clusters for scheduling patients is not yet extensively used. That is why it is common for cancer patients to be scheduled to receive treatments at only one cancer cluster, even though some of the treatments could be provided earlier by other clusters.

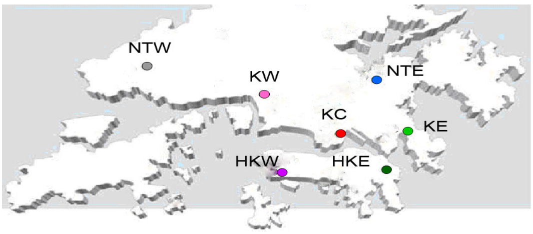


Figure 1. Seven geographically distributed cancer clusters in Hong Kong.

Generally speaking, once the case is suspected to be cancer for a patient, the doctor will specify the patient a treatment plan which contains a sequence of treatment operations. We denote the set of treatment operations as $\Gamma = \{\text{radiotherapy, surgery, chemotherapy}\}$.

To carry out the treatment operations, medical resources are needed. We denote the set of medical resources (or units) as $A = \{\text{radiotherapy unit, operation unit, chemotherapy unit}\}$. We assume that one treatment operation can only be performed at one medical unit of the corresponding type. A patient journey is defined as the duration from the date of histopathological diagnosis to the date of the last treatment operation completed.

2.2 Formulation

Let $K := A \times \mathcal{C}$ be the cartesian product of A and $\mathcal{C}$ giving the complete set of medical units, $M := K \rightarrow \Gamma$ be an one-to-one mapping between K and Γ specifying the treatment type of the medical units, and P be the set of cancer patients being scheduled.

Also, given a patient i, let N_Γ^i denotes the number of treatment operations needed, D_0^i denotes the diagnosis date, D_j^i denotes the date of the j^{th} treatment operation where $1 \leq j \leq N_\Gamma^i$, $V_j^i \in K$ be the unit at which the j^{th} treatment operation is performed, $Tr_j^i \in \Gamma$ be the type of treatment for the j^{th} operation, C_k be the daily capacity (i.e. number of patients that could be treated) of medical unit $k \in K$, T_t be the duration (in days) of treatment type $t \in \Gamma$, and Z be the set of dates on which patient scheduling is being considered.

With the assumption that all the patients are being treated equally in terms of urgency, the scheduling problem can be formulated as:

$$\min_D \sum_{i=1}^{|P|} \sum_{j=1}^{N_\Gamma^i - 1} (|D_j^i - D_{j+1}^i|) \tag{1}$$

with the following constraints to be satisfied:

$$D_{j+1}^i > D_j^i + T_{Tr_j^i} \tag{2}$$

$$\forall d \in Z \quad \left|\{i : D_j^i = d \wedge V_j^i = k \wedge Tr_j^i = M(k)\}\right| \leq C_k \tag{3}$$

$$D_j^i > D_0^i > 0 \tag{4}$$

The objective function in (1) is to minimize the time lags between treatment operations for cancer patients. Constraint (2) ensures the temporal constraints between treatment operations are not violated, constraint (3) is used to ensure all medical units are operating within their capacities. Constraint (4) ensures that patients would only be scheduled to receive treatment operations after their diagnoses.

3 SCHEDULING FRAMEWORK

Theoretically, patient waiting times could be minimized by optimizing (1). However, it is impractical to do so as it is hard to assume that a cancer cluster is willing to share its real-time resource allocation related data (e.g., C_k) with other clusters due to both technical and managerial reasons.

Hence, in this section, we propose the use of the multi-agent approach which tries to model each stakeholder as an autonomous agent and emphasizes on local interactions among the agents. It aims to minimize the information sharing requirement among the clusters and yet to obtain a good enough suboptimal result for the (global) patient journey optimization. In our proposed framework, there are two types of agents, namely *patient agents* and *resource agents*. They interact via some designed protocol for achieving the aforementioned optimization.

3.1 Patient Agent

A patient agent is used to represent one cancer patient and is denoted as P_i with $i = 1, 2, ..., |P|$. It stores the patient's treatment plan. As it is common that some treatment operations have to be performed in prior to another, the set of treatment operations to be received by a patient has to be ordered to satisfy certain temporal constraints. Hence, each patient agent P_i maintains an ordered set $Tr^i = \{Tr^i_1, Tr^i_2, ...Tr^i_{N^i_\Gamma}\}$ as its treatment plan.

3.2 Resource Agent

A resource agent is used to manage a specific medical unit. Here, we denote R_{ab} as a resource agent representing medical unit $a \in A$ at cancer cluster $b \in \mathcal{C}$. Each resource agent has full access to the schedule of the medical unit it represents, but not the others.

3.3 Scheduling Algorithm

We adopt a two-phase scheduling algorithm similar to what being proposed in [6, 9]. For each newly diagnosed cancer patient, a treatment plan is first designed and then the corresponding treatment operations are initially scheduled (initial assignment phase). Then, a timeslot-swapping process is enforced for shortening the patient journey (rescheduling phase). Here we assume that any two patient agents are willing to exchange their timeslots as far as none of their schedules is worsen (as suggested in [9]) and none of the temporal constraints as specified in *Eq.* (2) is violated.[1] Algorithm 1 gives a high-level description of this two-phases scheduling algorithm.

Algorithm 1 Scheduling Algorithm

```
for every patient agent $P_i$ do
    Initial assignment based on $P_i$'s treatment plan
    for each $P_i$'s treatment operation do
        Rescheduled to be performed earlier by exchanging timeslot with another patient agent with the help of the resource agent (rescheduling phase)
        if No involving parties are worsened in terms of their resulting overall schedules then
            The exchanging process is proceeded
        end if
    end for
end for
```

4 AGENT COORDINATION

In this section, more details about the scheduling algorithm are given, including 1) how the patient agents interact with the resource agents, and 2) how some "unnecessary" swappings can be rejected so as to further improve the scheduling optimality.

4.1 A bidding process for agent matchmaking

Figure 2 shows our proposed framework. As what have been introduced in Sections 3.1 and 3.2, there are two types of agents, namely patient agents and resource agents. In order to show clearly the coordination between agents, we further categorize patient agents into *initiating patient agents* and *target patient agents*. Initiating patient agents P_I are those patient agents who initiate a request for timeslot exchange. Target patient agents P_G are the others who are willing to participate in the exchanging process.

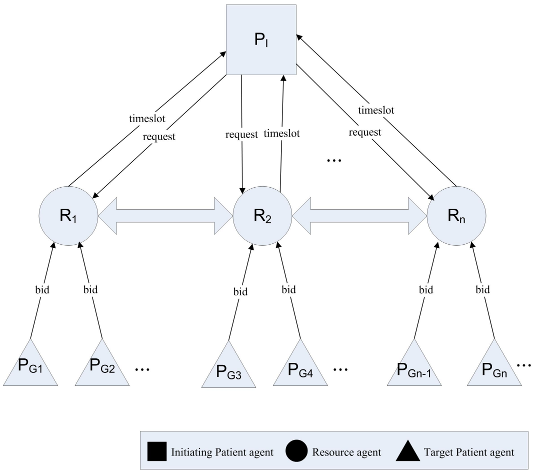


Figure 2. The proposed agent coordination framework for patient scheduling.

With the objective of shortening its patient journey, an initiating patient agent P_I first sends out a request for rescheduling to the corresponding resource agents R_{ab}. The request includes the earliest possible start date (*EPS*) and the latest possible start date (*LPS*) of its associated treatment operation. In order not to violate the temporal constraints between treatment operations, the *EPS* can be defined as:

$$EPS^I_j = D^I_{j-1} + T_{Tr^I_{j-1}} + \delta_1. \tag{5}$$

Note that δ_1 denotes how many days a patient should be admitted (if needed) before a treatment operation to be carried out. In our experiment, we set to be one. In practice, this value could be designated by healthcare providers in order to better suit their needs. With a similar argument, LPS is defined as:

$$LPS^I_j = D^I_j - 1. \tag{6}$$

Once a resource agent receives a request with EPS and LPS, it will first check whether there are available timeslots released by deceased patients which can fulfill the request. If yes, the released timeslot will be assigned to the initiating patient agent. If not, the resource agent will

[1] This assumption may imply that some policy-wise incentive has to be in place so that different medical units are willing to share their resources in this manner, which however is not the main focus of our study.

then pass the request to those patient agents (target patient agents, P_G) which reserved resources of the same type in the period from EPS to LPS. Those target patient agents who have received the request will submit a bid to the resource agent in response.

There are several factors needed to be considered in computing the bid value.

- First, the target patient agent should not have its last treatment operation in its treatment plan to be exchanged, or its last treatment operation has then to be performed later and thus it would end up with a lengthened patient journey.
- Second, as it is impractical to reschedule a patient's treatment operation without prior notification, we assume that the exchange of timeslots would not be considered if the initiating patient will have less than a week's time of notification.[2]
- Third, the target patient agent also has to ensure that the temporal constraints between its treatment operations would not be violated after the exchanging process.

Taking into account the above considerations, the bid value submitted by a target patient agent P_G is formulated as:

$$Bid^G = (D^G_{j_t} - EPS^I_{j_i}) + Last + Noti + Temp, \quad (7)$$

where $Last$, $Noti$ and $Temp$ are three binary variables. $Last = 0$ if the j_tth operation is not the last one for P_G, or ∞ otherwise. $Noti = 0$ if there is a week's time of notification for the target patient agent to be notified, or ∞ otherwise. $Temp = 0$ if there are no temporal constraints violated, or ∞ otherwise.

Among all the target patient agents, the one with the lowest bid value will be accepted and the timeslot swapping between the initiating agent and target agent will be proceeded. If two bids are found to be numerically identical, the resource agent will select one at random.

4.2 A coordination process for rejecting unnecessary swappings

A timeslot swapping as described in the previous section sometimes does not necessarily lead to ultimate improvement in patient journey. To illustrate that, suppose there is a patient agent with 3 treatment operations to be rescheduled. In case the last treatment operation could not be rescheduled to be performed earlier, any rescheduling of the first 2 treatment operations are essentially useless as the duration of the whole journey remains unchanged (see Figure 3(a)). As another example, even a shortened patient journey can be achieved, rescheduling of the first 2 treatment operations could also be useless if the rescheduling of the last treatment operation cannot be benefited from the rescheduling of the first two (see Figure 3(b)).

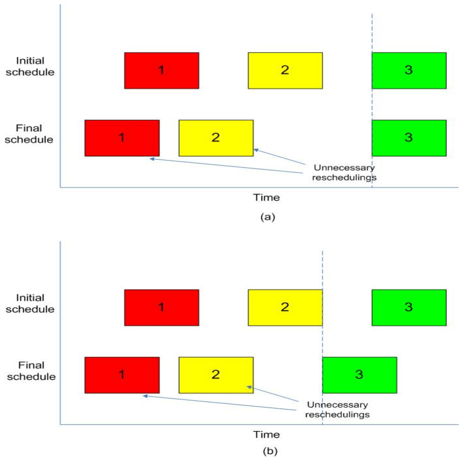


Figure 3. Unnecessary reschedulings.

In order that these useless swappings can be rejected so as to be reserved for other potentially more useful swappings, the scheduling algorithm could be modified in such a way that a resource agent after identifying the most optimal bid among the target patient agents will not notify the initiating patient agent immediately. Instead, it will pass the bid to the resource agent which is responsible for the succeeding treatment operation of the initiating patient agent. Having received such a bid, the resource agent could derive a new EPS, denoted as ${}^{(new)}EPS^I_{j_i+1}$. Clearly, unnecessary swappings occur if that resource agent could not find a bid among those received from the target patient agent P_G such that ${}^{(new)}EPS^I_{j_i+1} \leq D^G_{j'_t} \leq EPS^I_{j_i+1}$, where $Tr^I_{j_i+1} = Tr^G_{j'_t}$. In that case, the resource agent will notify its antecedent to discard the bid such that the corresponding timeslots would not be exchanged. In general, such a succeeding resource agent consultation process can be carried out in a recursive manner.

5 EXPERIMENTS

In order to evaluate the effectiveness of the proposed multi-agent framework, we first obtained a dataset containing the scheduled treatment plans of 4720 cancer patients being treated at the seven cancer clusters in Hong Kong with a diagnosis period spanning 6 months (from 1/7/2007 to 31/12/2007). The average length of the patient journey among all cancer clusters is 82.4 days. Based on the dataset,

[2] In general, the time of notification can be adjusted according to the real situation.

we have carried out simulations with the following 4 experimental settings:

Setting 1: Patient agents are willing to exchange timeslots with others whenever there is a Pareto improvement.

Setting 2: It is assumed that only 20% of the patients of each cancer cluster are allowed to undergo timeslot swapping.

Setting 3: It is assumed that patients are reluctant to travel for a long distance even though some of their operations can be scheduled earlier, and thus only swappings between two nearby cancer clusters are allowed. In particular, the neighborhood relationships are assumed to be

- $C_1 \rightarrow C_2$ or C_3
- $C_2 \rightarrow C_1$ or C_3
- $C_3 \rightarrow C_1$ or C_2 or C_4 or C_5 or C_6
- $C_4 \rightarrow C_3$ or C_6
- $C_5 \rightarrow C_3$ or C_7
- $C_6 \rightarrow C_3$ or C_4
- $C_7 \rightarrow C_5$

where $\alpha \rightarrow \beta$ implies that patients admitted in cancer cluster α would only be swapped to its neighboring cancer cluster β.

Setting 4: Timeslots released by deceased patients are allocated to the patient agents who have the longest patient journeys at a time point.

Given the four aforementioned settings, Figure 4 shows the average length of patient journey among the seven cancer clusters in Hong Kong.

The experimental results obtained show that, on average, the average length of journey can be reduced by 6.0 days for those 4720 cancer patients if no restriction is imposed on the exchange of timeslots whenever there is a Pareto improvement (Setting 1). Given only 20% of patients per cancer cluster are allowed for timeslot exchange (Setting 2), we found that the average length of journey could still be reduced by an average of 3.4 days. With the geographical restriction on allowing only swappings between nearby cancer clusters (Setting 3), the average length of journey can also be reduced by 4.4 days.

However, it should also be noted that according to Figure 5, the maximum length of journey remains unchanged. The reason is obvious as no one is willing to swap with those with the longest length of journey. Reductions on the maximum length of patient journey can only be observed for Setting 4 where the released timeslots due to deceased patients are allocated to those with the longest patient journey.

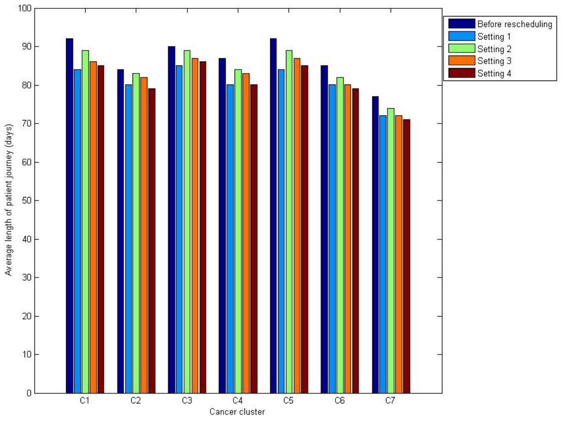


Figure 4. Average length of patient journey among the seven cancer clusters under 4 different settings.

5.1 Simulations revealing the impacts of varying the unit capacities

For all the results presented so far, it is assumed that the capacity of each medical unit is fixed. To study the cost-effectiveness of increasing the capacities of medical units for patient journey optimization, we had conducted several simulations in which timeslots were additionally allocated to each medical unit. Particularly, 3 different timeslot-allocation strategies were performed in our simulations:

1. Timeslots were added on a *daily* basis. In our simulation, 2 timeslots were added daily to each medical unit.

2. Timeslots were added on a *weekly* basis. In our simulation, 14 timeslots were added weekly to each medical unit.

3. Timeslots were added on a *monthly* basis. In our simulation, 60 timeslots were added monthly to each medical unit.

It is worth to note that, let say, if timeslots were added on a weekly basis, cancer patients would then be scheduled on a weekly basis too (i.e. a cancer patient would be scheduled to receive treatment operation in a certain week if the corresponding weekly capacity does not exceed its limit).

By Figure 6, it was found that when the timeslot-allocation strategy was changed from a daily basis to a weekly basis, and then to a monthly basis; the utilization of medical units will increase subsequently. The reason is that when the medical resources were allocated in a more

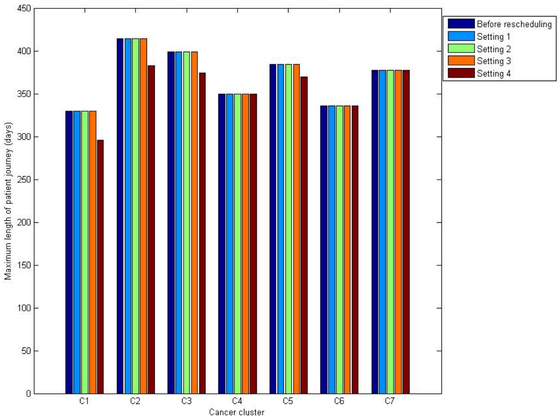


Figure 5. Maximum length of patient journey among the seven cancer clusters under 4 different settings.

flexible way (i.e. with a wider time frame considered), the possibility of assigning a cancer patient to receive treatment operation on a particular date will increase as a result.

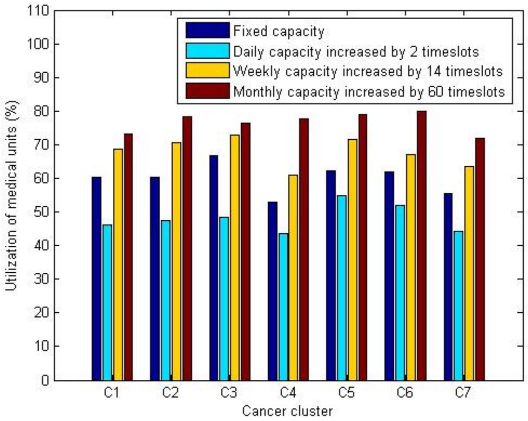


Figure 6. Utilization of medical units by varying the timeslot-allocation strategies.

Meanwhile, it was also observed that, both the average and maximum length of patient journey will drop significantly when additional timeslots were allocated to each medical unit. In particular, such significant drop can be found before (see Figure 7 and Figure 8) and after (see Figure 9 and Figure 10) the rescheduling phase. In fact, among the 3 timeslot-allocation strategies, we found that a monthly-basis strategy is the most effective one in achieving an optimal patient journey.

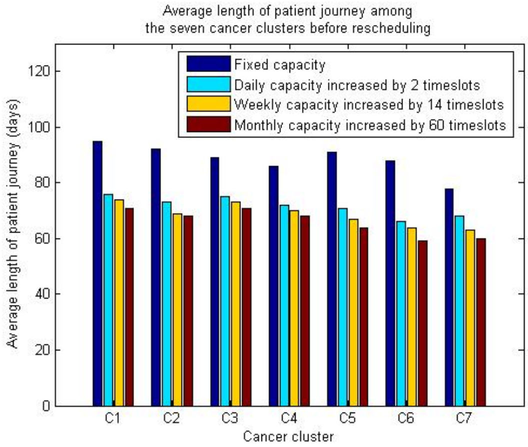


Figure 7. Average length of patient journey among the seven cancer clusters by varying the timeslot-allocation strategies (before rescheduling).

5.2 Simulations revealing the impacts of reducing the durations between treatment operations

During our simulations, we had also tried to investigate the impacts induced by the durations between treatment operations. At this time, while all the unit capacities are fixed, we reduced all the treatment durations by half.

Interestingly, it was observed that when the treatment durations were reduced by half, a significant reduction in both the average and maximum length of patient journey can also be achieved before (see Figure 11 and Figure 12) and after (see Figure 13 and Figure 14) the rescheduling phase. With such insight, we got an important implication: whenever the durations between treatment operations can be minimized in accordance with medical reasons, cancer patients could then enjoy themselves with less undesired waiting times. In other words, during the practical implementation, it is important for the healthcare provider to carefully and unbiasedly quantify such durations between treatment operations.

6 CONCLUSIONS

In this paper, a multi-agent framework was proposed for patient journey optimization. Particularly, by applying the framework, the shortening of a patient journey will not lengthen the journeys of the others. Also, all the temporal constraints among the treatment operations for each patient would not be violated during the scheduling process.

The effectiveness of the proposed framework has been demonstrated by applying it to a dataset containing 4720 scheduled treatment plans of cancer patients admitted to

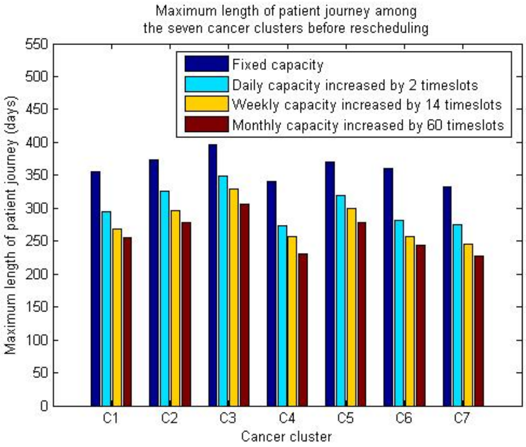


Figure 8. Maximum length of patient journey among the seven cancer clusters by varying the timeslot-allocation strategies (before rescheduling).

hospitals in Hong Kong. The effects of varying the unit capacities and treatment durations on the overall reduction in length of patient journey are also studied.

Because of the limited resources during practical implementation; in the near future, rather than routinely allocate a fixed amount of additional timeslots to each cancer cluster as what had been demonstrated earlier, we are going to assess how resources (or timeslots) should be allocated to cancer clusters in a more efficient and unbiased way such that the overall patient journey could be shortened in a greater extent.

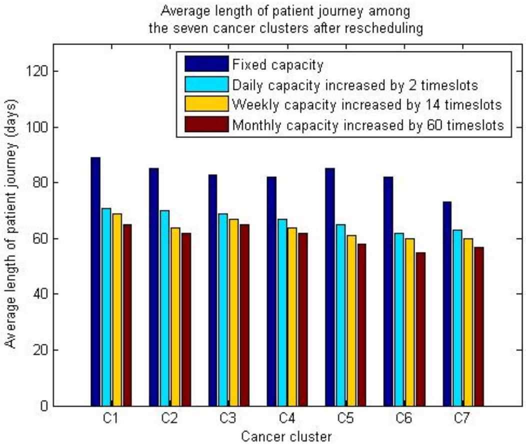


Figure 9. Average length of patient journey among the seven cancer clusters by varying the timeslot-allocation strategies (after rescheduling).

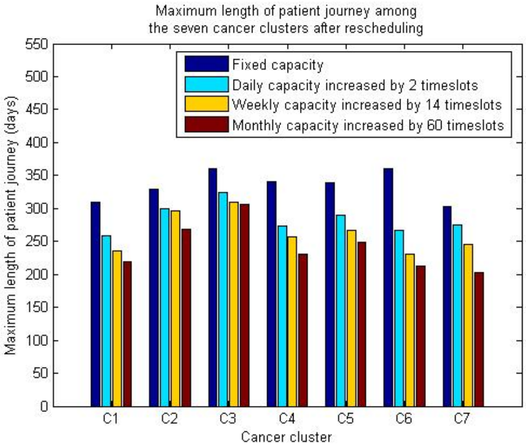


Figure 10. Maximum length of patient journey among the seven cancer clusters by varying the timeslot-allocation strategies (after rescheduling).

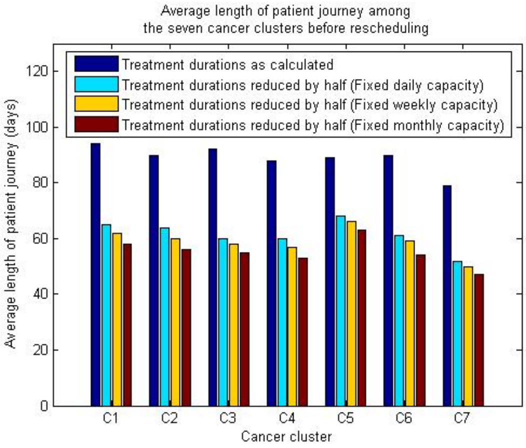


Figure 11. Average length of patient journey among the seven cancer clusters by reducing the treatment durations by half (before rescheduling).

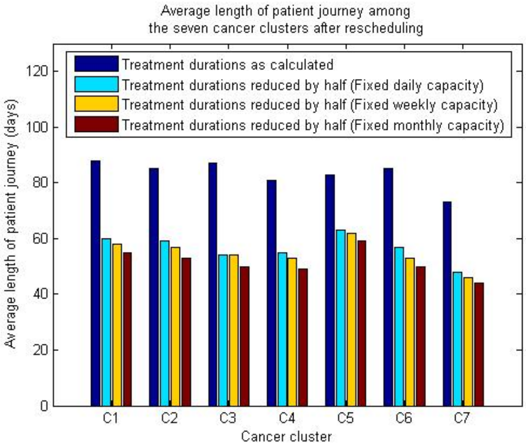


Figure 13. Average length of patient journey among the seven cancer clusters by reducing the treatment durations by half (after rescheduling).

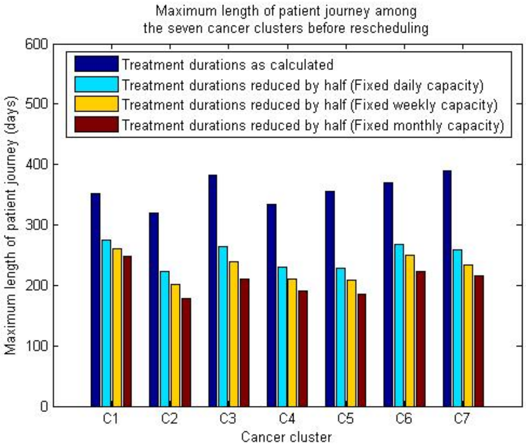


Figure 12. Maximum length of patient journey among the seven cancer clusters by reducing the treatment durations by half (before rescheduling).

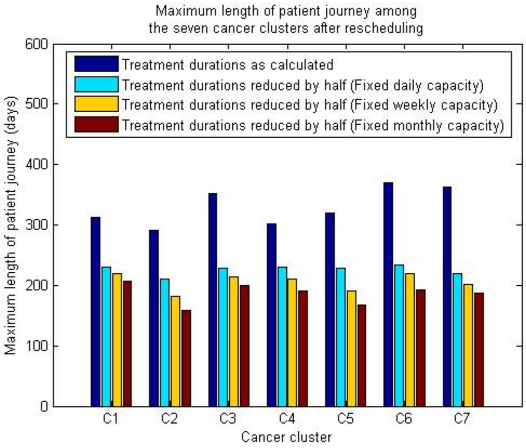


Figure 14. Maximum length of patient journey among the seven cancer clusters by reducing the treatment durations by half (after rescheduling).

ACKNOWLEDGEMENT

This is to acknowledge Hospital Authority in Hong Kong for providing the dataset to support this study.

References

[1] H. Czap and M. Becker. Multi-agent systems and microeconomic theory: A negotiation approach to solve scheduling problems in high dynamic environments. In *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 3*, page 83.2, Washington, DC, USA, 2003. IEEE Computer Society.

[2] K. Decker, J. Li, and Y. Demazeau. Coordinating mutually exclusive resources using gpgp. *Autonomous Agents and Multi-Agent Systems*, 3:200–0, 2000.

[3] Department of Health, UK. Now I feel tall: What a patient-led NHS feels like. 2005.

[4] X. Mao, A. Mors, N. Roos, and C. Witteveen. Coordinating competitive agents in dynamic airport resource scheduling. In *MATES '07: Proceedings of the 5th German conference on Multiagent System Technologies*, pages 133–144, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] J. Patrick and M. Puterman. Reducing wait times through operations research: optimizing the use of surge capacity. *Healthc Q*, 11(3):77–83, 2008.

[6] T. O. Paulussen, I. S. Dept, K. S. Decker, A. Heinzl, and N. R. Jennings. Distributed patient scheduling in hospitals. In *Coordination and Agent Technology in Value Networks. GITO*, pages 1224–1232. Morgan Kaufmann, 2003.

[7] V. Robu, H. Noot, H. La Poutré, and W.-J. van Schijndel. An interactive platform for auction-based allocation of loads in transportation logistics. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 3–10, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[8] M. Simunovic, A. Gagliardi, D. McCready, A. Coates, M. Levine, and D. DePetrillo. A snapshot of waiting times for cancer surgery provided by surgeons affiliated with regional cancer centres in Ontario. *CMAJ*, 165(4):421–425, August 2001.

[9] I. Vermeulen, S. Bohte, K. Somefun, and H. La Poutre. Improving patient activity schedules by multi-agent pareto appointment exchanging. In *CEC-EEE '06: Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, page 9, Washington, DC, USA, 2006. IEEE Computer Society.

[10] J. Vissers and R. Beech. *Health operations management : patient flow logistics in health care*. Routledge, 2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN, 2005.

[11] J. Vissers, J. Bekkers, and I. Adan. Patient mix optimization in tactical cardiothoracic surgery planning: a case study. *IMA Journal of Management Mathematics*, 16, 2005.

[12] M. Wang, J. Liu, H. Wang, W. K. Cheung, and X. Xie. On-demand e-supply chain integration: A multi-agent constraint-based approach. *Expert Systems with Applications*, 34(4):2683 – 2692, 2008.

[13] G. Weiss. *Multiagent systems : a modern approach to distributed artificial intelligence*. Cambridge, Mass. : MIT Press, 1999.

Methods of Video Object Segmentation in Compressed Domain

Cheng Quan Jia

Abstract

Traditional video object segmentation methods operate on the pixel domain, which require every frame in the video sequence to be decoded into raw data. This incurs additional processing and storage overhead which is unfavourable in real-time application. Recently, video object segmentation in the Compressed Domain, i.e. video compressed using Motion Compensation, Discrete Cosine Transform(DCT) and Quantization, have gained attention because it only requires the compressed video to be parsed to obtain the motion vectors and DCT coefficients for segmentation. This paper outlines the features present in the Compressed Domain that can be used in video object segmentation, segmentation methods published by recent researchers, and possible research areas in the future.

1 Introduction

Traditional video object segmentation is performed in the pixel domain, in which pixel data are obtain from full decoding of the video bitstream. The motion flow is extracted by comparing consecutive frames and the basic data used is intensity value from each pixel. Although the traditional approaches have reached certain maturity and give reasonable segmentation results, the processing and storage overhead in decoding every frame from a video sequence prevents these methods from application in real-time applications. Recently, Compressed Domain segmentation have gained attention not only because it does not require full decoding of the compressed video, but the motion information that are already present in the Compressed Domain could help video object segmentation.

The term Compressed Domain in literature refers to video compression methods in which motion compensation and Discrete Cosine Transform (DCT) are used to reduce the number of bits required to represent a video, examples include MPEG-1/-2/-4, H.261 and H.263. All of these compression standards achieve compression by exploiting two observations. Firstly, it is unusual for intensity values to change frequently over a small area(spatial redundancy). Secondly, consecutive frames along time-ordered sequence of frames are similar(temporal redundancy). The Compressed Domain address the first observation with DCT and Quantization and the second with Motion Compensation, both of which are described in the following two sections.

1.1 Intraframe Compression

Suppose we have a picture frame. Based on the first observation, the frame is divided into 8×8 macroblocks. Then DCT is performed on each macroblock. Discrete Cosine Transform turns the intensity values in a macroblock into representation of sums of cosine functions oscillating at different frequencies. The transform used in video compression is a two-dimensional one, which transforms the array of spatial intensity values to an array of DCT coefficients, each coefficient denoting the value of vertical and horizontal frequencies, i.e. frequencies of intensity change, in the macroblock. The DCT coefficients are arranged in the array in a way that the DC coefficient(constant) lies at the top-left element of the array and elements further to the right and down contains AC coefficients of cosine functions with higher horizontal and vertical frequencies respectively.

Quantization is employed on each DCT-ed macroblock in order to remove the high spatial frequency components. Each coefficient in the macroblock is quantized by a rounded division with a quantization value. To achieve bias against high-frequency components, a quantization table with higher quantization values towards the lower right corner is used. The resulting macroblock usually has the high-frequency components discarded. Finally, the DCT coefficients are subjected to Entropy Coding.

1.2 Motion Compensation

According to the second observation, temporal redundancy exists between consecutive frames in a video sequence. Even more bits can be saved using Motion Compensation rather than encoding frames as a whole image (as in Intraframe Compression). Again, the current frame is divided into macroblocks. Each macroblock is compared against the reference frame (a reconstructed frame previous to the current frame) within a small neighbourhood search window for the best match, i.e. the position with the least difference between current and reference macroblock. The result is a predicted motion vector and a predicted macroblock from the reference frame. Since the best match may not be identical to the current macroblock, the difference between the current macroblock and the predicted macroblock gives a difference macroblock to denote the prediction error. The difference macroblock undergoes DCT and Quantization. Finally the motion vector and the difference macroblock undergo Entropy Coding.

Note that to avoid propagation of error, the I-frame is sent after a number of P-frames. The forms Group of Pictures(GOP) structures of IPPP... frames. Also, MPEG standards employ Bidirectional Motion Compensation in addition to forward prediction. Another type of frame, the B-frame, is introduced in addition to the P-frame so that the GOP structure becomes IBB...PBB...PB... A B-frame is constructed by predicting from its previous I/P-frame and its next I-/P-frame such that two sets of motion vector and predicted macroblock are found. Both the predicted macroblock and motion vectors are averaged and compared against the current frame to generate the difference macroblock.

1.3 Features for Motion Segmentation

Since image change detection could be seen as a classification problem [7], this section lists the features available in the Compressed Domain. While full decoding is not required in the Compressed Domain, the motion vector(in interframes) and DCT coefficients(of image data in intraframes and difference image in interframes) are required for video object tracking. Parsing, in which the input binary bitstream is first entropy decoded then inverse quantized, extracts these necessary information [6]. Therefore, the information available in the Compressed Domain is the DCT coefficients of the picture macroblocks in the intraframes and difference macroblocks in the interframes, and motion vectors associated with the predictive macroblocks. Of particular importance is the motion vectors and top row and and left column DCT coefficients in the I-frames that denotes vertical and horizontal spatial frequency respectively [5, 3].

Note that the motion vectors mentioned above are generated for the best match in the reference frame rather than generated to denote video object motion. Including motion vectors that are uncorrelated to true motion degrades segmentation accuracy, which is discussed in the next section. Several assumptions should hold for the following discussion. Firstly, smooth and relatively small motion should exist in the input video otherwise most of the macroblocks would be intracoded instead of intercoded and we would have few motion vectors to work with. Secondly, the motion in the input video should span over more than one Group of Pictures.

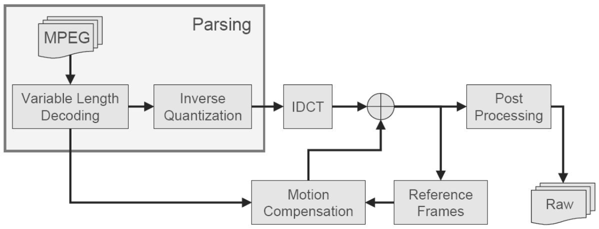


Figure 1. Parsing from MEPG bitstream[6]

2 Related Work

2.1 Video object segmentation using motion vectors

Since motion information in a video sequence is critical in video object segmentation, sparse motion vectors present in predictive macroblocks are used by many researchers for video object segmentation. Their methods involve constructing dense motion field by accumulating sparse motion vectors or obtaining average motion vectors over a series of P- or B-frames. As the motion vectors in encoded video sequences do not reflect object motion, outliers do occur. Many segmentation algorithms([1], [5], [3]) remedies this by median filtering or other methods.

Liu et al. [5] uses motion vectors to locate coarse video object regions then DCT coefficients as spatial feature to identify similar macroblocks. Firstly, the motion vectors are accumulated and median-filtered to obtain a motion field. Secondly, based on the motion field, the macroblocks in the object edge area of non-zero motion vectors are rectified by similarities of of DCT blocks. Rectification removes pseudo moving blocks(blocks out of a video object but within non-zero motion field) from the segmentation area and include pseudo still blocks(blocks within a video object but out of non-zero motion field) in the segmented video area and uses the belief that pseudo moving blocks and pseudo still blocks appear mainly in edge areas to aid rectification. It selects the moving blocks that appears in the edge area as the center moving block and finds similar blocks in the four-neighbourhood, if they do not exist then the center moving block is deemed pseudo moving block and be discarded from segmentation area. Otherwise, for each still block from the four-neighbourbood of blocks, calculate the distance between the features(the mean, horizontal edge, vertical edge and diagonal edge) of the center moving block and the chosen moving block and the distance between the center moving block and the neighbouring moving blocks; if the still block is closer to the center moving block than the neighbouring moving blocks, it is deemed as a pseudo still block and be included in the segmentation area.

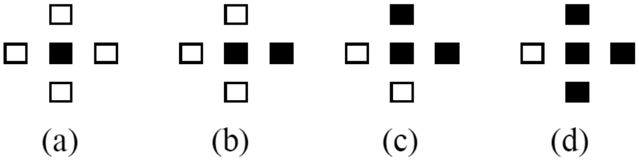


Figure 2. Four cases of edge area (a)Four still blocks around a moving block (b)Three still blocks around a moving block (c)Two still blocks around a moving block (d)One still block beside a moving block[5]

The experimental results of [5] show that the segmentation masks, while envelop the desired video objects, tend to be a few macroblocks wider than ground truth. The reason behind this is the inclusion of similar macroblocks(in terms of DCT coefficients) as pseudo still blocks.

Hariharakrishnan and Schonfeld [4] used only motion vectors as their segmentation feature. In [4], an adaptive block matching algorithm is used. From the initial object mask, macroblocks that lie entirely in the object are chosen as seed motion blocks when the other macroblocks in the object mask are labeled uncertain blocks. Then, from every three frames before the current frame, backward motion estimation is performed so that the average motion between consecutive frames is obtained. In case the average motion is higher than a threshold, motion is estimated from the previous frame instead. Then the object mask for the current frame is derived by motion compensated from the motion obtained in the previous step, and detect for occlusion and disocclusion. For disocclusion detection, the regions that will be uncovered in the current frame is estimated from previous frames using motion compensation. The uncovered regions are tested against the object for motion consistency, by first clustering motion vectors in the region using k-means clustering then comparing it with the motion vector of the object. If the difference between the two motion vectors is smaller than a threshold the uncovered region is treated as disocclusion and included in the object. [4] views occlusion and disocclusion as dual events, therefore occlusion detection is similar to disocclusion, except covered regions are obtained from the next frame and are tested for motion dissimilarity.

Yokoyama et al. [10] uses Vector-featured Images obtained in the MPEG sequence to discover and track moving video objects. In each image, macroblocks are classified into five types: the Current Block, the Reference Block, the Background Block, the Moving Block and the Unmoving Block. The Reference Block is a macroblock in the I-frame associated to an initial point of a motion vector; the Current Block is a macroblock associated to the end of a motion vector; the Moving Block is created at the overlap of Current Block and Reference Block and indicates a moving region; the Unmoving Block serves to keep track of regions that momentarily stop. An Unmoving Block is created when a Current Block generates zero motion, and decrease its brightness on successive frames, until its maximum life expires. When the object resumes motion, if the Moving Blocks overlaps with the Unmoving Blocks, the Unmoving Blocks are updated to Moving Blocks. Under this scheme, motion is detected if from the Moving Blocks if the Moving Region is large enough, otherwise the union of Moving and Unmoving Blocks is used instead. The moving object candidate is compared with previously registered objects to achieve object tracking.

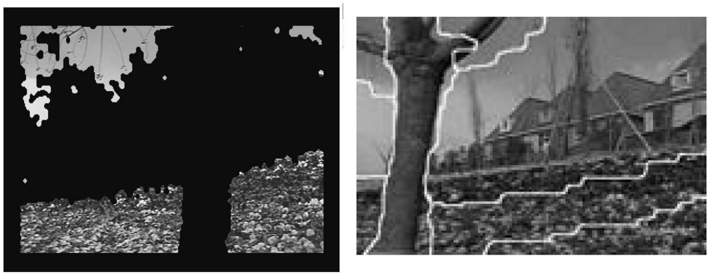


Figure 3. Extracted flower bed object from [1] (above) and segmentation result from [2] (below). A portion of the sky is included in their segmentation masks.

It is observed from the experimental results of [1] and [2] that the segmentation masks tend to cover areas other than the video objects. This is more prominent in areas with similar texture. It is because the encoder looks for the best match in the block matching process rather than true object motion. In addition, the motion field in P-frames, due to larger temporal distance to reference frames, are less reliable [8]. Therefore, contaminated with erroneous motion vectors, the aggregated motion field would not give a correct boundary. The system in [4] avoids this slightly since it has an initial segmentation mask to work with(it uses a four-band multi-valued segmentation followed by a lattice partition operator) and it switches to calculating the motion from the next frame when the average motion over the next three frames is higher than a threshold, and [10] imposes a minimum bounding rectangle to prevent inclusion of uncorrelated motion. Still, there should be some form of confidence measure that ensure a motion vector is approximated to real object motion, which is addressed in the following subsections.

2.2 Porikli et al.'s investigation in the Compressed Domain

The previous assertion that motion vectors alone do not suffice in accurate video object segmentation finds ground in an article by Porikili et al[6]. They proposed a video object segmentation system that experimented with almost all of the information present in the Compressed. The system uses the DCT coefficients in the I-frame motion vectors of P-frames in a Group of Pictures to construct Frequency-Temporal(FT) data structures for each macroblock in the GOP. The FT data consists of the following:

- The DC parameters(for Y, U, V channels) of the I-frame
- A subset of low vertical and horizontal frequency AC values
- A spatial energy term(total magnitude of AC coefficients) measuring spatial variance
- Aggregated motion flow of the corresponding macroblock

The aggregated motion flow of the macroblock is the mean of aggregated pixel motion vectors, which is obtained by interpolation of the filtered motion vector in a macroblock for all P-frames in a GOP then back-propagate from the last frame to the first. Porikli et. al [6] includes two segmentation approaches in their article. One uses FT volume growing, in which macroblock with the lowest local variance(derived from the energy in the local spatial and temporal neighbourhood) is chosen as the seed block and the volume is grown in both 2D spatial and temporal dimensions. The other employs Multi-Kernel Mean-Shift Segmentation, one in spacial-temporal dimension, one in aggregated motion vector space, one in DCT coefficient space. The process shifts these kernels at the same time computes their gradient, until their sink points are found. It goes on linking sink points closer than than a preset value from each other in the joint domain, to form clusters of sink points. The points in the clusters constitutes a segmentation volume. Both approaches would have volumes of negligible size removed and remaining volumes inflated. Finally, hierarchical clustering is performed and a partition tree is built by iteratively merging pairs of most similar volumes.

Porikli et. al's experimental results show that both segmentation approaches produce similar results. Also, a slight over segmentation using DCT coefficients followed by aggregated motion based clustering produces more accurate boundaries than single stage joint segmentation. They attribute this to the fact that motion boundaries tend to be deformed and erroneous. Also, using all of the DCT coefficients do not necessarily provide a stable segmentation in that the mean-shift algorithm becomes sensitive when AC components and spatial energy term are included. Note that the best combination stated above renders the system to segment video objects with similar average intensity value and texture, which in turn sensitive to intensity differences; in addition, the proposed algorithm favours moderate motion since spatial-temporal volumes would be disjoint in the presence of large motion.

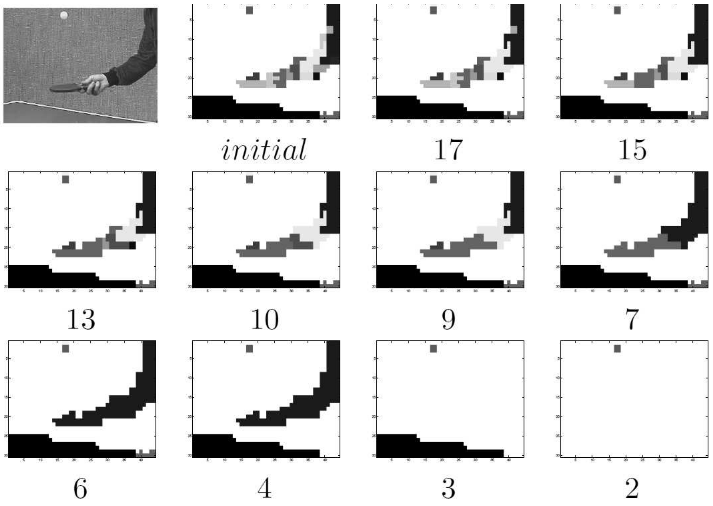


Figure 4. Porikli et al.'s segmentation results at the corresponding clustering levels. Note the volume growing process could not blend the lower part of the arm into other regions since its DCT coefficients were also significantly different.[6]

2.3 Approximation of Optical Flow and confidence measure

Coimbra [3] tries to approximate optical flow from motion vectors and DCT coefficients in MPEG-2 Compressed Domain, in comparison to the Lucas-Kanade algorithm. It reasoned that the optical flow and eigenvalues for confidence measure is parallel to motion vectors and AC coefficients found in Compressed Domain respectively. The proposed method obtains a smooth dense motion field from a Group of Pictures, in which motion vector magnitude is normalized and motion vector from the previous macroblock is interpolated to macroblocks that have no motion information. The motion obtained from this method is smoothed by median filtering to remove isolated motion vectors. Then the first vertical AC coefficient and the first horizontal AC coefficient (AC[1] and AC[8] in a macroblock) in the I-frame macroblocks is used as a confidence map. The confidence update step will have a 8×8 macroblock referencing a 16×16 image block in the I-frame, and the confidence of the motion vector of the macroblock is the weighted average of confidence in the 16×16 window. The optical flow estimation technique in [3] is employed in [9] to estimate motion for each macroblock in a video sequence to achieve action recognition and localization.

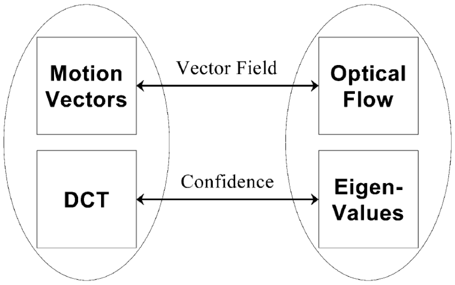


Figure 5. Comparison of the LK and MPEG-2 system[3]

The experimental results in [3] clearly demonstrates the effects of employing confidence measure on the obtained motion field. The motion vectors after harmonization step and median filtering have motion information uncorrelated to object displacement caused by illumination noise and aperture problem. After the confidence update step motion vectors with high confidence measure(usually those at boundary areas) are kept.

2.4 Pixel precision edge refinement in raw domain

While they are not working entirely in the Compressed Domain, the systems proposed by Babu et al. [1] and Chen and Bajic [2] can further refine their coarse segmentation results to achieve segmentation with pixel precision. Both algorithms first identify the edge macroblocks within the segmented area, then perform edge refinery to obtain the precise object boundaries. Both algorithm requires the edge macroblocks to be fully decoded in the edge refinement step. The algorithms still perform faster than raw domain segmentation algorithms, despite the fact that full decoding is performed the blocks.

The algorithm in [1] performs coarse segmentation by first obtaining a dense motion field by accumulating motion vectors over a few frames forward and backward then applying a 2D median filter and a Gaussian filter. Only reliable motion vectors i.e. vectors that correspond to macroblocks that have total DCT error energy less than a threshold is used in the process. If the macroblock is unreliable or intracoded, motion corresponding to the block is interpolated from the neighbouring blocks. Next it uses the Expectation Maximization(EM) Algorithm to compute the likelihood of each pixel to a number affine motion models, which is determined using k-means clustering. The result of the EM Algorithm is the coarse segmentation result. Edge refinement in [1] involves decoding the edge blocks and their eight neighbourhood, computing representative motion vectors for the block, defining a search range based on the motion vector, and matching a small pixel block against the previous frame within the search region.

Unlike [1], [2] obtains dense motion field using Motion Vector Integration, in which coherent motion integration(backward summing up of normalized MVs over frames) and incoherent motion integration(adding of MV magnitude over successive coherent integrations) are combined. Coarse segmentation is achieved using k-means clustering and blocks are identified as a boundary block if the block's eight neighbourhood are from more than one region. [2]'s edge refinement involves Pixel-based MV Integration, in which the pixels in a macroblock acquire the MV of their corresponding macroblock in the previous frame, to interpolate MV into boundary regions in order to increase their MV density, Canny Edge detector and morphological operations to obtain object boundary in pixel accuracy.

3 Observations

Several observations can be obtained from the above segmentation methods. To use motion vectors as a suitable indiction of object motion, a dense motion field should be constructed from the sparse motion vectors present in inter-coded macroblocks. [1], [2], [5], [6] and [3] either use motion accumulation or use motion aggregation to deduce the resulting motion field over frames. In addition, [1], [5], [6] and [3] clearly state that Motion Compensation in the encoding process only gives motion vectors of the best match macroblock but not the actual motion in the video sequence. This is particularly prominent when compensating large objects with similar interior texture. Also, the Motion Compensation step is suspectable to aperture problem as the process concerns only with displacement of texture in the macroblock. While median filtering removes outlying motion vectors, the filtering is also blind. The confidence measure introduced in [3] ensures to some degree that the motion vectors in macroblocks with high vertical and horizontal edge energy is trustable, and [1] chooses the motion vectors of inter-coded macroblocks that has DCT error energy lower than a threshold to be reliable motion vectors. For macroblocks that are intra-coded, motion vectors are interpolated to them from negihbouring regions.

While motion accumulation, motion vector interpolation and filtering guarantee a dense motion field, repetitive motion over large sequences of frames leads to motion cancelation. [2] migitate this by summing motion vector magnitudes over a series of frames. The segmentation method in [10] does not face this problem since it uses vector images to memorize object trajectory, but loses the benefit of filtering out unreliable motion vectors.

In addition to motion vectors, DCT coefficients from both intra-coded macroblocks and difference macroblocks are chosen as a segmentation feature in some aforementioned methods. [5] uses DC the component and AC edge energy as features for calculation of distance between macroblocks, and [6] uses DC component, AC edge energy and spatial energy to construct a feature vector for each macroblock. Note that the DC coefficient (in both intra-coded and inter-coded macroblocks) is not used as a determining feature since DC coefficient denotes average energy over an 8×8 block. Also, the AC coefficients used in [3], i.e. the first horizontal and vertical AC coefficients, are convincing measure of edge strength and can be used to identify object boundaries.

4 Conclusion

In this paper, we have introduced some recent methods on video object segmentation in the Compressed Domain. Compressed Domain video object segmentation involves first obtaining the motion vectors(in inter-coded frames) and DCT coefficients of macroblocks from the input video sequence by parsing then constructing object areas or boundaries from a dense motion field. To summarize the methods listed in this paper, a Compressed Domain motion segmentation should perform the following: upon receiving the parsed input video sequence, motion is accumulated over several frames so that dense motion are captured, motion vectors that are associated with a macroblock with error higher than a threshold is discarded and motion vector is interpolated from the block's neighbours [1]; the accumulated motion is subjected to confidence thresholding using confidence map obtained by [3]; moving objects are segmented from the initial frame, either by clustering macroblocks into motion models [1] or identify macroblocks that has high horizontal or vertical AC energy as edge block candidates then rectify the boundary by removing pseudo moving/still macroblocks [5]; the coarse object mask is detected for occlusion/disoccusion by discovering covering/uncovering regions and test for motion consistency/inconsistency [4]; finally edge refinement is performed on the coarse segmentation mask, in which the edge blocks and their eight neighbourhood are decoded, computing representative motion vectors for the block, defining a search range based on the motion vector, and matching a small pixel block against the previous frame within the search region [1].

While current Compressed Domain technique greatly reduce the processing and storage complexity of segmentation, due to fact that the block matching process in video compression is sensitive to changes of intensity values and problems associated with optical flow such as the aperture problem. Also not discussed in this paper are method to estimate motion in the presence of camera motion and video object segmentation in the presence of scene cuts. Both are interesting problems that worth investigation.

References

[1] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan. Video Object Segmentation: A Compressed Domain Approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):462–473, April 2004.

[2] Y.-M. Chen and I. V. Bajic. Compressed-Domain Moving Region Segmentation with Pixel Precision using Motion Integration. In *IEEE Pacific Rim Conference on Computers and Signal Processing, 2009*, pages 442 – 447, August 2009.

[3] M. T. Coimbra and M. Davies. Approximating Optical Flow Within the MPEG-2 Compressed Domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):103–107, January 2005.

[4] K. Hariharakrishnan and D. Schonfeld. Fast Object Tracking Using Adaptive Block Matching. *IEEE Transactions on Multimedia*, 7(5):853–859, October 2005.

[5] L. Long, F. Xingle, J. Ruirui, and D. Yi. A Moving Object Segmentation in MPEG Compressed Domain Based on Motion Vectors and DCT Coefficients. In *Congress on Image and Signal Processing, 2008*, volume 3, pages 605–609, May 2008.

[6] F. Porikli, F. Bashir, and H. Sun. Compressed Domain Video Object Segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(1):2–14, January 2010.

[7] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image Change Detection Algorithms: A Systematic Survey. *IEEE Transactions on Image Processing*, 14(3):294–307, March 2005.

[8] R. Wang, H.-J. Zhung, and Y.-Q. Zhang. A confidence measure based moving object extraction system built for compressed domain. In *The 2000 IEEE International Symposium on Circuits and Systems, 2000*, volume 5, 2000.

[9] C. Yeo, P. Ahammad, K. Ramchandran, and S. S. Sastry. High-Speed Action Recognition and Localization in Compressed Domain Videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1006–1015, August 2008.

[10] T. Yokoyama, T. Iwasaki, and T. Watanabe. Motion Vector Based Moving Object Detection and Tracking in the MPEG Compressed Domain. In *Seventh International Workshop on Content-Based Multimedia Indexing, 2009*, pages 201–206, June 2009.

Secure proximity detection

LI HONG PING
Hong Kong Baptist University
Kowloon Tong, Hong Kong

hpli@comp.hkbu.edu.hk

ABSTRACT

Nowadays, there is increasing number of mobile devices equipped with positioning capabilities (e.g. GPS), which ask location-dependent queries to Location Based Services (*LBS*). Normally, users expect LBS can help them find out the nearest point of interest and also share their location information through the social network web-site. (e.g. Twitter) Therefore, this raises a serious concern on location privacy of client. Previously, there are papers suggest number of methods (e.g. using dummies, cloaking, encryption) to preserve the location privacy of the client, while finishing the given mission by the user. In this paper, we will summarize the papers which are discussing this topic and propose a meaningful suggestion in the proximity detection area.

1. INTRODUCTION

More and more communication devices have implemented the user-positioning functions (e.g.GPS). Users can find out their location, by issuing location-dependent queries, for instance "find the nearest hospital" through the Location Based Services (LBS), which is provided by LBS provider (e.g. Google Maps)

Nevertheless, queries may disclose sensitive information about individuals, including user location, lifestyle and habits. Therefore people start to worry with the personal privacy, while they are using the Location Based Services (LBS).

Probably, there are three type of techniques can be used for protecting location privacy, including cloaking, dummy and encryption.

For the cloaking aspect, its main idea is to reduce the spatiotemporal resolution of user location. So the real user location is replaced by the cloaking region, in order to hide the user's location.

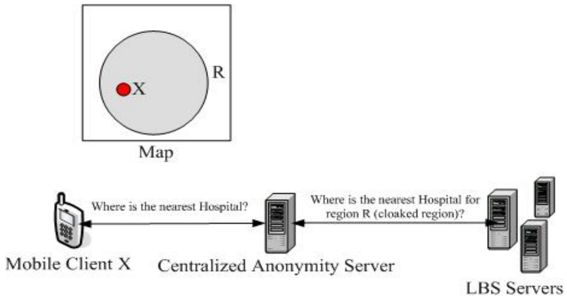

X
R
Map
Where is the nearest Hospital?
Where is the nearest Hospital for region R (cloaked region)?
Mobile Client X
Centralized Anonymity Server
LBS Servers

Fig. 1 Description of location cloaking

As Fig 1 show that, if mobile client X in location X, then he will sent out his location and convert into a cloaking region R(by himself or centralized anonymity server).Then send the query of region R to the Server. The concept of location k-anonymity was introduced in [9] where k is set to be uniform for all users. It focuses on the field of location privacy for mobile users through spatial and temporal cloaking of location and time information. We will give a further discussion in Section 2.

For the dummy aspect, the concept of this technique is user send her real location mixed with few dummy locations so that the attacker cannot tell which one is real.

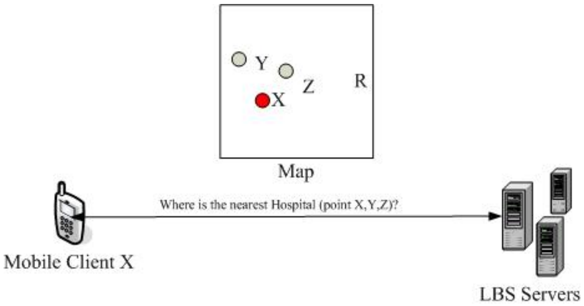


Fig. 2 Description of generating faked dummies

For example Fig. 2 show that when client X send his location service request to the LBS server, he will send out the faked dummies (Y,Z) simultaneously, in order to diversify the risk of the discovery of his actual location. Some kind of research [10, 11] focus on the user movement simulation, which create a more meaningful dummy, in order to increase the safety of protecting user location. Some of them [16,17] use *cryptographic* protocol-based approaches, which is based on a specific transformation fully known only to the clients, the server processes user queries without the ability to decipher exact user locations.

For the encryption side, its concept is to transmit the data from one party to another party, while both parties do not know what is another party sending. Secure multiparty computation (SMC) is one of the major topics in this aspect. The researches have started, since the Yao's Millionaire Problem [18] is published. The Yao's Millionaire is able to compare two private numbers, while the participating entities (including client,server,non-colluding third party) unable to know what the exact information transmit by the other parties. Since then, many research efforts have been made to develop more efficient SMC protocols for specialized functions and tasks, including secure sum, scalar product, add vector and set operation.

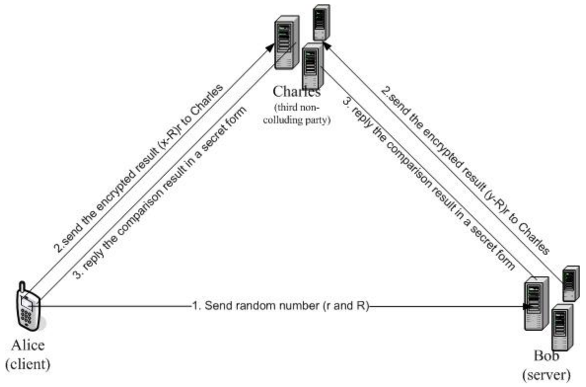

Charles
(third non-colluding party)
2.send the encrypted result (x-R)r to Charles
3. reply the comparison result in a secret form
2.send the encrypted result (y-R)r to Charles
3. reply the comparison result in a secret form
1. Send random number (r and R)
Alice
(client)
Bob
(server)

Fig. 3 Solution of Yao's comparison [3,4]

Fig. 3 is a graphical version of Yao's comparison solution. Alice (client) issues two random numbers to Bob (server), then Alice and Bob will both send their encrypted answer to Charles (third non-colluding party). Those paper [3,4] implement the solution of Yao's comparison, which can solve the problem in an efficient time and employ their own techniques to find the k-nearest neighbor(kNN) of the users.

Previously, there is most of the location privacy research focus on finding (kNN), the research in proximity detection start much later than previous one. The paper in this aspect [26, 27, 28, 29, 30] contribute different ideas in this area. For example [26] suggest VICINITYLOCATOR which do not require peer and peer communication with an adjustable region for proximity detection. However, that suggestion is still required to expose the approximate user location to the third party.

In this paper, we are going to suggest a solution, which can achieve the goal of proximity detection, while not expose the user's location in any manner.

2. RELATED WORK

Firstly, we review some previous work in the location privacy area and show the development idea of the solution of this paper.

Most existing solutions adopt one of the three technique (including cloaking, dummies and encryption) when handling the user location privacy problem.

The earliest proposal for location privacy protection is spatial cloaking [9]. Fig 4 shows us a brief picture of spatial cloaking by using a trusted anonymity server. Instead of sending a single user's exact location to the server, spatial cloaking techniques collect *k* user locations and send a corresponding (minimum) bounding region to the server as the query parameter.

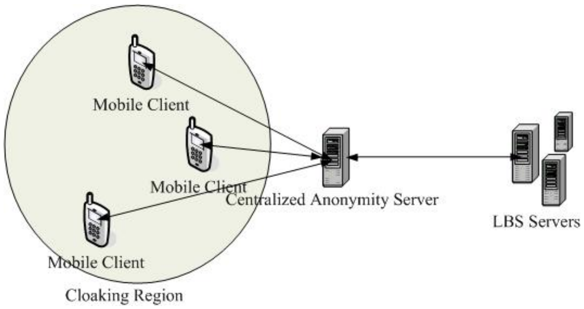


Fig. 4 Cloaking with trusted anonymity server

However, in the previous model, *k* is same for all users in the system. User cannot adjust the privacy level by themselves. It will seriously affect the quality of service when the user goes to an area which has low density of user. Because of the time consuming job on searching nearby users to form a cloaking region. After that, [5] extended this to a personalized k-anonymity model, that mean the user is enable to decide the balance between privacy and efficiency.

More recently, location cloaking algorithms advanced from cloaking of snapshot locations to continuous location updates [1, 6]. The cloaking of snapshot locations is not secure enough to prevent the leakage of location privacy, if an attacker (e.g., the service provider) can collect the user's historical cloaked regions as well as the user's mobility pattern (e.g., users' speed).

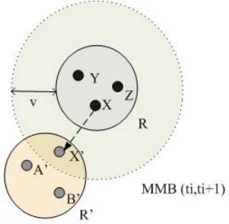


Fig. 5 Location Dependent Attacks

[21] shows that a circular cloaking region generally lead to a small result superset, so that less computational power and time is required for cloaking process. Fig 5. use circular cloaking region to show us the threat of snapshot locations cloaking. If we know that the maximum speed of client X, then we can calculate the maximum movement boundary (MMB) by creating a bigger circle which have the same center of the original cloaking region R. Then when client X emits his request for new cloaking region R'. We can find client X much more easily because client X must in the overlapping area of MMB and R'. Therefore we can see the importance of prevent the user location discovered by the continuous location tracking.

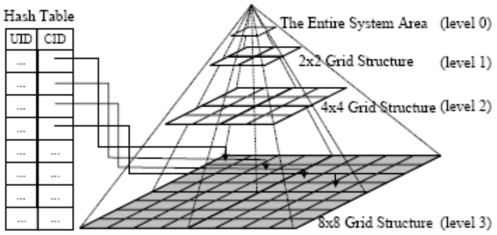

Hash Table
UID
CID
...
The Entire System Area (level 0)
2x2 Grid Structure (level 1)
4x4 Grid Structure (level 2)
8x8 Grid Structure (level 3)

Fig. 6 Location Dependent Attacks

The Casper [13] consists of two main components, the anonymity server and the privacy-aware query processor. The anonymity server blurs a user's exact location information into a cloaking spatial region based on user specified privacy requirements. The privacy-aware query processor is embedded inside the AS to deal with the cloaking spatial areas rather than exact location information.

This framework uses a quad-tree data structure that maps the location information into grids with different levels and resolutions. The cloaking algorithm goes through the quad-tree in a bottom-up fashion to find a spatial region that meets the privacy requirements.

This approach also requires the anonymity server to dynamically keep track of the locations of mobile devices in a fine spatial resolution. This can easily lead the anonymity server become a performance bottleneck when there are large number of client updates and requests. Due to the limitations of the quad-tree structure, the calculated cloaking region is often larger than required, which may cause lower service quality.

While the above cloaking algorithms need a centralized trusted third party to perform location cloaking.
Chow *et a.* [20] proposed a peer-to-peer cloaking algorithm based on information exchanges among mobile clients. The paper point out two shortcomings of the centralized privacy-preserving framework.
When the entire location privacy queries processed by a single anonymity server, that server will become the system bottleneck. As cloaking function is quite time-consuming solution, therefore if there are many clients using the service, the response time for the server will be very slow. Also, if it is compromised by an attacker, or forced to cooperate with a government agency, all the users' movement records. All the protected information will be totally revealed.

Therefore some papers present some kinds of decentralized approach which can achieve anonymity without an anonymity server, so as to solve the high communication overload problem while using the cloaking method.
Peer-to-peer cloaking [20, 22] is one of the solution, which utilizes peer-to-peer communications Each moving object probes its neighborhood to look for other moving objects and anonymizes its own location using the information collected.

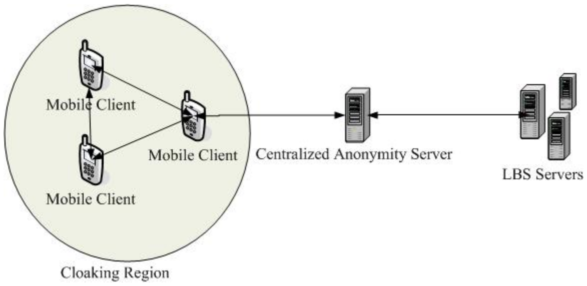


Fig. 7 Cloaking without trusted anonymity server

For instance the case of Fig 7 shows that a group of clients form a cloaking region by themselves. So that it can reduce the large amount of workload on the centralized server because the most time-consuming part (cloaking) has been done by the users.

However, the drawbacks of this approach are, for example it does not guarantee that the service requests can be fulfilled because there may not be enough peers nearby. As a result, the service availability is not consistent.

Moreover, privacy leakage may happen because the requester tends to be in the center of the cloaking region.

Furthermore, as the user's location always changes when they are using the location-base system. Therefore we are facing the dilemma of changing the cloaking peers or enlarging the cloaking region. If we employ the former case, we need to update the cloaking set frequently. Because of the limited capabilities of current mobile, a peer-to-peer system may also pose too much computational and communicational overhead. If we employ the latter case, the cloaking region will become larger and larger. As a result, the quality of service will degrade rapidly.

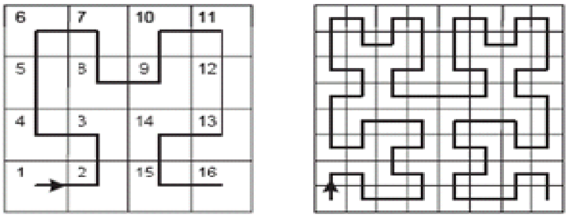


Fig. 8 Hilbert curve (Left (4x4) Right (8x8)

They also applied this algorithm to a distributed environment based on an annotated B+-tree index.
In [23], a decentralized approach based on Hilbert Curve is proposed to meet the reciprocity property requirements of the location cloaking algorithm. This approach guarantees the query anonymity even location information is disclosed to the adversary. However, each client needs to maintain relatively complex data structure and communication protocol as well as long range communication among peers. Therefore additional computation and communication cost may be posed to clients with limited capabilities.

Generating faked dummies is another kind of decentralized approach which does not involve any AS. For example, the paper [10, 11, 16, 17] lets the mobile clients generate false locations and send them along with the real locations to LBS. For every location update, a user would send n different locations to the server. Only one of them is true. The rest are dummies. Thus, the server cannot know which location is the actual one. However, it is still possible to detect false dummies through data mining techniques if the algorithm used for dummy generation is not selected appropriately.

Another way to hide the client location is using cryptography solution. Yao's [18] present how to exchange secret by using some comparison method. After that paper [3] propose a kNN protocol for horizontal partitioned data and provide a privacy-preserving algorithms which can handle the large dimensionality and diversity of attributes common in vertically partitioned data.

[2] use Private Information Retrieval (PIR) implementation to build up a location privacy protection framework which does not require an anonymizer or collaborating trustworthy users. However, the limitation of this implementation is the cell contents have to match the query result that may cause a high storage overhead because the server is required to store different type of content. Also, it is not easy to find the optimal size of grid partition in order to minimize the computation and communication time.

Except the research of finding kNN, proximity detection is also a valuable topic for us to discuss. Sometimes we may want to find friend or place where is within in a certain distance. kNN may give us too much information when most of the point of interest (POI) are near us or give us too limit information when those POI are too concentrate together. As a result, we may not get what we want because of the uneven distribution of POI. Location cloaking may give us the solution, but we can until know the approximate location. In order to have a better protection for the users, we should create a better framework that none of us (including central server) know the other users' location except we are within a certain distance.

3. SECURE PROXIMITY DETECTION

Under an agreement, users in a social group may allow users in the same group to know where we are, when we are nearby. Nevertheless, tradition proximity detection is still requiring us to expose our approximate location, in order to maintain the operation. Our solution is able to solve such problem. The workflow of the system is shown in the Fig 9.

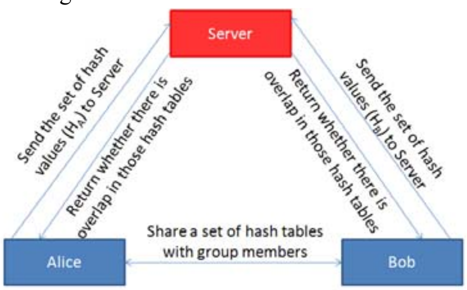

Server
Send the set of hash values (H_A) to Server
Return whether there is overlap in those hash tables
Return whether there is overlap in those hash tables
Send the set of hash values (H_B) to Server
Share a set of hash tables with group members
Alice
Bob

Fig. 9 System workflow

It is a solution which does not need to send the approximate location to any third party. Also, users in the same group can know you are nearby, if and only if you are within a certain distance. Therefore it can complete protect the user's location privacy, while providing the LBS.

First, we introduce the structure of our system. In order to simply the demonstration, we will assume in this system there are only 2 users, Alice and Bob. At the beginning Alice and Bob will communicate and create an agreement which contain an acceptable distance D. Then one of them will generate a set of hash tables which cell size is (D*D) and distribute those generated table to all other group members.

As 2 users are fallen in the same hashed cell, we can sure that they are within the distance D. However, there is still 75% of missing cases that we have not handled. Such as the example shown in the Fig. 10a. This problem can be solved by generating more same size hash tables that is randomly shifted to any direction Fig 10b.

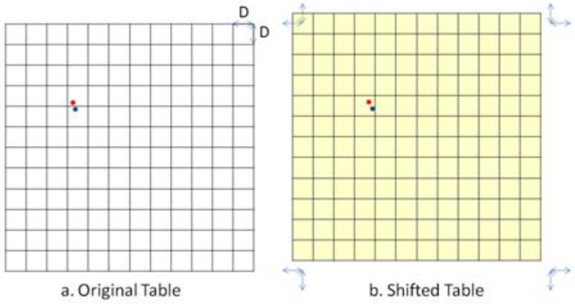


Fig. 10 Grid-based hash table

The original missing rate of a single hash table is 75%. However after 20 more randomly shifted table is added the missing rate will drop sharply to $(0.75)^{20}$ = 0.3%. After adding more tables, even there is only one overlap hash table we can still ensure that they are within a distance D.

3.1 Update handling

This solution is able to handle the continuous location update of the users. In the following picture Fig. 11, it shows the location of the user is at the center. Different color square represent different hash table. The user is required to update their location information to the server, whenever they enter or quit any grid cells in the hash tables.

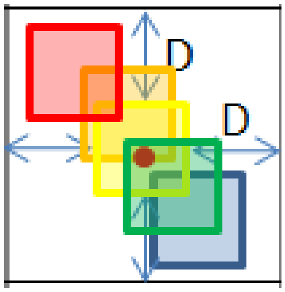


Fig. 11 Grid-based hash table

Based on the update rule mentioned previously, if the D is very small, the communication cost will become very high. Because of the frequent updates are required, when we always enter or quit the cell. Therefore we should have further improvement in the solution, so as to reduce the update cost.

3.2 Way to minimize the communication cost

We can adopt different sizes of hash table sets to help as to reduce the update cost. It is workable especially when 2 users are far away from each other, the distance between them longer, less update is required.

For instance we have already try 20 randomly shifted same size (size = D^2) hashed table, none of them are overlapped. Then we can conclude that they are very likely at least distance D apart from each other.

In the process of hash cell (size = D^2) randomly shift to different direction, it has also approximately cover the larger area (size = $(3D)^2$).

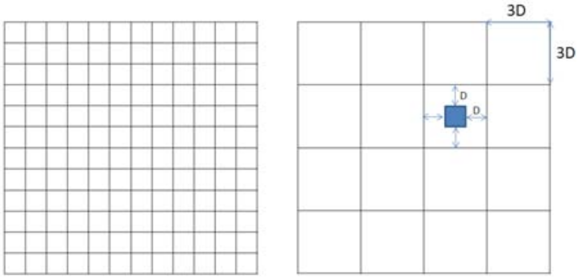


Fig. 12 Extension of from the base layer

In the example shown in Fig. 13, the distance between user A and B is 1.1 D, no matter how we shift the size D^2 hashed table. They will never fall in the same hashed cell. Therefore, we can at least ensure they are at least apart from each other more than distance D.

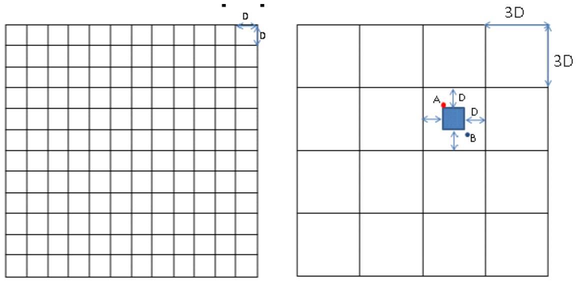


Fig. 13 Example in D^2 layer

Another example is the extended vision of previous example is shown in Fig. 14, the distance between user A and B is 1.1 D, no matter how we shift the size D^2 hashed table. They will never fall in the same hashed cell. Therefore, we can at least ensure they are at least apart from each other more than distance D.

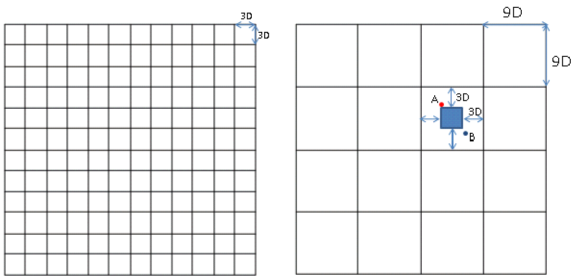


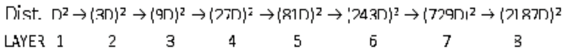
Fig. 14 Example in $(3D)^2$ layer

Finally, this concept can be further extend to $D^2 \rightarrow (3D)^2 \rightarrow (9D)^2 \rightarrow (27D)^2$ …etc. layers. Therefore less update is required if we know the larger distance we are apart from each other.

3.3 Structure of hash tables layers

For a certain group, they will exchange n sets of hashed tables (layer 1 – n) between each other.

If n = 8, then group member will need to share 160 hashed table when each has 20 randomly shifted table.



Dist.	$D^2 \rightarrow$	$(3D)^2 \rightarrow$	$(9D)^2 \rightarrow$	$(27D)^2 \rightarrow$	$(81D)^2 \rightarrow$	$(243D)^2 \rightarrow$	$(729D)^2 \rightarrow$	$(2187D)^2$
LAYER	1	2	3	4	5	6	7	8

We start the checking in a bottom-up manner and stop until the first overlap layer is found. Then the system will work as follow:

Let x is the current layer

1. no overlap in layer x ∩ no overlap in layer x+1
   increase the layer level (bigger cell)
2. no overlap in layer x ∩ no overlap in layer x+1
   remain in the same layer
3. overlap in layer x
   decrease the layer level (smaller cell)

4. CONCLUSION

This solution provides a safe and flexible ways in proximity detection. Also, the efficient layer structure reduce the communication cost to a reasonable level, which can be used in many practical situation.

REFERENCE

[1] T. Xu and Y. Cai. Location Anonymity in Continuous Location-based Services. In ACM GIS'07, pages 300--307, November 2007.

[2] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, “Private queries in location based services: Anonymizers are not necessary,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, Vancouver, Canada, Jun. 2008, pp. 121–132.

[3] Artak Amirbekyan and Vladimir Estivill-Castro. Privacy-preserving k-nn for small and large data sets. In Proceedings of the ICDM Workshops, 2007.

[4] Processing Private Queries over Private and Indexed Data

[5] Buğra Gedik , Ling Liu, Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms, IEEE Transactions on Mobile Computing, v.7 n.1, p.1-18, January 2008

[6] Xian Pan, Jianliang Xu, Xiaofeng Meng**,** Protecting location privacy against location-dependent attack in mobile services, Proceeding of the 17th ACM conference on Information and knowledge management, 2007

[7] Haibo Hu, Jianliang Xu, Non-Exposure Location Anonymity, Proceedings of the 2009 IEEE International Conference on Data Engineering

[8] Chengyang Zhang, Yan Huang, Cloaking locations for anonymous location based services: a hybrid approach, Volume 13, Issue 2 (June 2009) table of contents, Pages: 159 - 182

[9] M.Gruteser and D. Grunwald, "Anonymous usage of location-based service through spatial and temporal cloaking," Proc. Of the International Conference on Mobile Systems, Applications, and Services (MobiSys' 03), pp163-168, Scan Francisco, USA, 2003

[10] Hidetoshi Kido, Yutaka Yanagisawa, Tetsuji Satoh, An Anonymous Communication Technique using Dummies for Location-based Services, Pervasive Services, 2005. ICPS '05. Proceedings. International Conference

[11] Hidetoshi Kido, Yutaka Yanagisawa, Tetsuji Satoh, Protection of Location Privacy using Dummies Location-based Services, Data Engineering Workshops, 2005. 21st International Conference

[12] O. Goldreich. The Foundations of Cryptography, volume 2, chapter General Cryptographic Protocols. Cambridge University Press, 2004.

[13] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proc. of VLDB*, 2006.

[14] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing Location-Based Identity Inference in Anonymous Spatial Queries. *IEEE TKDE*, 19(12):1719–1733, 2007.

[15] Hua Lu, Christian S. Jensen, Man Lung Yiu, PAD: Privacy-Area Aware, Dummy-Based Location Privacy in Mobile Services, 2008.

[16] P. Indyk and D. Woodruff. Polylogarithmic Private Approximations and Efficient Matching. In *Proc. TCC*, 2006.

[17] A. Khoshgozaran and C. Shahabi. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In *Proc. SSTD*, 2007.

[18] A.C. Yao. Protocols for secure computations. In Proceedings of the 23nd Annual IEEE Symposium on Foundations of Computer Science, 1982.

[19] A.C. Yao How to generate and exchange secrets. In Proceedings 27th IEEE Symposium on Foundations of Computer Science.

[20] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based services. *ACM GIS*, Arlington, VA, 2006.

[21] J. Xu, X. Tang, H. Hu, and J. Du. "Privacy-Conscious Location-Based Queries in Mobile Environments." *IEEE Transactions on Parallel and Distributed Systems (TPDS)*,

[22] H. Hu and J. Xu. "Non-Exposure Location Anonymity." Proc. IEEE 25th International Conference on Data Engineering *(ICDE '09)*, Shanghai, China, March 2009

[23] Ghinita, G., ,Kalnis, P., Skiadopoulos, S.:Mobihide: A mobile peer-to-peer system for anonymous location-based queries. In: SSTD '07: 10th International Symposium on Advances in Spatial and Temporal Databases, Boston, MA, USA, Springer (2007) 221–238

[24] Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, Washington, DC, USA, IEEE Computer Society (2005) 620–629

[25] Xiaokui Xiao, Yufei Tao, Dynamic Anonymization: Accurate Statistical Analysis with Privacy Preservation. Proceedings of the 2008 ACM SIGMOD international conference on Management of data

[26] Laurynas Siksnys, Jeppe Thomsen, Simonas Saltenis, Man Lung Yiu, Private and Flexible Proximity Detection In Mobile Social Networks, 11th International Conference on Mobile Data Management (MDM 2010)

[27] K. Liu, C. Giannella, and H. Kargupta, "An Attacker's View of Distance Preserving Maps for Privacy Preserving Data Mining," in PKDD, 2006, pp. 297–308.

[28] S. Mascetti, C. Bettini, and D. Freni, "Longitude: Centralized privacy-preserving computation of users' proximity." in Secure Data Management, 2009, pp. 142–157

[29] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, "Privacy-aware proximity based services," in MDM, 2009, pp. 31–40.

[30] P. Ruppel, G. Treu, A. Küpper, and C. Linnhoff-Popien, "Anonymous User Tracking for Location-Based Community Services," in LoCA, 2006, pp. 116–133.

Survey of Content-base Music Information Retrieval

DENG Jie

Abstract

As the digital music becomes more and more huge, there is a wide range of music services, especially in music information retrieval (MIR). With the rapid progress of the information science and technology, the content-based music analysis and processing become hot area. MIR seeks to utilize the theory of music to better search database of music contents through musical expressed queries, that is to say, searching music by music. This area research has significant commercial and research promise.

This paper mainly gives a comprehensive review of the latest year's research on MIR. In addition, some key techniques and approaches to MIR will also be introduced, particularly focusing on content-based MIR. Main issues of content-based MIR will be summarized. Moreover, the paper introduces some basic music terms and concepts, and music processing. At the same time, some high level abstraction such as semantic description has broadened the future of MIR.

***Key words**: music information retrieval, content-based*

1. Introduction

With the fast development of the Internet and digital devices, there are enormous amount of digital music available for people accessing and downloading via PC and mobile device. People have great music information needs, for example seeking music songs accurately. Moreover, music information retrieval has a wide rang of potential applications. As the emergence of social network, music reviews, ratings, and some similar or personalized recommendations are becoming more important [1]. Therefore, they all bring great challenge to music information retrieval. Currently, music information retrieval is becoming a hot research area for solving the users' particular interest and needs. As the music is an art form whose elements are pitch, rhythm, dynamics, timbre and texture [2], it's very difficult to describe them. In addition, music is relevant to human auditory perception, which expresses emotion and mood. It is also very difficult to measure them. Therefore, at present, most of the music search engines use some metadata (the name of songs, artist and album, etc.) to classify and retrieval music, which is low-level search conditions, for example, Google and Yahoo all use text-base metadata to search music. Content-based music information retrieval search music by musical concepts such as melody or harmony to describe the content of music, which is a high-level search using real musical features. Because there have less efficient and effectiveness approaches to content-based MIR, less search engines use this retrieval type, such as Midomi.com, and ThemeFinder. However, the improvement of the retrieval approaches are still need made progress.

Currently, most of the researches on music information retrieval focus on the content based, the main idea on this approach is describing a set of music features computing directly from their contents [3]. Although there are some problems, researchers have already developed some systems to allow users retrieval music based on their content descriptors. MIR has many problems of traditional information retrieval. In addition, there are mainly three problems in MIR: music feature representation, music queries and music feature matching. Most of current content-based MIR mainly uses melody information, which consists of a combination of music notes and pitch. Music queries have vivid ways, for example users can query by humming or query by using keyboard imputing key words or some professional music notes. Matching techniques are also key retrieve step, which adopts different similarity measures to quantify the similar music works. There have already much content-base MIR approaches on melody feature matching.

This paper mainly focuses on the review of techniques and approaches for content-based music information retrieval. The paper is organized as follows. In section 2, a brief review of music theory will be given, which consists of basic music concepts and characteristics. Section 3 gives a detailed description on music information retrieval in real world. It introduces the user intent, music data and format, music query mode and processing, and some existing music search systems. Current music information retrieval key techniques will be explained in the section 4, followed by evaluation strategies of the MIR, which are discussed in Section 5. Finally, some conclusions are drawn in Section 6.

2. A Brief Review of Music

Music information retrieval is based on three filed subject: traditional information retrieval, musicology, and audio. The following figure 1 shows the content of computer music. This section will briefly introduce some music theory which is related to MIR. Nicola Orio [4] has given a review of music concepts and characteristics in "Music Retrieval: A Tutorial and Review".

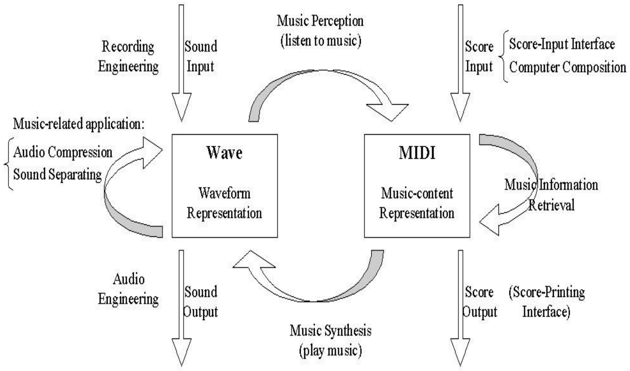


Figure 1, Computer Music

The following will give a brief introduction to music theory based on Nicola's paper.

2.1. Music Concepts

Music is an art form, which produce sounds and silences in time. Based on the music theory [5], there are three basic musical sound features: pitch, dynamics, and timbre. Pitch is the range from low to high, which dependent on frequency of sound. Dynamics is the intensity of sound, which is related to the sound amplitude. Apart from these basic music elements, there are also many common terms used to describe music [4], for example tempo, tonality, time signature(how many beats per measure) and key signature(symbols # and b). The staff is the basis of written music. Music note is represented symbols to express the length of the pitch. The following figure 2 shows polyphonic musical score measures.

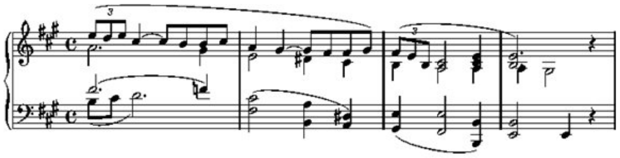


Figure 2, Example of a musical score (excerpt from *Premi`ere Arabesque* by Claude Debussy).

2.2. Music Characteristics

Commonly, music has two dimensions [4]: horizontal and vertical. In vertical dimension, for polyphonic music, the two or more sounds are playing at the same time, which result in a relevant perception and simultaneously active and aligned in the musical score. In horizontal dimension, it associates time to the horizontal axis. In addition, there are also other main dimensions of music which could be effectively used for music search, which consist of timbre, orchestration, acoustics, rhythm, melody, harmony and structure. Thus in theory, any of the above dimension or these combinations is able to use for a relevant descriptor of a music work.

However, in practice, the most used dimensions are melody, rhythm, and timbre. As the melody is soul of music, which is an organized succession of pitches, it is the foundation of music. In addition, melody can be represented by the main theme of music works. Thus, people always can easy memory music through melody. Rhythm is another key characteristic and will give us intuitive feeling, which is the movement of the music in time. Timbre is the quality of musical note which is used to distinguish the different music instruments. For the content of music not just the composer and his or her audience, but the occasion for which it was written and the time and place in which it was performed, it is not easy to express like pictures, because it is just a sense for human beings. Thus for the effective and efficient music retrieval, the above terms and characteristics in musicology are very useful.

3. MIR in the Real World

As adoption of digital audio, people are able to enjoy and spread music conveniently. One can generate amount of digital music with different styles and formats. Music information retrieval today exists with extremely diverse characteristics and content and different users. Thus, all these factors have inspired many companies such Google, Yahoo attempting to provide convenient and effective and efficient music search engines. Therefore, music information retrieval is now a popular research area and has a wide range of applications for both consumers and musicians.

As far as technological advances are improved, content-base music information retrieval has grown rapidly [15]. However, real world application of the music retrieval technology is currently limited [10].

3.1. Users and Search Mode

In Internet, there are a number of uses interact with music search engine to retrieve what they need. Commonly, there are three categories: casual user, professional users and music theorists or musicologists. Based on the different music levels, there are different search modes for different level users.

The most music search systems (e.g., iLike, Lala, Pandora.com) are based on the texts or tags, which are rich and expressive to describe the music. For example, these texts may contain artist, albums, tracks, lyrics, genre, and comments, etc [22]. Thus, the users search music only by inputting some of these keywords. In addition, some commercial music website may use these texts or tags to mine and estimate song similarity and artist similarity. Then it retrieves songs that the users are favorite and satisfied.

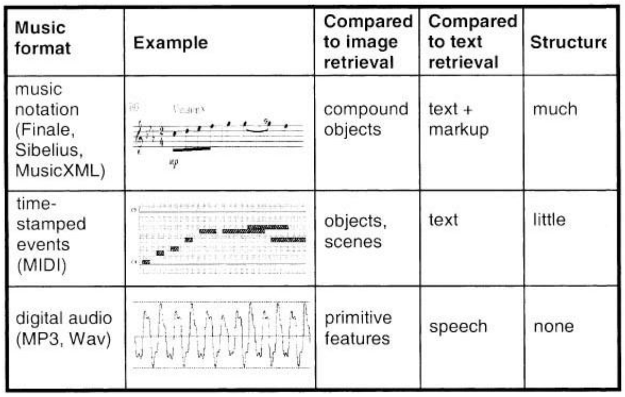
A few music search systems (e.g., midomi, tunespotting) are based on the sound or voice, that is to say they search music by singing or humming or playing part of a song to the computer or digital devices. The longer you sing or hum or play the more accurate the results will be retrieved for you.

Few music search systems are suitable for musical professionals. Some search engines (e.g. Themefinder) identify common themes in Western classical music, folksongs and Renaissance, which require you to have some musical knowledge. When searching music, you need to input some related music parameters, for example, pitch, interval, scale degree, gross contour and refined contour. The others (e.g., tunespotting) are based on the music notation, for example stave. The search engines use the Parsons Code, a rough description of the melodic contour to search music.

3.2. Music Data and Format

Understanding the music data and scope and format plays an important key in the complexity of music retrieval system design. The music data in this paper mainly refers to digital music audio. Music is usually divided into three categories based on the amount of concurrency: Monophonic, Homophonic, and Polyphonic [8]. Usually, audio recordings may be labeled with metadata. Metadata can be used to name, describe, catalog and indicate ownership or copyright for a digital audio file, as well as allow user characterizations of the audio content (ratings, tags, and other auxiliary metadata). When audio formats moved from analog to digital, it maybe embeds these metadata with the digital content itself.

There are mainly three musical formats: Symbolic format, Audio format and MIDI [9]. However, in computer, all the music is digital information, which has two expressions: based-on musical instrument digital interface and waveform data, for example mp3, wav. The most uncompressed audio file format is based on pulse code modulation, which samples analog signal, then quantize, and encoding to gain digital music files. MIDI is an industry standard which enables electronic musical instrument to communicate with computers. Figure 3 shows a comparison of current common music formats.



Music format	Example	Compared to image retrieval	Compared to text retrieval	Structure
music notation (Finale, Sibelius, MusicXML)		compound objects	text + markup	much
time-stamped events (MIDI)		objects, scenes	text	little
digital audio (MP3, Wav)		primitive features	speech	none

Figure 3, Common Music Formats

3.2. Music Processing

The first step of music information retrieval should be able to automatically process the musical file to extract useful features or content descriptors. Different musical file format should adopt different methods. Early research [17] mainly focuses on symbolic music, for example SMF and MIDI, due to it can easily extract melody features and mp3 and wma audio formats are not popular at that time. Currently, most of research on music feature extraction focus on the digital audio form. As the most important music feature is melody, extraction and segmentation of the melody are the key steps for music retrieval based on the melodic information.

In symbolic form, it's very easy to extract melody in monophonic score, while it's more complex to extract melody in polyphonic score, but the most common using statistical approaches. As the melody can be view as a sequence of hundreds of symbols in the symbolic form, thus shot sequence of the melody are more efficient for melody search. N-grams are often used to segmentation. In addition a priori music knowledge will also be used to segment melody. In audio form, the extraction and segmentation is more complex. The pitch and length of music note is the major elements of melody. Thus, the important process in audio music processing is extracting the pitch and determines length. The following sections will give a detail description to these features extraction and process.

3.4. Overview of Some Existing Music Search Systems

There are some common functions in some existing music search engines [6].

- Search by music related metadata: (artists, albums, tracks, music reviews, new release, etc.) Yahoo! Music and Allmusic are the examples of this type.
- Search by music lyrics: Lyrics.com and SongLyrics.com
- Music Media Management and Track Identification: Identify metadata for music tracks, for example Gracenote and MusicIP.
- Recommend similar music: by mining some music feature elements (melody, rhythm, tone color, etc) to recommend user some similar music.
- Recommend personalized music: by mining some users' information to recommend them some their favorite music.

In addition, Musipedia [31] can search music by whistling a theme, playing it on a virtual piano keyboard, tapping the rhythm on the computer keyboard, or entering the Parsons code. It also offers three ways of searching: Based on the melodic contour, based on pitches and onset times, and based on the rhythm. In high level, some system may cross media retrieve supporting natural language queries like mood, which contains semantic information [11].

4. MIR Key Techniques

As the past ten years research on music information retrieve, there have developed some key techniques on music representation, music feature extraction, segmentation, clustering and matching. Tseng [12] has given a content-based model for music collections. The following part will focus on the content-base music information retrieve key processes. Figure 4 shows the framework of the content-base MIR. Figure 5 shows the global MIR map. It contains three hierarchies: music layer, stored data layer, and information layer.

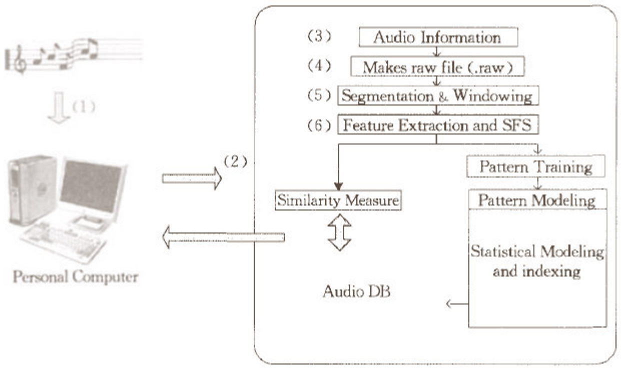

(1)
Personal Computer
(2)
(3) Audio Information
(4) Makes raw file (.raw)
(5) Segmentation & Windowing
(6) Feature Extraction and SFS
Pattern Training
Similarity Measure
Pattern Modeling
Statistical Modeling and indexing
Audio DB

Figure 4, Framework of Content-based MIR

4.1. Basic Approaches to MIR Analysis

In digital audio [7], the recordings are sound waveform, which expresses the frequency changes. In all the sound, the most important element is pitch. A sound consists of a fundamental frequency and a set of harmonic frequency. Thus a sound can be expressed using the following formulation.

$$S (f, t) =\Sigma amp (k, t) \sin (kft + phase (k))$$

n stands for number of frequency, amp (k, t) stands for the amplitude of frequency k in time t. In the representation of music, every music note stands for a special frequency. When analyzing the mp3 audio format, using the relationship of fundamental frequency and harmonic frequency is necessary.

When extracting audio feature, we should first window audio file, which divides it to small segment, every segment is called a frame, then use Fourier transform or fast Fourier transform, and finally compute the audio feature valued. There are two categories features: perceptual feature (pitch, dynamic, rhythm, etc) and non-perceptual feature (Mel-Frequency Cepstral Coefficients, Average Zero-crossing Rate, and Fundamental Frequency, etc). MFCC is a feature set popular used in speech processing and music modeling.

Audio music segmentation is implemented by detecting the features such as fundamental frequency or energy to segment edge. The edge is the suddenly change point of the audio features. By using automatic audio music segmentation, it's very convenient to segment the accompanying and sing part. There are some melodic segmentation approaches [18]. Fixed-Length Segmentation extract from a melody of subsequences of N music notes. Data-Driven Segmentation can be considered as an extension of N-grams approach. Perception-Based Segmentation will segment melody according to theory on human perception. Mixed Segmentation and Data Fusion will describe the melody with a set of note sequences. Query Segmentation requires will apply the same segmentation to music files and queries.

When combination of different audio music feature has done, it can be used to index audio music files by feature vector. Therefore, different similarity algorithms are able to adopt by computing these feature vectors distance. Then the similar audio music file will be clustered automatically. Therefore, when users query by a audio music sample, the system will automatically extract the feature and then formulate the query vector. Subsequently, the distance between query vector and vector space or previous formulated music cluster vector will be calculated. The threshold will be defined to determine the similarity standard. In addition, the similarity value can be also used as the output result rating.

4.2. Music Representation

From the music theory, music melody and rhythm are most used content-base MIR conditions. In fact, most music feature used to represent music is also melody and rhythm. There are mainly three melody representations: absolute, relative, and melody contour. Absolute method uses characters or numbers to encode absolute pitch and length, while relative method uses music interval as the pitch vector. Melody contour method [5] uses three characters to express the basic contour of the melody. Figure 5 shows the pitch contour descriptor and example.

In this method, S stands for the same with the previous note. U stands for higher than the previous note, while D stands for lower than the previous note. Comparison to the absolute and relative methods, contour can not retrieve the result accurately, but it decreases the search vector space.

$$PC_k(Pitch_{k,i}) = \begin{cases} "U" & if(Pitch_{k,i} > Pitch_{k,i-1}) \\ "S" & if(Pitch_{k,i} = Pitch_{k,i-1}) \\ "D" & if(Pitch_{k,i} < Pitch_{k,i-1}) \end{cases}$$

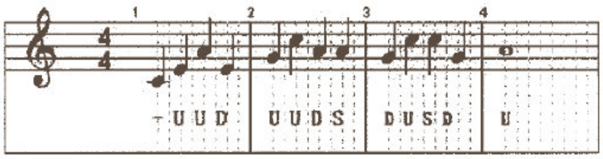


Figure 5, Melody Contour

Rhythm feature representation only considers the rhythm, which omits the melody. If it utilizes character to represent different length music note, maybe a short music score requires much encoding characters, which is not efficient and convenient. Timbre is related to the spectrum of sound, thus Fourier transform is the most frequently used tools in timbre representation and analysis.

4.3. Music Feature Extraction

The majority of music features are melody, timbre, rhythm, thus this part will mainly explain extracting these features [25]. As to melody extraction, pitch is the main element. There are category algorithms: time domain and frequency domain models. Autocorrelation function (ACF), Average magnitude difference function (AMDF), and Simple inverse filter tracking (SIFT) are all common used time domain methods. Harmonic product spectrum and Cepstrum are frequency domain method.

Rhythm is the most useful features for extraction. Most popular music and rock music always use tempo tracking to recognize rhythm. It’s very difficult to describe timbre, but it has a close relationship with performance instruments. Mel-Frequency Cepstral Coefficients (MFCC) has already used as a content descriptor of musical sound. Short-Term Fourier Transform Features (FFT) [21] is a set of features related to timbre textures and not captured using MFCC.

4.4. Music Retrieval

There is mainly two categories content-base music information retrieval: symbolic data and audio data. Researchers have made a lot of research on MIDI, which transmits pitch and intensity of musical notes to play. However, waveform music data has widely used nowadays, thus waveform music retrieval becomes hot. The following part will introduce these two category music information retrieval approaches.

4.4.1. Searching Symbolic Data

Melodic retrieval based on sequence matching [14] is to retrieve music by strings. Strings are able to represent melody, rhythm, and contour and interval sequence and so on. Thus the most of string matching techniques have been applied in using pitch and melody contour. Melody contour has used this approach, which is represented by only three character symbols. Another approach applies pattern matching techniques to query the sequence.

Polyphony music retrieval usually adopts n-grams model to index and match music notes. In this model, every music symbols will be converting to sequence of melody and interval. It highlights musically relevant sequences of music notes, which will undergo a number of different normalization. Patterns were computed using pith or rhythm or their combined information. Alternatively string matching, statistical model based method compares the query probability of attribute to corresponding attribute. Most of the current systems use hidden Markov Mode (HMM) to model a set of themes which are extracted from musical files. In addition, a combined method whose distance function is dynamic time warping (DTW) has been computed.

Sequence matching may be very efficient, but it may require the sequence compared to the entire music file in music collections. In addition, some geometric approaches have been applied to find the best matches with the geometric representation of each music file.

4.4.2. Searching Audio Data

Waveform music file is different from audio symbolic music, which is not able to structure the music directly. The typical waveform music retrieval system contains two parts: building music database and query process [13]. Building music database contains feature extraction and computation for example energy distribution and contour. There is great effort to extract feature of every music file in collections. Currently, music fingerprint is the hot technology to analyze sound wave of music files, which is able to represent a piece of music. Similar or the same music maybe has the same music fingerprint. The process of music fingerprint extraction consists of preprocess, Short-time Fourier Transform. Then it will retain unique identification code, which are also used to recognize music file.

For the high level of search, perception feature based method is proposed to retrieve audio music files. This method extracts the abstract description in audio files, then it will window the audio waveform, and then using Discrete Fourier Transform (DFT) to get parameters and frequency energy, and finally compute the relevant feature vectors value. At last, the similarity will be computed from the query audio to all the music files in database. Lie Lu [23] has proposed a new approach to query by humming. Forevermore, collection-based method has been proposed, which use feature extractor to convert PCM signal to collection form, making processing them like processing music note collections. In addition, Self-organized Map technology is also adopted to retrieve audio music files.

4.4.3. Similarity Measures

When the system retrieves the music file, the search result is also very important. Similarity measures will contribute to the search result. From the above music retrieval key techniques introduced, the most of the music similarity measures are based on audio features, for example MFCC, and computed with statistical approaches, for example the Gaussian Mixture Models (GMM). There are also some popular distance measures used for similarity computation in music retrieval, for example Euclidean, weighted Euclidean, Hausdorff, Mallows, IRM and so on.

As the social network development, some social similarity measures are also becoming important. They are based on the tags, reviews, ratings, etc, which are able to discover the relevant information, while not detecting the audio music files. In addition, the combination of these two ways may have better search results to users.

5. Evaluation of Music Information Retrieval

The evaluation of music information retrieval system is necessary for the system's effectiveness. Symbolic files (MIDI, etc.) and audio files (mp3, wma, etc.) are all should be evaluated. Downie and Nelson [20] have proposed an effectiveness evaluation of n-grams, which is using statistical analysis. There are mainly two popular evaluation measures [19]: precision and recall. Precision refers to the percentage of the retrieved music files which are relevant to the query conditions. Recall refers to the percentage of all the relevant music files in the retrieve music database.

$$\text{Precision} = \frac{\text{Number of retrieved references that are relevant}}{\text{Number of references that are retrieved}}$$

$$\text{Recall} = \frac{\text{Number of retrieved references that are relevant}}{\text{Number of relevant references}}$$

In 2004, Audio Description Contest first attempted build comparative benchmark of MIR algorithms, which has five different tasks: genre classification, artist identification, tempo induction, rhythm classification and melody extraction [27]. It also provides some training and test data. In the following years, it has added other tasks: music similarity and cover song identification.

6. Conclusions and Future Work

This paper gives a comprehensive review of music information retrieval. A brief review of music and some basic music theory are also presented. There are three category users: casual user, professional users and music theorists or musicologists. Based on the different music levels, there are different search modes for different level users. Most music feature used to represent music is also melody, rhythm and timbre. There are mainly three melody representations: absolute, relative, and melody contour, which use U, D and S these three characters. For melody feature extraction, time domain and frequency domain models are applied. ACF and SIFT are separately these two methods. Tempo tracking usually used to recognize and extract rhythm. FFT is a set of features related to timbre textures and not captured using MFCC. Symbolic data (MIDI) search and audio data (MP3, WMA) search are different content-based music information retrieval category. The paper also introduce some evaluations of MIR, Music Information Retrieval Exchange (MIREX) evaluates frameworks and test music collections. For future work, high level semantic music information [28], for example, emotion, mood, and music content abstract will be detailed explore.

References

[1] Jin Ha Lee, J. Stephen Downie. Survey of Music Information Needs, Uses, and Seeking Behaviors: Preliminary Findings.

[2] J. Sephen Downie. Music Information Retrieval.

[3] Michael S. LEW, Nicu Sebe. Content-Based Music Information Retrieval : Current Directions and Feature Challenges.

[4] Nicola Orio. Music Retrieval: A Tutorial and Review.

[5] Roger Kamien. Music: An Appreciation.

[6]Alexandros Nanopoulos. Music search engines: Specifications and challenges.

[7] Jonathan Foote. An overview of Audio Information Retrieval.

[8] Davis Pan. A Tutorial on MPEG/Audio Compression.

[9] Dan Berger. A Music Data Mining and Retrieval Primer.

[10] Donald Byrd, Tim Crawford. Problems of music information retrieval in the real world.

[11] Michael Fingerhut. Music Information Retrieval or how to search for music and do away with incipits.

[12] Aura Lippincott. Issues in Content-based Music Information Retrieval.

[13] Chil-Chin Liu, Po-Jun Tsai. Content-based Retrieval of MP3 Music Objects.

[14] Michael Clausen, Frank Kurth, Roland Engelbrecht. Content-based Retrieval in MIDI and Audio.

[15] Tseng, Y. H. (1999). Content-based retrieval for musiccollections. In Proceedings of ACM SIGIR.

[16] Remco C. Veltkamp, Frans Wiering, Rainer Typke. Content Based Music Retrieval.

[17] Yuen-Hsien Tseng. Content-Based Retrieval for Music Collections.

[18] Giovanna Neve and Nicola Orio. A Comparison of Melodic Segmentation Techniques for Music Information Retrieval.

[19] Jia-Lien Hsu, Arbee L.P. Chen, Hung-Chen Chen and Ning-Han Liu. The Effectiveness Study of Various Music Information Retrieval Approaches.

[20] Downie, S. and Nelson, M. (2000). Evaluation of a simple and effective music information retrieval method.

[21] Musipedia. http://www.musipedia.org/

[22] Wei-Ta Chu, Wen-Huang Cheng, and Ja-Ling Wu. Semantic context Detection Using Audio Event Fusion.

[23] Lie Lu, Hong You, Hong-Jiang Zhang. A New Approach to Query By Humming in Music Retrieval.

[24] Hung-Ming Yu, Wei-Ho Tsai, and Hsin-Min Wang. A Query-by-Singing System for Retrieving Karaoke Music.

[25] Jeremy Pickens. A Survey of Feature Selection Techniques for Music Information Retrieval.

[26] Mark Levy and Mark Sandler. Music Information Retrieval Using Social Tags and Audio.

[27] Hwei-Jen Lin, Hung-Hsuan Wu. Efficient geometric measure of music similarity.

[28] Iman S. H. Suyoto, Alexandra L. Uitdenbogerd, and Falk Scholer. Searching Musical Audio Using Symbolic Queries.

AOC-Based Efficient Waiting Time Management in Hospital

Li Tao

Abstract

Long waiting list or waiting time in public health is an endemic and challenging problem faced by many countries in the world. To manage waiting time is a complex problem because it involves both impersonal factors (e.g., inefficient patient scheduling) and human factors (e.g., dynamically change patient behaviors). Traditional mathematical modeling (e.g., queuing theory) has been used to deal with various scheduling problems (e.g., care units scheduling optimization). However, these approaches, which are often processed in a centralized manner and often regard the health care system/hospitals as deterministic systems (e.g., constant patient arrival rate, stable care service capability), are insufficient to tackle distributed patient scheduling problems involving patient dynamic behaviors in the real world. On the other hand, system dynamics approach is efficient in qualitatively analyzing causing factors of waiting time but meet challenges to address the same problem of scheduling patient flow as mentioned above. In this paper, we mainly focus on a distributed dynamic patient scheduling (DDPS) problem which aims to alleviate the overall waiting time in a hospital. Based on a self-organized computing paradigm called autonomy-oriented computing (AOC), we provide a distributed strategy in which autonomous entities corresponding to organizations are deployed in a temporal constraint network (edge denotes temporal constraint between two nodes), and are capable of scheduling patients to shorten waiting time in total by individual behaviors such as competition and cooperation.

1 Introduction

Long waiting list or waiting time in public health is a notorious problem in most of the countries all over the world [23]. It is reported by [14] that in 1999, there were almost 2 million patients waiting for outpatient services and 1 million patients waiting for inpatient or special care. Averagely, 49% inpatients should wait at least 3 months and 26% waited more than 6 months. Similarly, the waiting time for public health supported patients varies from about 3 to 4 months for patients in Norway [4][14].

The reasons of excessive waiting time can be divided into two levels: (i) macro level: e.g, government policies, resource allocation planning, and (ii) micro level, like limited resources, inefficient doctor-patient interaction, fluctuation of patient number and patient type, central controlled organization management, non-cooperate independent machine-like organizations (departments/units) and etc. For all the reasons above, waiting time problem can not be easily tackled from normal mathematical approaches (e.g, queueing theory)[7][9] and traditional top-down systems approaches (e.g, system dynamics)[21][22].

1.1 Challenges

How organizations managing waiting time more efficiently is a hard task because it involves: (i) impersonal factors like scarce care resources, inefficient resource management and patient scheduling, and (ii) human factors such as redundant doctor-patient interaction, unpredictable patients behaviors as dropping a prearrange treatment, and so on. Figure 1 shows the main direct and indirect causing factors of waiting time and the importance of waiting time management in health care system.

In addition, real waiting time management problem is complex in nature because of:

- **Distributed health care resources**: Health care resources with attributes of quality/quantity/types are distributed geographically which causes various problems, e.g., convergence referral flow, fluctuation of treatment cost and waiting time, distinct treatment plan, and so on.
- **Dynamically changing demands**: Request for health care is dynamically changing because of unpredictable diseases, uncertainty patient decision making, patient status natural transfer (e.g, from sick to healthy or to dead), and etc.
- **Independent organization decision making**: Traditionally, organizations in health care system are regarded as a part of deterministic system. They almost make decisions (e.g., patient scheduling) based on own

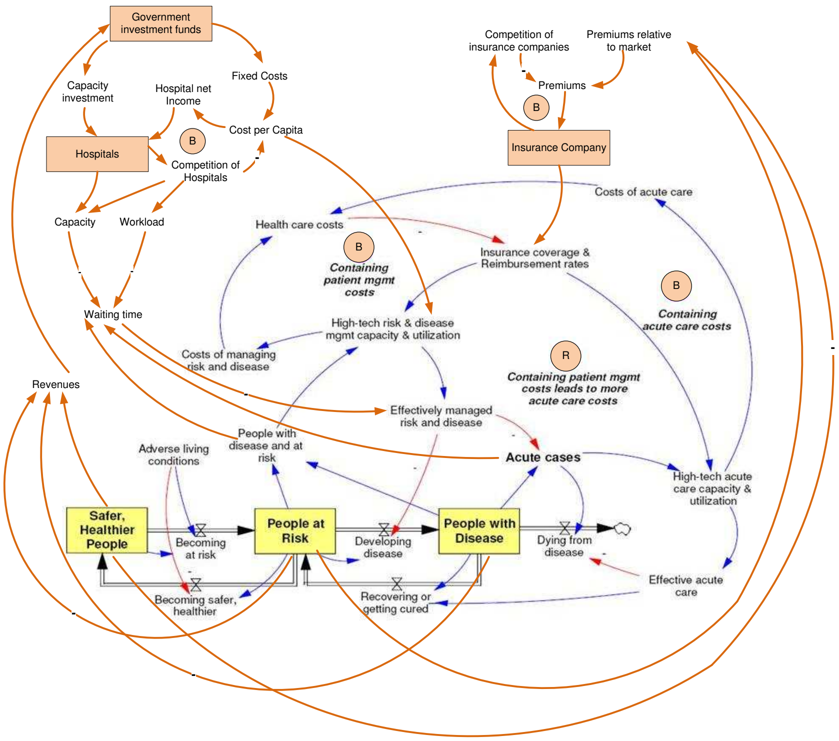

Government investment funds
Competition of insurance companies
Premiums relative to market
Fixed Costs
Capacity investment
Hospital net Income
Premiums
B
Cost per Capita
Hospitals
B
Competition of Hospitals
Insurance Company
Costs of acute care
Capacity
Workload
Health care costs
B
Containing patient mgmt costs
Insurance coverage & Reimbursement rates
B
Containing acute care costs
Waiting time
High-tech risk & disease mgmt capacity & utilization
Costs of managing risk and disease
R
Containing patient mgmt costs leads to more acute care costs
Revenues
Effectively managed risk and disease
Adverse living conditions
People with disease and at risk
Acute cases
High-tech acute care capacity & utilization
Safer, Healthier People
Becoming at risk
People at Risk
Developing disease
People with Disease
Dying from disease
Effective acute care
Becoming safer, healthier
Recovering or getting cured

Figure 1: Causing factor of waiting time from system dynamics approach. This graph shows that hospital capability, workload efficiency and number of patients have direct impact on waiting time, and waiting time directly influences effectively managed risk and disease. (This graph is drawn based on the work of [12][13])

interests (e.g., first in first serve) rather than the global objectives (e.g., minimize total waiting time).

- **Non-linear and dynamic coupling between organizations**: The coupling relationships among organizations can not be easily expressed as linear function because they dynamic change case to case. E.g., there is weak coupling among organizations in routine, while strong coupling in emergency cases.

- **Incomplete non-centralized information**: Information (e.g., about patient treatment plans, patient decisions, organization circulation) is partially available for organizations as well as for patients due to the open, distributed, dynamic and large scale (thousands of patients, number of organizations) nature of health care system.

In the past, several approaches have been used to solve waiting time related problems, ranging from mathematical modeling to complex system theories. Queueing theory [9][7] has been adopted to schedule patients in the context of single organization with static and global patient information. However, these piecemeal works cannot easily extended to solve an open, dynamic and large scale (means not limited in a department or one kind of resource, but at a hospital level with hundreds of departments and resources) scheduling problem. Also, complex systems theories like system dynamics have been used to study waiting time and predict the efficiency of management polices from global level. This top-down approach has limitations to provide a practical solution representing the dynamics and to reveal the basic rules which resulted in such a complex problem.

1.2 Our Consideration

In order to efficient patient scheduling in an open, distributed, dynamic and large scale context, it is necessary for us to understand what are the forces causing waiting time and what are the mechanisms underline such a distributed dynamic patient scheduling (DDPS) problem. In this paper, we study DDPS problem aims to shorten waiting time in the hospital level from a new perspective called Autonomy Oriented Computing (AOC)[15][16][18]. AOC is a new paradigm which model phenomenons or problems from a bottom-up approach. The patterns or solutions for a complex problem will be emerged from a natural like self-organize process. Based on AOC framework, the organizations in hospital can be regarded as intelligent entities. They behave (e.g., compete, cooperate) according to intrinsic behavior rules towards their own goals. Ultimately, global objective (an efficient patient schedule) will be achieved through the process of entities self-organization.

In detail, there are some interesting and important questions we should try to answer:

- What are the organization connection structures at different levels of health care system? Intuitively, we suppose they are network-liked structure. So, what kind of network it belongs to?

- What are the characteristics of this kind of network and how to identify? For example, what are the key nodes (bottlenecks in treatment flow) and key edges? How about the coupling between nodes? These kinds of characteristics may have important physical meaning.

- What are the mechanisms with which organizations can achieve a global target by process of self-organization?

- What patterns or structures can emerge from these adaptation evolution process? E.g., whether the network structure will be evolved from self-organization process?

Specifically, this paper will focus on how to model and design strategies to solve DDPS problem based on AOC framework. The main objective of this paper is not only to provide a distributed solution for dynamic scheduling problem, but to verify that whether a behavior based distributed strategies are more suit to solve DDPS like problems (e.g., supply chain management in e-market, task scheduling in web service, and so on).

The rest of this paper is organized as follows. In section 2, we briefly summarize some typical related work related to efficient waiting time. In section 3 we define and formulate DDPS problem in detail. Section 4 proposes our strategy based on AOC. Lastly, we summarize this paper in section 5.

2 Related work

To better waiting list/time, endeavors include (i) exploring solutions and strategies from theoretical study, and (ii) carrying out practices in real world.

Theoretically, extended from classic job shop scheduling problem (JSSP) [5], traditional mathematical modeling techniques like queueing theory [7] [9]have been commonly adopted to schedule patients. Paper [7] constructs a queueing model to assess the impact of service outages, to approximate patient flow times and to evaluate a number of practical applications. Paper [9] surveys a range of queuing theory results in the areas of waiting time and utilization analysis, system design, and appointment systems. It also considers results for systems at different scales, including individual departments (or units), health care facilities, and regional health care systems.

However, there strategies extended from JSSP are not suit to solve DDPS problem in real world because there are some apparent differences between them.

- JSSP has global information about tasks and executive time. It can provide an optimal solution by central planning without regard to polynomial calculation time. But DDPS faces dynamic changing tasks with local information.

- JSSP is a complicated problem while DDPS is a complex problem which needs the ability of reorganizing or self-organizing in real time.

- JSSP often focuses on a small scale. Here, small scale means how to schedule number of tasks onto a few pipelines which have simple linear connection structure. However, DDPS involves a much bigger and complex scale. The number of organizations (similar to pipeline) is big. And there are network liked temporal interconnected structures among organizations.

- JSSP has a static problem space and is a NP-complete problem. But the problem space of DDPS is dynamic changing and more complex.

DDPS also can be regarded as a kind of distributed Constraint Satisfy Problems. It has temporal constraints among some organizations and maximum treatment time limit. As the similar reasons we mentioned above (JSSP is one kind of CSP problem), we can not use the classic methods (e.g, Backtracking method) as well as multi-agent oriented constraint satisfaction approach proposed by [17] to solve DDPS problem.

Due to the limitation of mathematic modeling strategies to efficient manage waiting time, many other researches resort to complexity science to rethinking of health care managerial and this way is believed to be a promising attempt [8][19][20]. System Dynamics (SD)[6], which deal with internal feedback loops/causing flow (figure 1 is an example) that affect the behavior of an entire system through interrelated components, is commonly used to understand the roles and relationships of different components in health care system, and to analyze the efficiency of health care management strategies and policies. For example, paper [10] predicts policies (subcontracting, a program of extending the working day to the afternoon, and waiting list updating) to entail excessively long waiting list in Spain are useless in the long-term. paper [21] and [22] find out that waiting time cannot be shortened proportionally to substitutability among NHS hospitals. Although this traditional top-down complex system approach offers ways to think/understand health care systems that enable us to have new insights about the nature and functions of it, they cannot provide practical solutions and cannot address some key problems such as DDPS.

Practically, many governments (e.g., England [2], Canada [3]) carry out policies to improve waiting time by strategies of giving political pressure and central direction to organizations which mimicked a command economy, and placing a much heavier emphasis on choice, competition and plurality of provision. However, these efforts are not efficient in practical. The real practice conducted by NHS do not just to cut waiting times but 'to move the NHS away from a culture of waiting to a culture of booking' [11]. According to Civitas statistics (number of patients waiting and length of time spent waiting for an inpatient appointment is shown as an example in Figure 2), although the waiting time has been improved to some extent, there are still a lot of tough works to do. So, DDPS problem has practical significance in real world.

3 DDPS Problem

As mentioned above, the goal of this study is patients scheduling automatically and dynamically arranged by cooperated organizations in a hospital. In this section, we will provide a simple scenario to illustrate this problem, then analyze and formalize it in detail.

3.1 Scenario

In a classic hospital, there are several loosely coupled but relatively independent organizations like reception, diagnosis, electrocardiogram and etc. The capability of each organization is stable. Some organizations have temporal constraint relationship. For example, department for consultation is normally visited after register. But electrocardiogram examination and radiological examination do not have sequence requirement.

The behaviors of patients dynamically change. Patients come into hospital randomly day to day. They may join in (e.g, referral from other hospitals) or quit (e.g, referral to other hospitals, decide not continue to this treatment, die, recovery and etc.) from waiting queue of each organization randomly. A simple example of patient states transition is shown in figure 3.

3.2 DDPS Problem Statement

Given context above, consider a set of m patients and n organizations with stable capability (Υ_i is the predefined capability of organization i per day), the goal of our task is to design an adaptive and distributed strategy to minimize the total waiting time given limited and stable capability of each organization in a hospital.

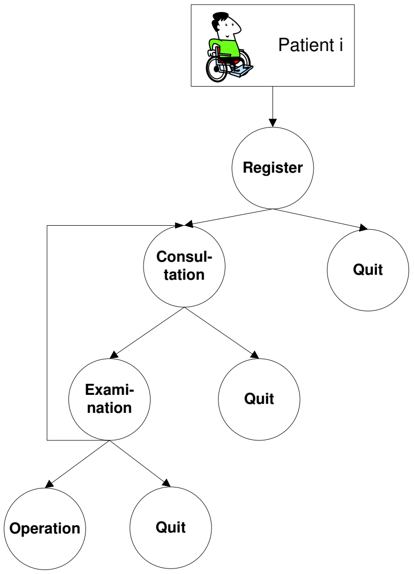


Figure 3: Illustration of patient treatment state transition. In each stage, patient may make a decision about whether come to next stage treatment or quit from current waiting queue. This also demonstrates one kind of patient dynamics.

As mentioned in section 1, we suppose the structure of organizations in hospital is a network (figure 4 is an example). Here, each node means an organization or a department in health care system and directed edge points the temporal order of two organizations. That means, some organizations should be visited in sequence (single directed arrow shown in figure 4) while others need not follow the temporal constraint (double directed arrow shown in figure 4).

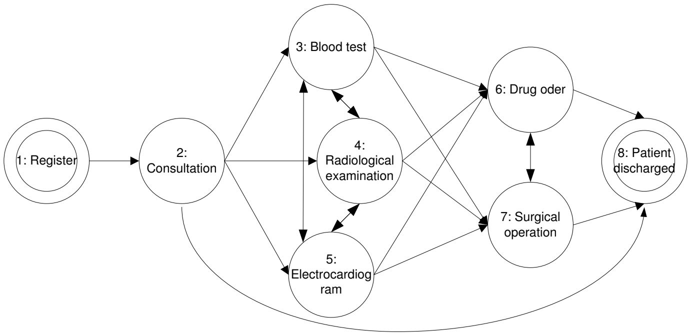


Figure 4: Illustration of organization network of hospital. In this graph, node means department in a hospital. Directed edge shows temporal constraints.

Definition 1: ***Hospital Organization Network***. Graph $G =< V, L >$ is a predefined hospital organization

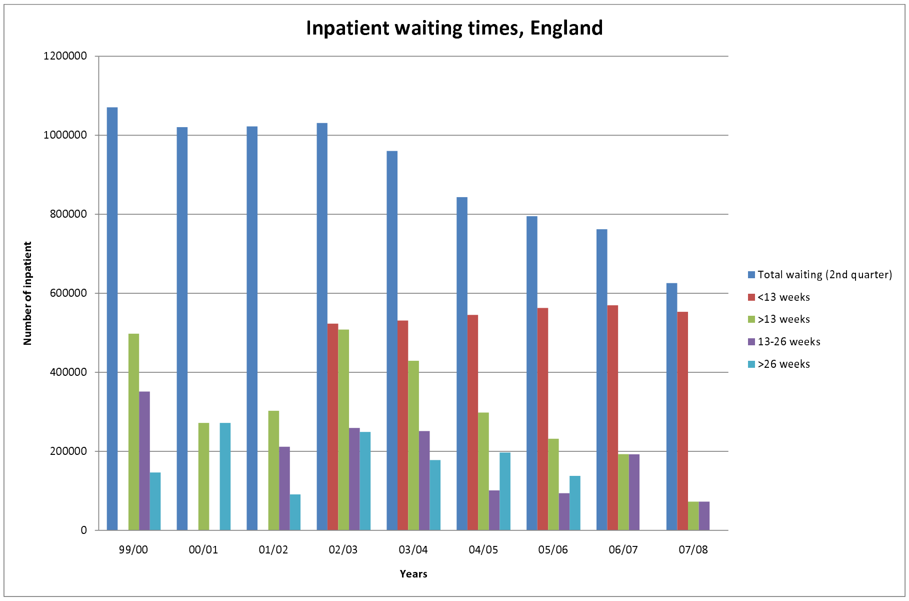


Figure 2: Number of patients waiting and length of time spent waiting for an inpatient appointment, England, 1999/2000-2007/2008 (data is not complete in year 1999/2000-2001/2002). The statistic shows that although the waiting time of inpatient has been controlled less than 26 weeks, the number of waiting inpatient is still large. (Data is adopted from [1][2])

network. $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes, which denotes the set of n organizations, and $L = < v_i, v_j > |1 \leq i, j \leq n, i \neq j$ is the set of directed edges which shows the sequential constraints. For example, the directed link $< v_i, v_j >$ means that the work on v_i should be done earlier than on v_j.

Definition 2: A ***Global Patient Treatment Information*** is a growing matrix $PatientTreat = \{pt_{kj} | 1 \leq k, 1 \leq j \leq n\}$. Where pt_{kj} is a triple which denoted as $< treat_{kj}, in_{kj}, out_{kj} >$. Here, $treat_{kj}$ is the estimated treatment time with organization j for patient k. in_{kj} is the actually treatment start time and out_{kj} is the actually treatment finished time.

Definition 3: ***Patient Personal Waiting Time***. Let $ST_i = in_{i1}$ denotes the start time (time to enter into the first department which normally be the registration department) and $ET_i = out_{in}$ denotes end time (time to leave the last node) of patient i in the hospital. Thus, a waiting time of patient i can be expressed as:

$$W_i = ET_i - ST_i - \sum_{j=1}^{n} treat_{ij}$$

Definition 4: ***Waiting Time Efficiency rate*** can be calculated by two ways. The first way is to see the average waiting time with a fixed patient number m.

$$\bar{WT} = \frac{\sum_{i=1}^{m} W_i}{m}$$

Another way is to see the median waiting time of m patients. This index is denoted as M_{e-wt}. To calculate this index, we first need sort the patient personal waiting time from small to large. Then we can use the equation below:

$$M_{e-wt} = \begin{cases} W_{\frac{m+1}{2}}, & \text{if } m \text{ is odd} \\ W_{\frac{m}{2}} + W_{\frac{m}{2+1}}, & \text{if } n \text{ is even} \end{cases}$$

4 AOC-Based Strategy Formalization

Following AOC framework given in [15][16], each organization is regarded as an autonomous entity which interact with each other as well as the environment and behave according to own behavior rules to reach their local objective. The global objective which is to minimize average waiting time will be achieved in the last.

Definition 5: Each node has an ***Entity*** e. It can be described by a state space $< id, patientOrder, utility, rules >$. Where id is the unique ID of organization where this entity resident. $patientOrder$ is the calling schedule arranged by entity based on $utility$ and $rules$. The $rules$ includes some local behavior like greedy-select, cooperative-select, competitive-select.

Definition 6: The ***Local Environment*** of entity e_i denotes as E_{l_i}. $E_{l\_i} =< neighourID, neighourID.patientOrder >$ In our strategy, an entity can only get the local information which send by parent nodes (which have a directed edge into this entity)and sibling nodes (which have bidirectional edge with this entity).

Definition 7: ***Global Environment*** $E_g = PatientTreat$ provides the prospective treatment time of each patient at all organizations.

An important issue in designing a distributed scheduling strategy is how to enable entities rapidly scheduling patients with direct/indirect interaction. we design three kinds of behaviors for entities to select and arrange their own patient treatment queue.

(1) **Greedy Selection Behavior** is a personal behavior strategy of entity that each entity will select patient with shorter treatment time prior than those with longer treatment time. Of course, the temporal constraints of overall treatment time period of one patient should be considered.

(2) **Cooperative Behavior** will happen among parent nodes and children nodes. If the order of patient i at organization j is later than the order in the schedule of its children $j.children$, then entity j and children entities $j.children$ will cooperate to adjust the patient order. The strategies of how to cooperate will be studied in later work. In this paper, we randomly chose one entity to keep his order. Therefore, other entities will adjust the patient order to keep up the changes and to satisfy the temporal constraints.

(3) **Competitive Behavior** will happen among sibling entities. If the orders of patient i among sibling entities are contradictory, then entities will compete with each other to win the handling priority. The strategies of how to compete will be designed in later work. As well, in this paper, we randomly chose one entity to keep his order. Other entities will adjust the patient order to keep up the changes and to satisfy the temporal constraints.

5 Summary and Future Work

Organizations (departments/units) in hospital can be regarded as living cells or autonomous entities. This paper assumes that a tough and complex problem—-distributed dynamic patient scheduling problem will be solved by local behaviors like cooperation and competition of individual entity. An initial model yields this idea has been proposed from Autonomy Oriented Computing (AOC) approach.

The purpose of our work is not only to provide an efficient distributed and natural like solution to ease waiting time, but also to reveal that shaping the perspective from bottom-up approach (e.g, organization's local behaviors to system's global pattern perspective in health care) is a different but useful tool when solving complex problems. Our work also provide a practical example for complex organization management problems in other areas besides health care in real world.

Future works include (i) to fine-tune our AOC-based model and strategies to better match the situations in real world, (ii) to justify the efficiency and analyze the characteristics of this approach. As well, due to the critical role of organization structure, we will (iii) study some important issues related organization structure in health care system (i.e, organization structure formation and evolution in health care) which have been mentioned in section 1 in the future.

References

[1] Civitas: Institute for the study of civil society. http://www.civitas.org.uk/nhs/nhsperformance.php.

[2] Department of health. http://www.performance.doh.gov.uk/waitingtimes/.

[3] Health canada. www.hc-sc.gc.ca.

[4] *The Reform of Health Care Systems: A Review of Seventeen OECD Countries*. OECD, 1994.

[5] D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3, 1991.

[6] F. E. Cellier. *Continuous*. Springer-Verlag, New York, USA, 1991.

[7] S. Creemers and M. Lambrecht. Healthcare queueing models. Technical report, Katholieke Universiteit Leuven, 2008.

[8] L. Edgren. The meaning of integrated care: A systems approach. *International Journal of Integrated Care*, 8, 2008.

[9] S. Fomundam and J. Herrmann. A survey of queuing theory applications in healthcare. *ISR Technical Report*, 24, 2007.

[10] B. Gonazalez-Busto and R. Garcia. Waiting lists in spanish public hospitals: A system dynamics approach. *System Dynamics Review*, 15(3):201–224, 1999.

[11] J. Gubb. Why are we waiting: an analysis of waiting times in the nhs. http://www.civitas.org.uk/nhs/waitingtimes_Jan_08.pdf.

[12] G. Hirsch and C. S. Immediato. Microworlds and generic structures as resources for integrating care and improving health. *System Dynamics Review*, 15(3):315–330, 1999.

[13] G. Hirsch and S. Immediato. Design of simulators to enhance learning: Examples from a health care microworld, quebec city, canada.

[14] M. Hoel and E. M. Sather. Public health care with waiting time: The role of supplementary private health care. *Journal of Health Economics*, 22(4):599–616, July 2003.

[15] J. Liu. Autonomy-oriented computing (aoc): The nature and implications of a paradigm for self-organized computing (keynote talk). 2008.

[16] J. Liu, X. Jin, and K. C. Tsui. *Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling*. Springer, 2005.

[17] J. Liu, H. Jing, and Y. Tang. Multi-agent oriented constraint satisfaction. *Artificial Intelligence*, 136(1):101–144, March 2008.

[18] J. Liu and K. C. Tsui. Toward nature-inspired computing. *Communications of the ACM*, 49(10):59–64, 2006.

[19] R. R. McDaniel and D. J. Driebe. Complexity science and health care management. *Advances in Health Care Management*, 2, 2001.

[20] R. E. Powell. Health care: A systems perspective. http://www.exponentialimprovement.com/cms/uploads/healthcare4.pdf.

[21] L. Siciliani. Does more choice reduce waiting times. *Health Economics*, 14, 2005.

[22] L. Siciliani and S. Martin. An empirical analysis of the impact of choice on waiting times. *Health Economics*, 16, 2007.

[23] A. van Ackere and P. C. Smith. Towards a macro model of national health service waiting lists. *System Dynamics Review*, 15(3):225–252, 1999.

Characterizing Multiplex Social Dynamics with Autonomy Oriented Computing

Lailei Huang

Abstract

The study of social dynamics and social networks is the key to reveal the mystery of how local human interactions might lead to diversified global social patterns. Traditional work on these fields generally neglects a key character of social interaction - the multiplexity effect or the fact that people interact through different social context with different purpose and the potential coupled result. To study this problem, we select, inspired by existing work, two specific social interaction contexts: (1) cultural interaction and (2) decentralized social search. Extensive work has been done to characterize dynamics in both of them. However, the existing system level, centralized computational and mathematic models cannot characterize the fundamental social mechanism that forms the real world pattern (i.e. the persistence of cultural diversity, the efficient navigability of social network) because they neglect two important characters of social entities: the local autonomous behavior (i.e. the independent decision making ability under different situation) and the interaction incentive. To characterize the multiplex effect and those limitations, we propose a computational framework based on the bottom-up, autonomy-oriented computing (AOC) paradigm. Under the framework, a baseline model including a simple multiplex coupling mechanism is implemented. The simulation-based experiments demonstrate that the local autonomous coupled behavior could generate great influence on the global performance such as the resource utilization efficiency.

1 Introduction

The problem of linking the gap between local interaction among social entities (individuals, groups, organizations, etc) and the global society level patterns lays in the center of sociology [7] as well as social behavior data mining and modeling [12]. Social dynamics [4] and social network analysis [27] [30] are two research fields focusing on this problem. The former models the dynamic change of individuals' status like binary opinion and the resulting group dynamics such as rumor spreading [4]. The latter analyzes the static or dynamic characters of social tie (e.g. local connectivity) and the global patterns of the social network such as small world and scale-free structure [3].

1.1 The Multiplex Social Dynamics

Traditional work that modeling social dynamics and social networks generally neglects [30] one of the key characters of social interaction - the multiplexity effect. Multiplex means a social tie could serve as the channel to support social interaction with different roles and purposes [9][23] at the same time. And the cumulated result within one social interaction context will generate influence on the other. Therefore we believe that the work modeling the local global problem that does not take the multiplex effect into account may fail to reveal the true picture of social interaction.

To be concrete, we focus our study of multiplex effect on two specific social interaction contexts: (1) *cultural dynamics* and (2) *decentralized social search*. Here, culture refers to things (opinion, behavior, etc) that people could influence on one another [2]. Culture dynamics means people with similar features tend to cluster together with time passing by. On the other hand, decentralized social search means people search and share social resources with each other, i.e. valued goods (wealth, information, expertise, etc) that is embedded in the personal network accessible through social tie [14].

We choose these two social interaction contexts for two reasons: (1) *rationality* because the relationship of them has been discussed in sociological literatures [21] [16][22][24] and (2) *comparability* because a great number of modeling work has devoted to study each dynamic process so that we may compare our result with them. To name a few, the Axelrod model [2] and the extensions [5][29] characterize the persistence of cultural diversities with the local interaction of entities under different situations. Kleinberg [10][11] proposes a mathematic model to explain the small world phenomenon [28] and more importantly the fact that people can find the short route in the social network based only on local view. Following work [15] observe the pattern predicted by an extension of the model from the real world data.

1.2 Challenges

Although the above mentioned work characterizes the local global problem computationally or mathematically, they do not consider the multiplex character of social dynamics. Besides, a key limitation of these work is neglecting two most important characters of social entity: the local autonomous behavior (i.e. how entity behave differently under different situations) and the interaction incentive (i.e. why certain interaction take place).

Specifically, in the agent-based model [2] as well as the extensions [5][29]. Every agent is *homogeneous* in their behaviors during cultural interaction, such as randomly partner selection. In other words, the agent is not *autonomous* or *intelligence*. Because of this, the entire system can be implemented by only one scheduler agent. This centralized modeling perspective limits these models in characterizing and explaining the richness of real-world local social interactions.

On the other hand, although the Kleinberg models [10][11][15] characterize the social search behavior in a distributed and localized manner, their models cannot answer the further questions like why social networks can self-organize themselves to such a form that efficient local search can be supported [12]. This is because (1) the local autonomous behavior in search is not accounted and (2) the *open*, *dynamic* characters of social networks are neglected.

1.3 Our Considerations

In this work, we consider human society as an *open*, *dynamic* complex system. Here, the openness means both endogenous and exogenous perturbations can happen, such as the *cultural drift* discussed in [2][5]. The dynamic property refers to the constant creation and dissolution of social relationships. Under this perspective, social dynamic processes such as cultural dynamics and decentralized social search happen as the result of the localized autonomous interaction among individuals.

To achieve this purpose, we propose a computational framework based on the bottom up autonomy-oriented computing (AOC) paradigm [17]. Based on the proposed AOC framework (we will refer to AOC-MSD framework below), the local autonomous behavior and interaction incentive of social entities can be explicitly defined. Besides, the openness and dynamic aspects of human society can also be modeled.

In this paper, we report the current research progress on building the AOC-MSD framework. A baseline model including a simple multiplex coupling mechanism is implemented. The simulation-based experiments demonstrate that, even under this simple mechanism, the local autonomous coupling behavior could generate great influence on the global performance such as the resource utilization efficiency.

The rest of the paper is organized as follows. In section 2, we summarize several related work on social dynamics and social networks modeling. The detailed problem statement is given in section 3. The autonomy-oriented computing framework is proposed in section 4 with the baseline model implemented in section 5. The preliminary experimental results are presented in section 6. In section 7 we draw the conclusion and point out the future directions.

2 Related Work

In this section, we first justify the interrelationship of cultural dynamics and social resource sharing dynamics by surveying sociological literatures. Then we highlight modeling work on each types of social dynamics.

2.1 Relationship of the Two Dynamic Processes

To infer the relationship between the dual dynamic processes, two questions can be naturally raised: (1) does the interpersonal similarity (i.e. sharing more cultural attributes) affect the resource mobilization and utilization? (2) does the social support process inversely affect the interpersonal cultural interaction?

The answer to the first question is definitely yes. Picklert et al. [24] demonstrates that similarities among people can foster mutual support. Lizardo [21] points out that cultural taste helps constructing social relations, establishing networks of trusting relations and facilitating resource mobilization. Lin also points out that ”greater similarity of shared resources may be indicated by the greater homophily among members” and vise versa [16]. Therefore, the local cultural interaction could generate effect on resource sharing behaviors.

The answer to the second question is, however, still elusive. McPherson et al. [22] calls for study the effect of multiplex social ties on the process of homophily interaction. However, neither empirical analysis nor theoretical modeling has been done to address this question. Therefore, it is an open question and we intend to address it in this research project.

2.2 Cultural Dynamics Modeling

The work on cultural dynamics mainly focuses on the following research question: why does the heterogeneity of cultural tastes among a group of people persist under the mutual reinforcement of homophily principle [22] and social influence?

The purpose of the Axelrod [2] model is to address this question. However, it is a centralized and system-level model in the sense that the entities in the model are *not autonomous*. All the behaviors are randomly based probability rules and they are *homogeneous* for all individuals. For example, the ego chooses the interaction partner uniformly randomly from its neighbors. Although under this assumption the core mechanism of homophily interaction and social influence can be emphasized, it is far from realistic social interaction scenarios. Therefore, other social mechanisms and external factors truly exists in people's local cultural interactions cannot be accounted.

The network co-evolution model [5] mentioned above is also centralized and top-down in this sense. For example, in the random rewiring rule proposed in [5] an individual randomly chooses another one in the entire population and adds a new link with him. Although in [29], the authors consider the heterogeneity of individual's contact accessibility, other aspects are similar to the Axelrod model thus no explicit autonomous behavior exists for an individual. This centralized modeling feature also hold by other extensions and variations of the Axelrod model such as [13][8]. A good survey of this line of models can be referred to [4].

2.3 Decentralized Social Search

Why does social networks, without any centralized control, could self-organize themselves to such a state that search strategy based only on local view can be adopted to efficiently locate faraway target (e.g. people far across the country [28], resource arbitrarily distributed [6])?

Kleinberg [10] proposes a mathematic model to study this problem. A model parameter r is used to tune the probability that any two individuals u and v on the two dimensional lattice with distance d is connected at long range. The result shows that when r equals 2, a decentralized search algorithm can make best use of the geographic information in the long range connections to do efficient search.

A following work by Liben-Nowell et al. [15] empirically observes the geographic information and friendship links within a dataset extracted from LiveJournal - an online social network site. They generalize the Kleinberg model by taking the non-uniform distribution of the population density into consideration. The data analysis result shows the the social network in LiveJournal indeed shows the navigability predicted by their model.

Although the above mathematical model and related simulation characterizes the efficient navigability of social network, they still have not solved the mystery raised ahead because we can still ask the question why people form friendship in such a form that efficient navigability can be achieved [12]? Therefore, the underlying factors that generate the navigability of social network is still a *fascinating open question* [12].

Recently, several work adopting the autonomy-oriented computing method [17] has deepened our understanding on this question. Specifically, Zhang and Liu [32] proposes a representative model of dynamic evolving social network from the perspective of service transaction. Following this view, Qiu and Liu [25] studies the issue of efficient decentralized search in the trust network. Based on the proposed trust relationship evolving mechanism, they find the network could self-organize itself to scale-free connection pattern as well as achieving efficient target finding.

In this work, we will follow the AOC method to formulate the social resource search and sharing dynamics. However, our work focuses on the multiplexity effect of social dynamics stated in detail below.

3 Research Problem Statement

As mentioned above, the purpose of this study is to examine the *multiplexity effect* of social dynamics. In current stage, we focus our attention on two specific social interaction contexts, namely, cultural dynamics and resource sharing dynamics. We need to answer the following three categories of questions so that the research problem can be specified.

1. To formalize the cultural dynamics, we modify the framework proposed in [2] by modeling the autonomous behavior so that more realistic interaction scenario can be captured. Specifically, the following questions should be addressed:
   - How to design the autonomous behavior rules in the cultural interaction i.e. partner selection? What kind of incentive is suitable to model the motivation of cultural interaction? How to formally define it? How do individuals update their incentives?
2. In order to characterize the social resource sharing process from the autonomous perspective, we need to formally answer:
   - How to model social resource? What is the resource sharing dynamics? How do individuals generate resource needs and enquiry messages? How does an enquiry information propagate in the social network? How do individuals make referral if they do not have the resources? How do they offer help if they have access to them?
3. Most importantly, to specify the term *multiplexity effect* we need to answer:
   - How to design a mechanism that couple the two types of social dynamics at the local interaction level?

- What are the measurements to characterize the global performance of cultural dynamics and resource sharing dynamics?
- How does the local coupling mechanism affect the global performance of each dynamic process?

To build the AOC-MSD framework to study these questions, we summarize them into the following tasks:

1. Modeling the two types of social dynamics which includes two tasks: (1) construct the individual's social interaction profile, i.e. individual attributes, autonomous behavior rules; (2) design the mechanism reflecting the coupling relationship of the two types of behaviors.
2. Measuring the evolution of cultural regions and the resulting cultural diversity. Examining the resource utilization efficiency with predefined measurement. Currently we use: (1) the average resources accessability; (2) the ratio of satisfied enquiry to the total number of enquiry messages.
3. Testing the way different cultural diversity affect the resource utilization efficiency. Inversely, examining the way different resource utilization affect the evolving of culture diversity.

4 The AOC-based Framework

4.1 Basic Ideas

As is mentioned above, we intend to characterize the *multiplex social dynamics problem* based on the methodology of AOC [18][19][17] so that the *distributed*, *autonomous* characters of the local social interaction and the *open*, *dynamic* features of the global social patterns can be modeled.

One of the key purposes of AOC is modeling complex system [17] from the bottom-up perspective. The building-blocks of the complex systems or entities autonomously interact with each other and the environment through the existing or dynamically changing coupling relationship. Through the locally incorporated self-organization computability, certain global patterns will emerge (e.g. web surfing regularities [20], HIV Immune patterns [31]).

In this AOC-based framework, we consider the human society as a complex system composed by social entities interacting with each other in cultural and resource sharing dynamic processes. In Section 4.2 we introduce the profile of social entities. The local autonomous behavior of entities is presented in Section 4.3. To characterize the multiplexity effect, the local behavior coupling is presented in Section 4.4.

4.2 Social Entity Profile

4.2.1 Basic Ideas

Active entities are the key elements of an AOC system because they represent the building-blocks of the complex system under consideration. In the AOC-MSD framework, an entity represents an individual involved in the cultural interactions and/or resource sharing interaction. Below we introduce the profile of an social entity.

Definition 1 *A social* ***entity*** e *represents an individual involved in social interactions, i.e. communicating mutual interests, sharing social resource. It can be formally defined as a tuple, i.e.* $e_i =< id, attribute, rules >$, *where* id *denotes the identifier,* $attribute = \{attr_{cul}, attr_{res}\}$ *represents the attribute of* e *related to culture and resource sharing dynamics respectively.* $rules = \{rule_{cul}, rule_{res}\}$ *denotes the behavior rules related to the two dynamic processes.*

The entity profile defined above demonstrates the idea that any entity can involve in both cultural and resource sharing dynamics. Therefore, they have states related to each of them, represented by $attr_{cul}$ and $attr_{res}$. Below, we introduces them respectively.

4.2.2 Cultural Interaction Attributes

Culture at the individual level can be understanded as the things that people can influence one another [2] such as movie taste. In the current framework, we follow the definition of culture proposed in [2] in which each entity hold a cultural vector, denoting the current states on each different cultural features. The cultural trait represents the state value of the entity on one feature. Formally, it could be defined as:

Definition 2 *The* ***Culture vector*** *of entity* e, *i.e.* $e.V_{culture} = [\sigma_1, \sigma_2, \ldots, \sigma_{F_c}]$ *has* F_c *number of* ***cultural features****. Each feature may stand for cultural related individual characters such as musical choice, clothing preference, reading interests, etc. Each feature has* q *possible* ***trait*** *values i.e.* $\sigma_i \in \{0, q-1\}$ *to represent the difference of individuals on that feature.*

Therefore, in the current framework, we have $attr_{cul} = V_{culture}$. Based on the above definition, we could further introduce the *cultural similarity* [2][5] to characterize how *similar* two entities are.

Definition 3 *The* ***culture similarity*** $sim(i,j)$ *denotes the overlap of the cultural feature value of two entities* e_i *and* e_j, *i.e.*

$$sim(i,j) = \frac{1}{F_c} \cdot \sum_{f=1}^{F_c} \delta(\sigma_{if}, \sigma_{jf}) \tag{1}$$

4.2.3 Resource Sharing Interaction Attributes

Social resource means valued goods (wealth, information, expertise) that is embedded in the personal network and accessible through direct neighbors or further referrals [14].

Similar to the $attr_{cul}$ entities have resource sharing related state in the AOC-MSD framework. It denotes whether or not an entity holds or can access certain social resource. Formally, it could be defined as:

Definition 4 ***Social resource*** *embedded in the social network is denoted as r_i, i.e. the ith resource type. The resource accessibility of an entity e is represented as* ***resource memory****, $e.M_{resource} = [r_1, \ldots, r_{R_c}]$,$r_i \in \{0, 1\}$. R_c is a parameter representing the number of possible resources in the artificial society.*

Therefore, in the current framework, we have $attr_{res} = M_{resource}$. The state variables of an entity e, i.e. $e.V_{culture}$ and $e.M_{resource}$ are dynamically changed by the local autonomous behaviors introduced below.

4.3 Local Autonomous Behaviors

4.3.1 Basic Ideas

In the AOC approach, the autonomy is reflected by the different local behavior rules triggered internally or externally by certain conditions [17]. In the AOC-MSD framework, the local behavior rules naturally related to the two social interaction contexts.

The list of the behaviors identified in culture and resource sharing dynamics are listed in Table 1. In this table, two columns correspond to two social interaction contexts, each of which contains three categories of behaviors to be specified below.

Table 1: Local behavior in two social interactions.

Cultural interaction	Resource sharing dynamics
initiating	resource enquiry
parter selection	information handling
homophily conform	enquiry checking

4.3.2 Cultural Interaction Behavior Rules

The cultural interaction related behavior rules are summarized in Table 2.

The first column denotes four types of cultural interaction initiating strategies. The first two types are intuitive. The *affinity* means each entity will decide whether or not to initiate an interaction according to its past cultural interaction results - or accumulated utility. It serves as the interaction incentive. In this framework, we name this utility as *affinity*. The rationality and definition will be given in section 4.4. R^{TH}_{happen} means each entity will decide whether to initiate a cultural interaction only after the average local sensed cultural similarity in its neighbors exceeds a threshold value.

Table 2: Local behavior strategies in cultural interaction.

Initiating	Partner selection	Homophily
always R^{AW}_{happen}	random R^{RD}_{select}	objective R^{obj}_{homo}
random R^{RD}_{happen}	affinity R^{AF}_{select}	subjective R^{sbj}_{homo}
affinity R^{AF}_{happen}	multiplexity R^{MP}_{select}	
threshold R^{TH}_{happen}		

The second column corresponds to the partner selection behavior. Three strategies are included. The random and affinity based rules are similar to the previous ones in first column. The multiplex strategy R^{MP}_{sel} means an entity e_i chooses an interaction partner not only depend on the current affinity but also affected by the result of the other types of social interaction.

The third column denotes how two entities perform the homophily interaction. $R^{obj}_{homophily}$ is used widely in the system-level model [2][5]. In this strategy, the entire cultural vector is used to calculate the cultural similarity. Compared with $R^{obj}_{homophily}$, the strategy $R^{sbj}_{homophily}$ means the entity only sense parts of its neighbor's cultural features. Besides, in this subjective perceive [26] rule, we could further modeling the biased perception as well as erroneous/fake perception.

4.3.3 Resource Sharing Behavior Rules

The detailed resource sharing related behaviors are summarized in Table 3.

Table 3: Local behaviors in resource sharing dynamics.

Resource enquiry	Information handling	Enquiry checking
happen	reply	utility updating
generating	referral	
parter selection		

The meaning of the behaviors in the first column is similar to the cultural behaviors. In the second column, R^{res}_{reply} denotes the feedback strategies if the entity holds the resource being requested. R^{res}_{trans} means that if the entity does not hold the enquired resource, whether and how will it transmit the enquiry. Finally, $R^{res}_{trust\_upd}$ denotes how the entity update the utility. The notion of *trust* is presented below.

4.4 Multiplex Coupling Relationship

4.4.1 Basic Ideas

In the AOC approach, *coupling* denotes certain constraint relationship among the entities and/or with the environment [17]. In this sense, the multiplexity character of social tie can be considered as, in essence, a *coupling relationship* among entities' local autonomous behavior in different dynamic processes. In this section, we present a *social tie strength* based multiplex coupling mechanism.

The basic assumption under this multiplex mechanism is: the strength of social tie is affected by the contact time and frequency [7] in different social interaction contexts, i.e. cultural interaction and resource sharing interaction. Inversely, the tie strength will further affect the interaction probabilities.

4.4.2 The Nature of Social Tie

In the AOC-MSD framework, the nature of social tie is the *cumulate results* of multiple social interactions. In the cultural dynamics, the like minded persons tend to cluster together and become more similar. The process naturally changes the *affinity relationship* among community members. Here, *affinity* means "ideas,ideals and causes shared by a tight community" [1].

On the other hand, the resource sharing dynamics demonstrates *why* and *how* people provide mutual support or different kinds of services to each other. Interpersonal *trust* can be taken as the natural production of this process. Here, *trust* reflects the belief or confident of ego that the relationship partner will behave as expected [9], e.g. provides service (e.g. resource sharing or referral) as needed [25].

Formally, the idea can be presented as following:

Definition 5 *The* ***affinity*** *of entity e_i to e_j, i.e. $e_i.affinity(e_j, t)$ denote how much rewards e_i has received from the past cultural interaction with the neighbor e_j. It can be calculated as:*

$$e_i.affinity(e_j, t) = \frac{1}{1 + e^{-suc_{ij}(t)}} \tag{2}$$

where $suc_{ij}(t)$ is the times of successful cultural interaction between e_i and e_j. Here successful *means e_i is successfully influenced by e_j at certain time t, i.e. change one of its cultural feature value.*

Definition 6 *The* ***trust*** *of entity e_i to e_j, i.e. $e_i.trust(e_j, t)$ denote the accumulated reward entity e_i has gained from past resource sharing with e_j, i.e.*

$$e_i.trust(e_j, t) = \frac{1}{1 + e^{-sat_{ij}(t)}} \tag{3}$$

where $sat_{ij}(t)$ is the satisfied enquiry times entity e_i get from entity e_j so far.

Definition 7 *The* ***tie strength*** *$e_i.tieStrength(e_j, t)$ perceived by e_i on e_j is determined by both the affinity accumulated in the cultural interaction as well as the trust formed in the past resource sharing dynamics, i.e.*

$$e_i.tieStrength(e_j, t) = \alpha \cdot e_i.affinity(e_j, t) + (1 - \alpha) \cdot e_i.trust(e_j, t) \tag{4}$$

where $\alpha \in [0, 1]$ is a coefficient to adjust the weight of cultural interaction as well as the resource sharing interaction.

After defining the multiplex social tie, we may adopt it in the local autonomous behavior of entities in either of the social dynamic processes. Currently, the baseline model introduced below does not implement this mechanism, we will explore it in the future work.

Finally, the overview of the framework is illustrated in Figure 1.

5 Baseline Model

In this section, we introduce a baseline model implemented under the AOC-MSD framework . The purpose is to take the AOC-by-Prototyping approach [18][19] to incrementally address the proposed research question.

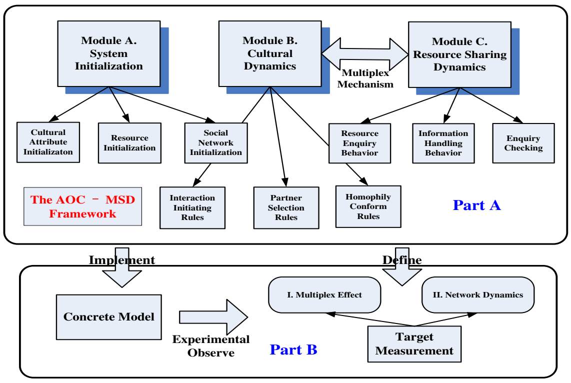


Figure 1: The AOC-MSD framework; In order to design a concrete model, three modules should be implemented: (1) initialization module, (2) cultural dynamics module and (3) resource sharing module. The target measurements focus on two aspects: (1) the multiplex effect and (2) the social network dynamics.

5.1 Cultural Dynamics Module

The model mechanism of cultural dynamics is demonstrated in Table 4. In the implementation, each social entity initiates the cultural interaction with one of its neighbors uniformly randomly chosen. The probability of homophily conform is proportional to the cultural similarity (definition3).

Table 4: Cultural dynamics module implementation. Detailed definition refers to section 4. Objective based rule means the probability of homophily interaction is based on the overlap of the cultural vectors.

Interaction Behavior	**Implementation**
Initiating behavior	Always initiate the culture interaction
Partner selection	Random choose a neighbor to interact
Homophily conform	Objective based rule

We have not included any autonomy mechanism in the cultural module in the baseline implementation such as affinity based initiating and partner selection rule. We intend to keep the model as simple as possible so that we may add one autonomous feature at a time and test the resulting effect.

5.2 Resource Sharing Dynamics Module

The model mechanism of resource sharing dynamics is shown in Table 5. The detailed meaning of the implemented behavior rules can be referred to section 4.

Table 5: Resource sharing dynamics module implementation. Detailed definition refers to section 4

Interaction Behavior	**Implementation**
Resource enquiry	R^{AW}_{happen}: happen at each time step R^{SG}_{gen}: single need each time step R^{RD}_{select}: random neighbor chosen
Information handling	R^{AW}_{reply}: always reply R^{SM}_{reply}: cultural similarity based R^{AW}_{trans}: always transmit
Enquiry checking	$R^{res}_{trust\_upd}$: not included

5.3 Multiplex Coupling Mechanism

In the baseline model, the tie strength coupling mechanism introduced in section 4.4 is not implemented. Instead, we design a simpler coupling mechanism including the following to aspects:

1. *Local profile based coupling* which reflects how the entities' state or local behavior in one dynamic process will affect those in the other type. More specifically, we make the assumption that entities *reply behavior* in social resource sharing interaction can take two possibilities: (1) always reply and (2) cultural similarity based reply. In the later one, the entities' cultural feature value will affect the resource sharing process. Experiment three below will address this.

2. *Global dynamics happen order* which reflects the happening order of the two dynamics. Two possibilities are: (1) *serial* coupled order and (2) *parallel* order. The former means the resource sharing dynamics happen after the convergence of cultural dynamics (i.e. no more local change of cultural feature value). The later means two dynamic processes will happen simultaneously. In the current baseline model, we implement the serial version.

5.4 System Initialization

To implement the baseline model, two further aspects should be mentioned:

- In the baseline model, we keep the initial social network structure as a two-dimensional lattice, like [2][5]. Besides, we do not incorporate any network dynamics, i.e. building and/or breaking social tie.

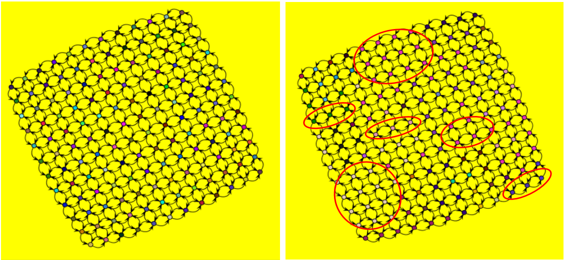
- Only one resource (i.e. R_a) is considered in the baseline model which is randomly distributed initially. We further introduce a model parameter eta to reflect the abundance of the resources (definition 4), i.e. initially there will be $N \cdot eta$ entities hold the resource.

6 Preliminary Experiments

In this section, corresponding to the above baseline model, we design three experiments that characterize: (1) cultural evolution (experiment one); (2) resource utilization (experiment two) and (3) multiplex coupling mechanism (experiment three). The model parameters used in these experiments is summarized in Table 6.

Table 6: The model parameters used in the following experiments

Symbol	**Description**
N	Number of social entities
F_c	Cultural feature number (definition 2)
q	Value range of one cultural feature (definition 2)
R_c	Number of social resource type (definition 4)
eta	The ratio of entities initially chosen as resource holder. It reflects the abundance of resources.
T_{lim}	Max enquiry message valid cycles (definition 4)



(a) A possible initial state T= 0 - random color

(b) In the middle run T≈10^3 – color regions emerge

Figure 2: Cultural region evolving, with different colors denote different cultural vector value. The result shows that (a) initially at $T = 0$ nearly all entities are different with each other; (b) at $T = 1000$ steps multiple cultural regions emerge.

6.1 Evolution of Cultural Diversity

Motivation In this experiment, we intend to observe how cultural regions evolve. The target measurement is *cultural diversity* which follows the definition in [2][5]:

Definition 8 *The **culture diversity** $C_{div} = \langle S_{max} \rangle / N$ denotes the average size of the largest cultural region normalized by the the number of entity N.*

Here, one a *culture region* denotes a connected part of the social network within which all entities have the identical cultural features and out of which the values are entirely different.

Experimental Setting To observe the cultural region evolving, we use $N = 100, F_c = 3, q = 10$ for ease of visualization. To observe the clear phase transition phenomenon, we set $F_c = 3$ and change $q \in \{0, 1, \ldots, 40\}$, $N \in \{100, 400, 900\}$.

Observation and Discussion Figure 2 demonstrates a representative run of cultural dynamics in the baseline model which shows that after 1000 cycles, cultural regions with identical features (colors) emerge. This can be explained by the positive feedback mechanism designed in the model: the more similar two individuals the more likely they will interact thus will become more similar. Therefore, the overall tendency in the cultural dynamics is the convergence of cultural feature values.

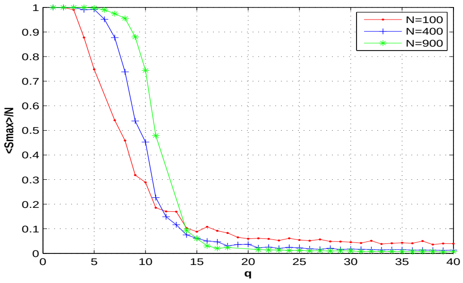


Figure 3: Phase transition of cultural diversity with the change of cultural trait value. With the decrease of q, the cultural diversity change from mono-state to diversity state in a sudden manner

Figure 3 illustrates the phase transition phenomenon observed in [2]. The result demonstrates that for fixed cultural feature value F_c, slightly change of trait range or q will lead to a dramatic change of the final cultural region. Besides, small q value will lead to the mono-cultural region while large q value will lead to the highly cultural diversity.

6.2 Resource Utilization Efficiency

Motivation In this experiment, we intend to observe the global performance of resource sharing dynamics. The target measurement is defined as follows:

Definition 9 *The **resource access ratio** of an entity at time t or $R^i_{acc}(t)$ is the number of resources entity e_i could access at time t proportion to the total number of resources in the artificial society. The **average resource access ratio** is the average of R^i_{acc} on all entities.*

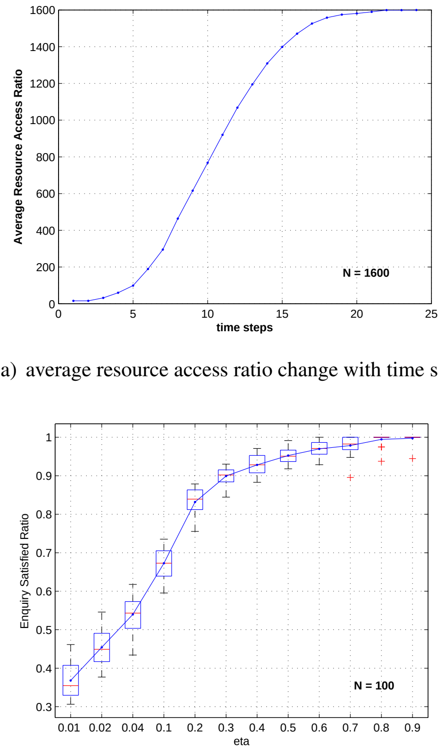
Intuitively, the more resources accessible by all people in the social network, the more utilization efficiency of the society. Another intuitive aspect is related to how many enquiries are satisfied defined as:

Definition 10 *The **enquiry satisfied ratio** S_{eq} is the fraction of all satisfied enquiry messages to the total generated enquiry messages.*

Experimental Setting To observe the resource access ratio, we use $N = 1600, R_c = 1, eta = 1\%$. To examine the enquiry satisfied ratio under different resource abundance, we use $N = 100, R_c = 1, eta \in \{0.01, 0.02, 0.04, 0.1, 0.2, \ldots, 0.9\}$. In both scenario $T_{lim} \sim Noraml(2, 5)$.

Observation and Discussion Figure 4(a) shows that all entities could access the resource R_a finally under the experimental setting. Besides, the convergence curve exhibits the sigmoid shape.

The first observation is not plausible in the sense that in reality, some entities may not be able to access the resource no matter how many enquiries are made through his local social circle. Besides, we conjecture that the sigmoid curve is due to a positive feedback in the baseline model setting. A model hypothesis is once an entity gets the accessability of the resource, it can further offer it to others. As a result, the more entities can access R_a the quicker the rest entities



(a) average resource access ratio change with time steps

(b) enquiry satisfied ratio change with resource abundance parameter *eta*

Figure 4: Resource utilization efficacy:(a)all entities could access the resource finally,(b)the more abundance of resource the more enquiry messages are satisfied

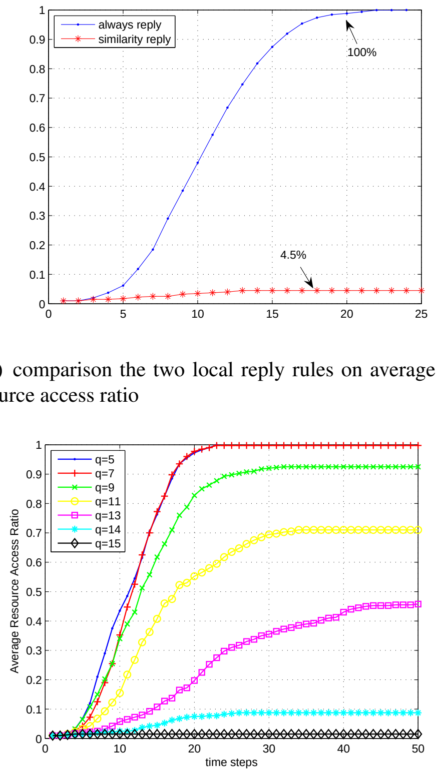
can access it. This observation leads us to further consider the **type** of resource to be modeled, i.e. the information or expertise resource is rather different from the material based resource.

The result in Figure 4(b) is trivial in the sense that the more resource abundance in the society, the easier an enquiry can be satisfied. However, an efficient resource utilization society should be, under certain model mechanism, less abundance resource can still satisfy most resource enquiries. We plan to address this issue in the future model implementations.

6.3 Multiplex Coupling Effect

Motivation In this experiment, we intend to test the multiplex coupled mechanism implemented in the baseline model.

Experimental Setting To examine the effect of local profile based coupling we use $N = 400, F_c = 1, q = 15, R_c = 1, eta = 1\%, T_{lim} \sim Normal(2, 5)$ in the first part of the experiment. To test the effect of cultural diversity on the resource sharing, we change $q \in \{5, 7, 9, 11, 13, 14, 15\}$ in the second part.



(a) comparison the two local reply rules on average resource access ratio

(b) comparison the effect of different cultural diversity on average resource access ratio

Figure 5: The effect of cultural similarity based reply rule on resource utilization:(a)entities using the *always reply* rule could quickly access the resources while with *cultural similarity based reply* only 4.5% could access the resource,(b)with the increase of q, the order parameter control cultural diversity, the average resource access ratio decrease quickly.

Observation and Discussion From Figure 5(a), we may observe the final average resource access ratio is inhibited under the cultural similarity based reply rule. We conjecture the reason for the inhibition can be explained as follows: (1) because of the serial dynamics happen order, the resource sharing dynamics will happen on cultural region that has already formed (see experiment one); (2) the resource is initially randomly distributed, therefore entities in one cultural region may not be able to access the resource located in the other regions because of the cultural boundary.

In order to validate the above explanation, we change the cultural trait range value q and observe the effect of cultural diversity on the average resource utilization. As we have observed in Figure 3, there is a phase transition in the maximal cultural region size by increasing the q value. Therefore, we may expect that, with the increase of q value, the effect of inhibition will also be increased. Figure 5(b) demonstrates this conjecture. More than 90% entities could access the resource in the society when q is smaller than 9. However, when q increases to 14 the proportion decreases to less than 10%.

7 Conclusion and Future Work

In this work, we propose an autonomy-oriented computing framework, the AOC-MSD framework, to study the multiplexity effect of social dynamics, namely the multiple channels of social interactions and the coupling results of the dynamics processes. Specifically, we formalize the local autonomous behaviors in cultural taste based social interaction and resource sharing based social interaction. In this framework, a social tie strength based coupling mechanism is proposed to characterize the interaction incentive and multiplex nature of social interaction.

A baseline model based on the AOC-MSD framework is implemented and three preliminary experiments have been done to characterize: (1) the cultural evolution, (2) the resource sharing efficiency and (3) the multiplexity effect. Experimental results show that the cultural similarity based reply rule can generate significant influence on the resource utilization efficiency. Specifically, the more cultural diversity exists in the society, the less possibility community members could access the resources in need.

Our future work will mainly focus on the following aspects: (1) refining the design of entity's resource profile and the resource initialization rule, (2) implementing the tie strength based multiplex mechanism and examining the coupling effect by conducting systematic experiments, (3) incorporating the network dynamics in the local autonomous behaviors to capture the creation and dissolution of social relationship and observe its effect on the evolving of cultural diversity and the resource utilization efficiency.

References

[1] Affinity. *http://en.wikipedia.org/wiki/Affinity_(sociology).*

[2] R. Axelrod. The dissemination of culture. *Journal of Conflict Resolution*, 41(2):203–226, 1997.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[4] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 348, 2008.

[5] D. Centola, J. C. Gonzalez-Avella, V. M. Eguiluz, and M. S. Miguel. Homophily, cultural drift and the co-evolution of cultural groups. *Journal of Conflict Resolution*, 51(6):905–929, 2007.

[6] R. Crane, M. Cebrian, G. Pickard, W. Pan, and A. Madan. Mit floats ideas in darpa balloon challenge. *http://news.cnet.com/8301-1023_3-10411211-93.html*, 2009.

[7] M. S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78, 1973.

[8] P. Holme and M. Newman. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E*, 74, 2006.

[9] R. J.Lewicki, D. J.Mcallister, and R. J. Bies. Trust and distrust:new relationships and realities. *Academy of Management Review*, 23(3):438–458, 1998.

[10] J. Kleinberg. The small-world phenomenon:an algorithmic perspective. *In Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.

[11] J. Kleinberg. Complex networks and decentralized search algorithms. *Proceedings of the International Congress of Mathematicians (ICM)*, 2006.

[12] J. Kleinberg. The convergence of social and technological network. *Communications of the ACM*, 51(11):66–72, 2008.

[13] K. Klemm, V. M.Eguiluz, R. Toral, and M. S. Miguel. Nonequilibrium transitions in complex networks: a model of social interaction. *Physical Review E*, 67, 2003.

[14] G. Lai, N. Lin, and S.-Y. Leung. Network resources, contact resources, and status attainment. *Social Networks*, 20:159–178, 1998.

[15] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *PNAS*, 102(33):11623–11628, 2005.

[16] N. Lin. *Social Capital. International Encyclopedia of Economic Sociology. Edited by Jens Beckert and Milan Zagiroski*. Routledge, London, 2006.

[17] J. Liu. Autonomy-oriented computing(aoc): The nature and implications of a paradigm for self-organized computing(keynote talk). *in Proceeding of the 4th International Conference on Natural Computation(ICNC'08) and 5th International Conference on Fuzzy Systems and Knowledge Discovery(FSKD'08)*, 2008.

[18] J. Liu, X. Jin, and K. C. Tsui. Autonomy oriented computing: From problem solving to complex systems modeling. *Multiagent Systems, Aritificial Societies, and Simulated Organizations. Springer, Secaucus, NJ, USA*, 2004.

[19] J. Liu, X. Jin, and K. C. Tsui. Autonomy oriented computing(aoc): Formulating computational systems with autonomous components. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(6):879–902, 2005.

[20] J. Liu, S. Zhang, and J. Yang. Characterizing web usage regularities with information foraging agents. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 16:566–584, 2004.

[21] O. Lizardo. How cultural tastes shape personal networks. *American Sociological Review*, 71(5):778–807, 2006.

[22] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[23] L. M.Verbrugge. Multiplexity in adult friendships. *Social Forces*, 57(4):1286–1309, 1979.

[24] G. Plickert, R. R.Cote, and B. Wellman. It's not who you know, it's how you know them:who exchanges what with whom. *Scoial Networks*, 29(3):405–429, 2007.

[25] H. Qiu, J. Liu, and N. Zhong. A dynamic trust network for autonomy-oriented partner finding. *AMT'09*, pages 323–334, 2009.

[26] E. M. Rogers and D. K. Bhowmik. Homophily-heterophily:relational concepts for communication research. *The Public Opinion Quarterly*, 34(4):523–538, 1970-1971.

[27] W. S and F. K. *Social Network Analysis:Methods and Applications*. Cambridge Univ. Press, Cambridge, 1994.

[28] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[29] B. Wang, Y. Han, L. Chen, and K. Aihara. Limited ability driven phase transitions in the coevolution process in axelrod's model. *Physics Letter A*, 2009.

[30] D. J. Watts. A twenty-first century science. *Nature*, 445:489, 2007.

[31] S. Zhang and J. Liu. A massively multi-agent system for discovering hiv-immune interaction dynamics. *MMAS04,LNAI*, 3446:161–173, 2005.

[32] S. Zhang and J. Liu. Autonomy-oriented social networks modeling: Discovering the dynamics of emergent structure and performance. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(4):611–638, 2007.

Estimate the Number of Relevant Images in Infinite Databases Using Two-Order Markov Chain

Xiaoling Wang

Abstract

In image retrieval the most commonly used performance measures are precision and recall. However, to determine the number of relevant images in an infinite database presents a significant challenge as the relevant parameters are not directly observable. In our research, we use internet as a vehicle to investigate this problem, and evaluate search results from major Web Image Search Engines (ISEs). We also investigate whether the cumulative relevance of images in different results pages follows particular stochastic behaviors, such as Two-Order Markov chain whose probability of event occurrence depends on not only a current state but also the immediate past state transitions. From such model, we shall estimate the total number of relevant images for major image search engines.

1 Introduction and Related Work

Due to the increased importance of the Internet, the use of image search engines such as Google, Yahoo, and Msn is becoming increasingly widespread. However, for many web ISEs, it is difficult for users to make a decision as to which web ISE should be selected. It is obvious that the more effective the system is, the more it will offer satisfaction to the users. Therefore, retrieval effectiveness [1], [2], [3], [4] becomes one of the most important parameters to measure the performance of web image retrieval systems [5], [6], [7], [8], [9], [10], [11], [12]. As we know, the most commonly used performance measures are the precision P and recall R [10], [13], [14], [15], [16], [17], [18], [19], [20], [21], but to compute recall R is rather difficult as the total number of relevant images is not directly observable in such a potentially infinite database.

Many researchers have conducted studies to evaluate the retrieval effectiveness of web ISEs. Ece Çakır et al. [7] described the retrieval effectiveness of image search engines based on various query topics. Fuat Uluç et al. [14] described the impact of the number of query words on image search engines. However, none of these studies describe how to estimate the total number of relevant images for the image search engines. All of them only view the first two page results. In the study by Sprink and Jansen [16], data collected from Dogpile was analyzed and one of the findings was that the percentages of the users that viewed only the first page and those that viewed only the first two pages of document search results were about 71% and 15.8%, respectively. Although many works used recall as the measure to evaluate the image search engines, not many papers work on the estimation of the number of relevant images in infinite databases. An algorithm called sample-resample is presented in by Si and Callan [15]; in environments containing resource descriptions already created by query-based sampling, the sample-resample method uses several additional queries to provide an estimate of the database size. Therefore, if the database size has been known, then the distribution of relevant images can be estimated.

In our paper, we model the probabilistic behavior of the distribution of relevant images among the returned results by evaluating the performance of some widespread web ISEs.

In next section, we introduce how to apply Two-Order Markov Chain Model [23], [24], [25], [26] to image retrieval and describe the queries selection. The experimental results and the validation of the models will be discussed in section 3. Finally we summarize our works, and present some directions of future work in the last section.

2 Basic Model and Queries Selection

2.1 Two-Order Markov Chain Model

Since in internet image search, results are returned in units of pages, we shall focus on the integer-valued stochastic process $X_1, X_2, \ldots$, where X_J represents the aggregate relevance of all the images in page J, which may be estimated by

$$X_J = \sum_{i \in J} Z_{Ji} \tag{1}$$

where $Z_{Ji} = 1$ if the i^{th} image on page J is relevant, and $Z_{Ji} = 0$ if the i^{th} image on page J is not relevant.

As to this stochastic modeling of cumulative page image relevance, we shall investigate in particular the Two-Order Markov Chain Model whose probability of event occurrence depends on not only a current state but also the immediate past state transitions. Therefore, a Markov chain of order m where m is finite, is a process satisfying

$$Pr\{X_n = x_n | X_{n-1} = x_{n-1}, \cdots, X_1 = x_1\}$$
$$= Pr\{X_n = x_n | X_{n-1} = x_{n-1}, \cdots, X_{n-m} = x_{n-m}\} \quad (2)$$

Where m is less than n.
In other words, the future state depends on the past m states. It is possible to construct a chain (Y_n) from (X_n) which has the 'classical' Markov property as follows:
Let $Y_n = (X_n, X_{n-1}, \ldots, X_{n-m+1})$, the ordered m-tuple of X values. Then Y_n is a Markov chain with state space S^m and has the classical Markov property.

In our paper, we estimate the relevant images for web ISEs based on Two-Order Markov chain. Therefore, according to what have described above, Y_n is a special sequence, represented as
$Y_n = X_n X_{n-1}, n = 1, 2, 3, \ldots,$
Where Y_n is a Markov chain with a state space S^2 and has the classical Markov property.
The transition probability is a process satisfying

$$Pr\{Y_n = y_n | Y_{n-1} = y_{n-1}\}$$
$$= Pr\{X_n X_{n-1} = x_n x_{n-1} | X_{n-1} X_{n-2} = x_{n-1} x_{n-2}\} \quad (3)$$

And the transition probability is satisfying the condition

$$Pr\{X_n X_{n-1} = x_n x_{n-1} | X_{n-1} X_{n-2} = x_{n-1} x_{n-2}\}$$
$$= \begin{cases} 0, & \text{if } x_{n-1} \neq x_{n-1} \\ \geq 0, & \text{if } x_{n-1} = x_{n-1} \end{cases} \quad (4)$$

Based on our previous study, the state space for One-Order Markov chain model is S={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}. The state space for the Two-Order Markov chain should be S^m corresponding, where m is equal to 2. Therefore, the original state vector is a 1-by-441 matrix and the transition probability matrix is a 441-by-441 matrix. We can also effectively estimate the initial probabilities if the sample is large enough. The probabilities are placed in a vector of state probabilities:

$$\pi(J-1, J) = (\pi_{00}, \pi_{01}, \pi_{02}, \pi_{03}, \cdots, \pi_{ij}), \quad (5)$$

Where π_{ij} is the probability having i relevant images in page J and having j relevant images in page $J-1$.
Therefore, from this model, we can estimate the number of relevant images by pages by using the formula in page J:

$$\pi(J-1, J) = \pi(J-1, J-2) * P, J = 2, 3, \ldots, n. \quad (6)$$

Then according to the original state vector and transition probability matrix, we could estimate the total number of relevant images for web ISEs.

2.2 Testing Image Search Engine and Queries Selection

We choose Google, Yahoo and Msn as the testing ISEs. Because from [22], the total market share of Google, Yahoo and Msn are 90.2%, namely, Google [27], Yahoo [28], and Msn [29], whose market shares are 64%, 16.3%, and 9.9%, respectively.
Google (www.google.com)
Yahoo (www.yahoo.com)
Msn (www.bing.com) (new msn image search engine)

In our experiment, total 100 queries are used, which 70% queries are provided by the authors and 30% of them are the popular search suggestion term observed from Google when you submit a keyword. 70% queries consists of one-word, two-word and more than three-word queries, which range from simple words like apple to more specific query like apple computers and finally progressing to rather specific

Table 1: Part of Sample Query list

Categories	Sample Queries
One-word Queries	Apple Dolphin Octopus Facebook Roxy Wildlife Skiing Alleyway Maldives Puppy
Two-word Queries	Apple Computer Plane Crash Octopus Card Outer Space Night Scene Daisy Flower Street-Art Baby Cry Afghan Child Twin Towers
Three-word Queries	Man Wearing Hat Macro Fly Eyes Sunrise and Sunset Jordan Basketball Nike Black and White Portrait HongKong Night Scene Flowing in the Wind Michael Schumacher Ferrari Chinese Opera Mask Victoria Harbour HongKong

Table 2: Test Query List

Categories	Test Queries
One-word Queries	Bangkok Stockholm Parkour
Two-word Queries	Sahara Desert Solar System Liu Xiang Abstract Smoke
Three-word Queries	Clown Fish and Sea Anemone Couple Silhouette at Sunset Mexico City Skyline

search, such as the query may be contain three or more elements. While 30% queries consists of the popular suggestion term with the lowest returned results and the largest returned results when we submit a keyword what the Google shows us. A part of selected queries are given in Table 1 and Table 2 shows us the testing queries.

3 Experimental Results

In our experiment, 100 queries are submitted to each of ISE, and we record the number of relevant images page by page. Based on Spink and Jansen's study [12], evaluating the images in first two pages is enough. Such a finding seems useful for the users who only want to find less than forty images. However, it could not satisfy the users' need who wants to search much more images. Therefore, the number of relevant images in the first ten pages is recorded for each query and we will model the Two-Order Markov Chain Model to help us to estimate the number of relevant images per page based on the recorded data. In the following section, we will discuss the results by applying the Two-Order Markov chain model we mentioned in section 2.

3.1 Two-Order Markov Chain Model

Fig. 1 illustrates the actual results returned by test ISE for all the test queries and the estimating results by using Two-Order Markov Chain Model we obtain from the example queries for Google. According to this figure, we can see that the Two-Order Markov Chain Model fits the test queries well, no matter for the one-word query, two-word query or

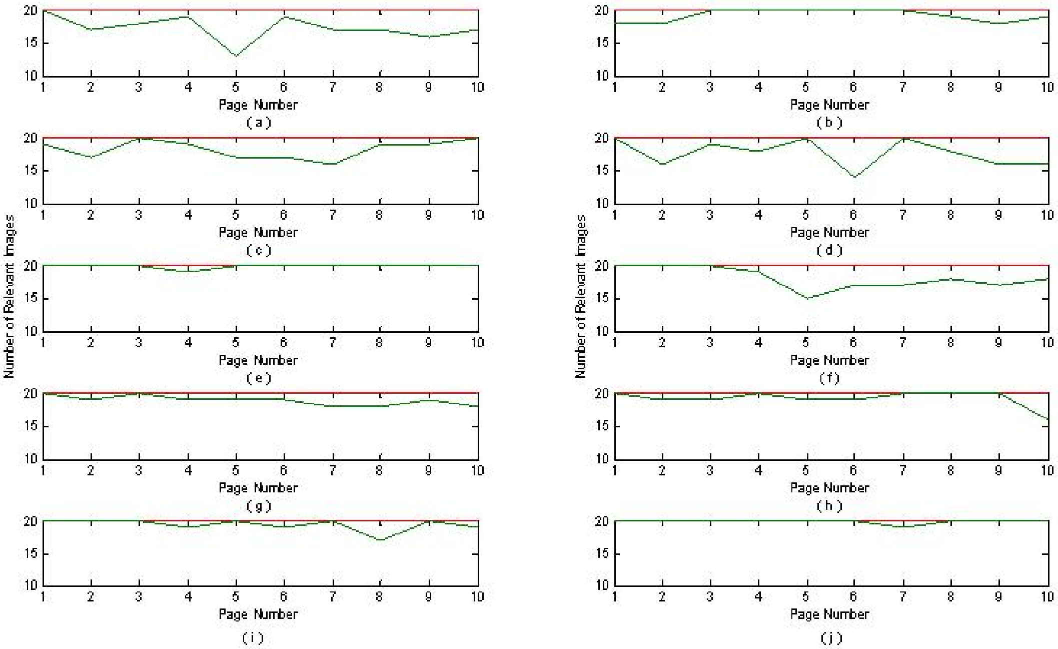


Figure 1: Test results of different queries for Google (a)bangkok. (b)stockholm. (c) parkour. (d) sahara desert. (e) solar system. (f) liu xiang. (g) abstract smoke. (h) clown fish and sea anemone. (i)couple silhouette at sunset. (j)mexico city skyline

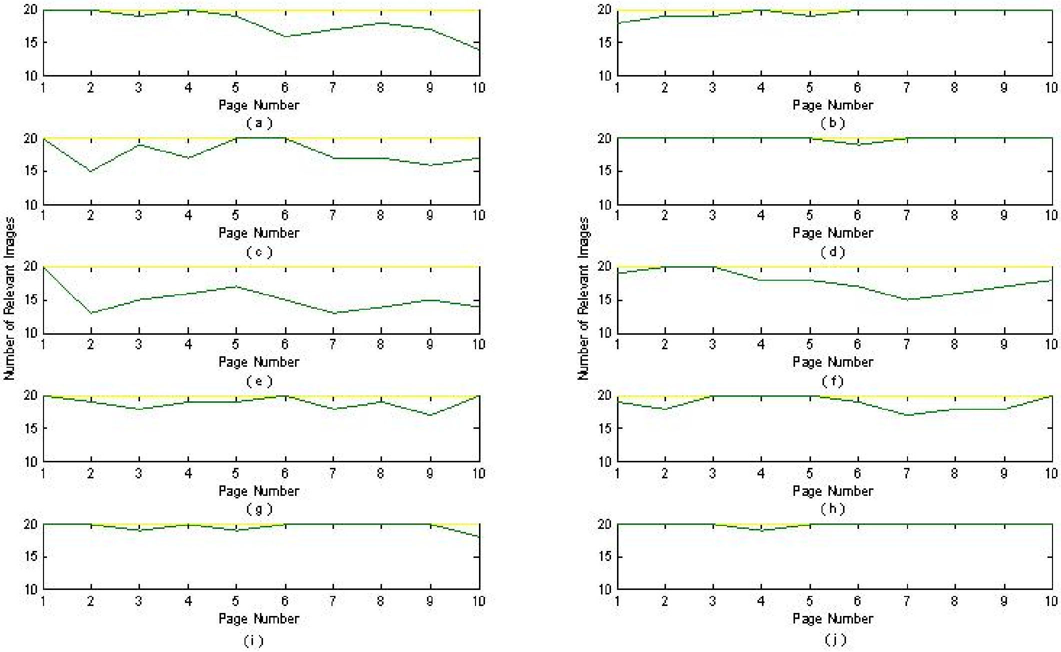


Figure 2: Test results of different queries for Yahoo (a)bangkok. (b)stockholm. (c) parkour. (d) sahara desert. (e) solar system. (f) liu xiang. (g) abstract smoke. (h) clown fish and sea anemone. (i)couple silhouette at sunset. (j)mexico city skyline

three-word query. Although there are some distinctions, the distinctions are not large except for several points. But apparently, the actual results are smaller than the predicting results. However, all of this are based on human judgement, in order to be more precise, we will use MAE to measure whether we can use such a model to estimate the total number of relevant images for the web ISE.

Fig. 2 provides the actual results of different test query for ISE Yahoo. The Figure shows that the Two-Order Markov Chain Model also fits the test queries quite good.Especially for the query called Stockholm, the model fits the data perfectly. And as the results we observed from Google, the actual results are smaller than the predicting results too. But apparently for query Solar System, the model fits the actual results quite bad. However, for most test queries, it is a good predicting model based on human judgment.

Fig. 3 gives the test results of different test query and the predict results given by Two-Order Markov Chain Model for ISE Msn. The Figure tells us that the Two-Order Markov Chain Model fits the test queries also well for the test queries. Meanwhile, it appear that the trend between the actual results and predict results seems quite the same. But for the query couple silhouette at sunset, it seems that we can't use such a model to predict the number of relevant images. Therefore, the model that we investigated is not a perfect model which can perfectly suitable for every query. Finally, we will also use MAE to measure whether we can use such a model to estimate the total number of relevant images for the web ISE.

3.2 Measure of Forecast Accuracy

The mean absolute error (MAE) is a common measure of forecast error in time series analysis. MAE is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. In the previous section, we just judge the accuracy of the models we investigated based on human judgment; however, human judgment is not very precise and well persuade. Therefore, in this section, we will use MAE to measure the Two-Order Markov Chain Model to judge whether the model we investigated can well estimate the number of relevant images per page for major ISEs.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}|e_i| \tag{7}$$

As we know, the smaller the MAE the better the model is. In our paper, that means the smaller the MAE, the better

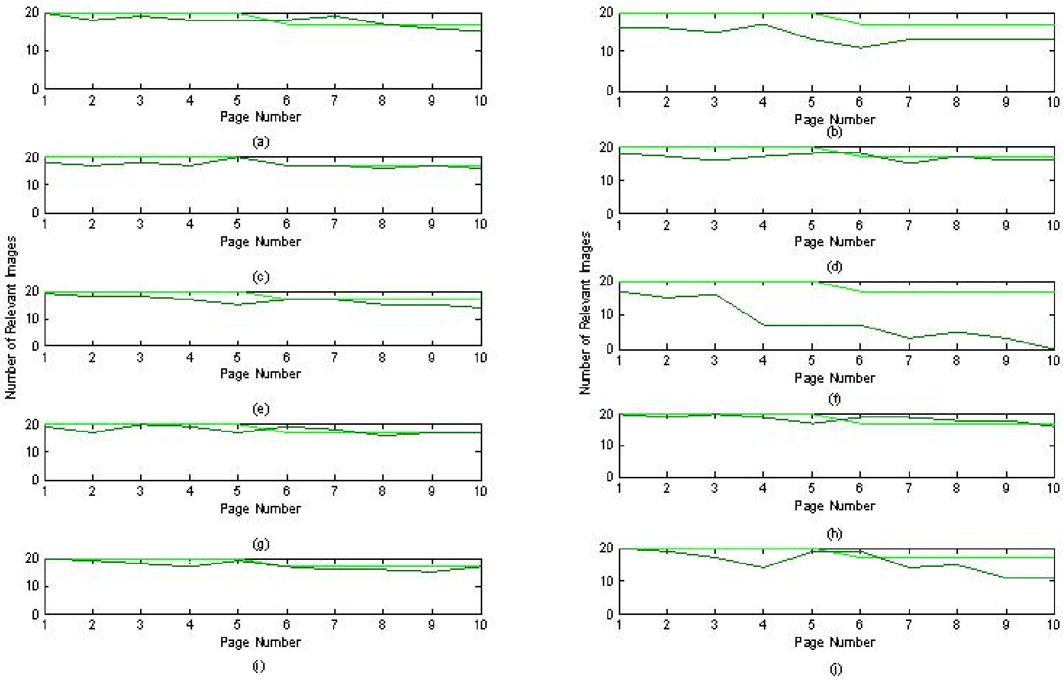


Figure 3: Test results of different queries for Msn (a)bangkok. (b)stockholm. (c) parkour. (d) sahara desert. (e) solar system. (f) liu xiang. (g) abstract smoke. (h) clown fish and sea anemone. (i)couple silhouette at sunset. (j)mexico city skyline

Table 3: MAE for Different Models and Different Search Engines

MAE		Two-Order MC MODEL		
		Google	Yahoo	Msn
One-word Queries	Bangkok	2.7	2.0	1.3
	Stockholm	2.3	0.1	1.9
	Parkour	1.1	1.1	1.2
Two-word Queries	Sahara Desert	0.8	0.5	4.5
	Solar System	0.1	4.8	2.0
	Liu Xiang	0.8	1.1	1.2
	Abstract Smoke	0.1	0.1	3.0
More than three-word Queries	Clown Fish and Sea Anemone	1.7	2.2	1.2
	Couple Silhouette at Sunset	1.9	2.2	10.5
	Mexico City Skyline	0.6	0.4	1.1
MMAE		1.21	1.45	2.79

the model can well estimate the number of relevant images for the web ISEs. The table 3 shows us the results of measurement of forecast accuracy for the model.

According to Table 3, for the Two-Order MC Model, 73.3 percent of MAE is smaller than 2.0, which means the percentage of most deviation between the actual results and the forecast results is less than 10 percent. Therefore, we could conclude that the Two-Order MC Model is good for us to use them to estimate the number of the relevant images for web ISEs. Meanwhile, according to the MAE, we could say that we can use the Two-Order MC Model to estimate the number of relevant images for the web ISEs, because all the MAE listed in the Table 3 is small enough. But there are exceptions; the MAE of query Couple Silhouette at Sunset is 10.5 which are quite large.

4 Conclusion and Future Work

Currently, estimating the number of relevant images in the infinite image search engines is quite hard, but it is so important for us. Therefore, we develop a set of image queries to investigate models to estimate the number of relevant images in infinite ISEs. And using some queries to validate the model we obtain.

In this paper we applied the Two-Order Markov Chain to investigate the probabilistic behavior of the distribution of relevant images among the returned results for the major ISEs. We have carried out experiments based on the returned results of 100 training queries and 10 test queries. For Two-Order Markov Chain Distribution, we were able to construct the transition probability matrix and the original state probability. Then apply the Two-Order Markov Chain to calculate the probability of all probable number of relevant images. Finally we picked up the largest probability and regarded its corresponding number as the number of relevant images of current page. We found that Two-Order Markov Chain Model could well present the distribution of relevant images among the returned results for the major ISEs and the percentage of deviation is less than 10 percent.

Future work includes considering whether the cumulative relevance of images in different results pages follows hidden Markov chain.

References

[1] P. Over, C.H.C. Leung, H. Ip, and M. Grubinger, ”Multimedia Retrieval Benchmarks,” *IEEE Multimedia,* vol. 11, no. 80-84, Apr. 2004

[2] Nikhil V. Shirahatti, Kobus Barnard, ”Evaluating Image Retrieval,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05),* vol. 1, pp. 955-961, 2005

[3] J.R. Smith, ”Image Retrieval Evaluation,” *IEEE Workshop on Content - Based Access of Image and Video Libraries (CBAIVL),* pp. 112, 1998

[4] Marchand Maillet, S. and Worring, M. ”Benchmarking image and video retrieval: an overview.” In *Proceedings of the 8th ACM international Workshop on Multimedia information Retrieval (MIR ’06),* pp. 297-300, 2006

[5] Jaroslav Pokorný, ”Web Searching and Information Retrieval,” *Computing in Science and Engineering,* vol. 6, no. 4, pp. 43-48, July/Aug, 2004

[6] K. Stevenson and C.H.C. Leung, ”Comparative Evaluation of Web Image Search Engines for Multimedia Applications”, In *Proceedings of IEEE International Conference on Multimedia and Expo,* July 2005

[7] Ece Çakır, Hüseyin Bahçeci, Yıltan Bitirim, ”An Evaluation of Major Image Search Engines on Various Query Topics.” In *Proceedings of the Third International Conference on Internet Monitoring and Protection (ICIMP ’08),* pp. 161-165, 2008.

[8] Münevver Tuğçe Elagöz , Mehtap Mendeli , Remziye Zeden Manioğluları , Yıltan Bitirim, ”An Empirical Evaluation on Meta-Image Search Engines,” In *Proceedings of the Third International Conference on Digital Telecommunications (ICDT ’08),* pp. 135-139, 2008.

[9] Tumer, D., Shah, M.A., and Bitirim, Y. , ”An Empirical Evaluation on Semantic Search Performance of Keyword-Based and Semantic Search Engines: Google, Yahoo, Msn and Hakia.” In *Proceedings of the 2009 Fourth international Conference on internet Monitoring and Protection (ICIMP ’09),* pp. 51-55, 2009.

[10] Ishioka, T., ”Evaluation of Criteria for Information Retrieval.” In *Proceedings of the 2003 IEEE/WIC international Conference on Web intelligence,IEEE Computer Society,* pp. 425, 2003

[11] Jiang, H., ”Study on the Performance Measure of Information Retrieval Models.” In *Proceedings of the 2009 international Symposium on intelligent Ubiquitous Computing and Education (IUCE ’09),* pp. 436-439, 2009.

[12] Demirci, R.G., Kişmir, V., and Bitirim, Y., ”An Evaluation of Popular Search Engines on Finding Turkish Documents.” In *Proceedings of the Second international Conference on internet and Web Applications and Services (ICIW ’07),* pp. 61, 2007

[13] Raquel Kolitski Stasiu, Carlos A. Heuser, and Robertoda Silva, ”Estimating Recall and Precision for Vague Queries in Databases”, In *Proceedings of Springer-Verlag Berlin Heidelberg(CAISE ’05),* LNCS 3520, pp. 187-200, 2005

[14] F. Uluç, E. Emirzade and Y. Bitirim, ”The Impact of Number of Query Words on Image Search Engines”, In *Proceedings of Second International Conference on*

Internet and Web Applications and Services (ICIW '07), pp. 50, 2007

[15] Si, L. and Callan, J., "Relevant document distribution estimation method for resource selection." In *Proceedings of the 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval (SIGIR '03),* pp. 298-305, 2003

[16] A. Sprink and B.J. Jansen, "Searching Multimedia Federated Content Web Collections," Online Information Review, vol.30, no.5, pp. 485-495, 2006

[17] Jin, X. and French, J.C., "Improving image retrieval effectiveness via multiple queries." In *Proceedings of the 1st ACM international Workshop on Multimedia Databases (MMDB '03),* pp. 86-93, 2003

[18] Black Jr, J.A., Fahmy, G., and Panchanathan, S., "A Method for Evaluating the Performance of Content-Based Image Retrieval Systems." In *Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and interpretation (SSIAI '02),* pp. 96, 2002

[19] Nottelmann, H. and Fuhr, N., "Evaluating different methods of estimating retrieval quality for resource selection." In *Proceedings of the 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval (SIGIR '03),* pp. 290-297, 2003

[20] Hauptmann, A.G. and Christel, M.G., "Successful approaches in the TREC video retrieval evaluations." In *Proceedings of the 12th Annual ACM international Conference on Multimedia (MULTIMEDIA '04),* pp. 668-675, 2004

[21] Longzhuang Li, Yi Shang, "A new statistical method for performance evaluation of search engines." In *Proceedings of the 12th IEEE international Conference on Tools with Artificial intelligence (ICTAI '00),* pp. 208, 2000

[22] Nielsen NetRatings, "Google accounts for more than half of all web searches, while Yahoo and Msn combined account more than one fourth, according to Nielson Netrating", April 2009. http://www.polepositionmarketing.com/emp/april-2009-search-engine-2/

[23] Thomas W. Miller, *Data and Text Mining: A Business Applications Approach.* Upper Saddle River, N.J.: Pearson Prentice Hall, c2005

[24] W.S. Kendall, F. Liang, J.S. Wang, *Markov chain Monte Carlo: innovations and applications,* Singapore; Hackensack, NJ: World Scientific, c2005

[25] Render, R., R.M. Stair Jr., and M.E. Hanna. *Quantitative Analysis for Management-Chapter 16,* 10th Edition, Prentice Hall, 2009

[26] Dani Gamerman, *Markov chain Monte Carlo: stochastic simulation for Bayesian inference,*London; New York : Chapman and Hall, 1997

[27] Google Search Engine `www.google.com`

[28] Yahoo Search Engine `www.yahoo.com`

[29] Msn Search Engine `www.bing.com`

A Local Region-Based Approach For Lip Contour Extraction Using Localized Active Contour Model

Xin Liu

Abstract

Lip contour extraction is crucial to the success of a lipreading system. This paper presents a local region-based approach for lip contour extraction using the localized active contour model. The proposed approach utilizes a combined semi-ellipse as the initial evolving curve to split the local neighborhoods into local-interior and local-exterior respectively, and then computes the localized energy for evolving and extracting. This method is robust against the noise, rotation, deformation and teeth appearance, and not limited to a fixed lip model. Experiments show its promising result in comparison with the existing methods.

1 Introduction

Lip contour extraction has been extensively studied in recent years [2, 13, 7]. It is one of the most important techniques for human-machine interface applications such as lip reading [10], audio-visual speech recognition [8] and facial expression analysis [6]. Nevertheless, it is a non-trivial task to find a robust and accurate method for lip contour extraction due to large variations caused by different speakers, noise, illumination conditions, low contrast between lip and skin, high deformable level of lips, and so forth.

In the past decade, a number of techniques have been proposed to achieve lip contour extraction, which can be categorized into two major classes: the edge-based approach and the model-based approach. The edge-based method mainly utilizes the low level spatial cues such as color and edges to achieve lip localization and extraction [13]. Often, the performance of such a method will deteriorate when there is a poor contrast between lip and surrounding skin regions. In contrast, the model-based approach, which builds a lip model with a small set of parameters, generally outperforms the former one. Examples include deformable template (DT) [3] [7] and active shape model (ASM) [9]. The DT algorithm utilizes a parametric model to describe the lip contour, which is, however, sensitive to the deformation and irregularity of the lips. The ASM utilizes a series of landmark points, which are controlled within a few modes derived from a training data, to describe the lip shape. In general, the training process of the ASM is quite time-consuming. Besides the DT and ASM methods, the active contour model (ACM) or snakes [2] is another typical example along this line. The conventional ACM allows an initial contour to deform by minimizing a specific global energy function to produce the desired segmentation. Paper [2] has shown the success of this method in its application domain, but this method is somewhat sensitive to the parameter initialization and the image noise.

Furthermore, when objects have heterogeneous statistics, it is found that the localized active contour model(LACM) [4] can generally achieve a satisfactory segmentation results while the conventional ACM fails. In the LACM, the evolving curve splits the local neighborhoods into local-interior and local-exterior respectively. Subsequently, the localized energy for evolving and extracting can be computed. However, improper parameters such as large radius or far away evolving curve in LACM can lead to the wrong extracting results. See the Fig. 1 and Fig. 2 for the examples.

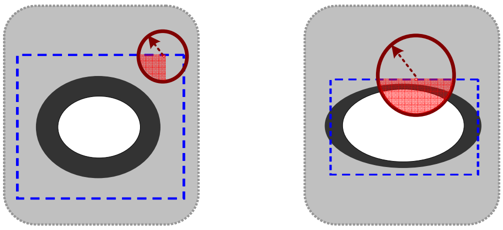


Figure 1. Improper parameters expression.(a)ulterior evolving curve with small local radius, (b)proper evolving curve with large local radius.

In this paper, we propose a local region-based approach for lip contour extraction method using the localized active contour model. We find a combined semi-ellipse as the initial evolving curve which can be fitted well in the LACM, meanwhile, the proper parameters in LACM can be automatic selected. Experiments have shown the promising results of the proposed algorithm in comparison with the existing methods.

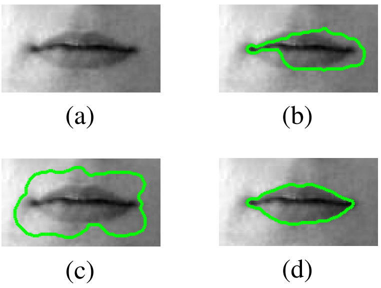


Figure 2. (a) Lip image with noise affects, (b) Conventional ACM based extracting result, (c) LACM based extracting result with improper parameters, (d) LACM based extracting result with proper parameters.

2. Overview of LACM

This section will overview the framework of LACM [4], in which the only assumption is that the foreground and background regions will be locally different.

The statistical analysis of local regions leads to the construction of a group of local energies about each point on the evolving curve, in order to optimize these local energies in its own local region, each point is considered individually, consequently, the point's component of the local energy is computed by splitting the local neighborhoods into local-interior and local-exterior using the evolving curve.

In this paper, I denotes a pre-specified image defined on the domain Ω, C denotes a closed contour represented as the zero level set of a signed distance function ϕ, i.e., $C = \{u\,|\phi(u) = 0\}$ [4]. The interior of C is specified by the following approximation of the smoothed Heaviside function:

$$\mathcal{H}\phi(u) = \begin{cases} 1, & \phi(u) < -\varepsilon \\ 0, & \phi(u) > \varepsilon \\ \frac{1}{2}\left\{1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\phi(u)}{\varepsilon}\right)\right\}, & otherwise. \end{cases} \quad (1)$$

Similarly, the exterior C can be defined as $(1 - \mathcal{H}\phi(u))$.

The derivative of $\mathcal{H}\phi(u)$, a smoothed version of the Dirac delta is used to specify the area adjacent to the curve.

$$\delta\phi(u) = \begin{cases} 1, & \phi(u) = 0 \\ 0, & |\phi(u)| < \varepsilon \\ \frac{1}{2\varepsilon}\left\{1 + \cos\left(\frac{\pi\phi(u)}{\varepsilon}\right)\right\}, & otherwise. \end{cases} \quad (2)$$

Parameters u and v are expressed as independent spatial variables to represent a single point in Ω, respectively. Using this notation, the characteristic function $\mathcal{B}(u, v)$ marked the local regions in terms of a radius parameter r can be described as follows:

$$\mathcal{B}(u, v) = \begin{cases} 1, & \|u - v\| < r \\ 0, & otherwise. \end{cases} \quad (3)$$

Using $\mathcal{B}(u, v)$, we define an energy functional in terms of a generic internal energy functional F, the resulting energy functional $\mathrm{E}(\phi)$:

$$\mathrm{E}(\phi) = \int_{\Omega_u} \delta\phi(u) \int_{\Omega_v} \mathcal{B}(u, v) \cdot F(I(v), \phi(v))dudv, \quad (4)$$

where the functional F is a generic internal energy measure used to represent local adherence to a given model at each point along the contour. This energy relies on the assumption that foreground and background regions should have maximally separate mean intensities which can cause the curve to move.

Therefore, a localized region-based energy formed from the global energy by substituting local means for global ones is shown here [12]:

$$F = -(\mu_{in}(u) - \mu_{out}(u))^2, \quad (5)$$

$$\mu_{in}(u) = \frac{\int_{\Omega_v} \mathcal{B}(u, v) \cdot \mathcal{H}\phi(v) \cdot I(v)dv}{\int_{\Omega_v} \mathcal{B}(u, v) \cdot \mathcal{H}\phi(v)dv}, \quad (6)$$

$$\mu_{out}(u) = \frac{\int_{\Omega_v} \mathcal{B}(u, v) \cdot (1 - \mathcal{H}\phi(v)) \cdot I(v)dv}{\int_{\Omega_v} \mathcal{B}(u, v) \cdot (1 - \mathcal{H}\phi(v))dv}, \quad (7)$$

where the localized versions of the means $\mu_{in}(u)$ and $\mu_{out}(u)$ represent the intensity mean in local interior and exterior regions around a point u, respectively.

By ignoring the image complexity that may arise outside local region, only contributions from the points within the radius r of the contour are considered. Finally, for the purpose of keeping the curve smooth, a regularization term is added as is commonly done in active contour segmentation energies. Meanwhile, the arclength of the curve is penalized and weighted by a parameter λ, and the final energy $\mathrm{E}(\phi)$ is given in the following:

$$\mathrm{E}(\phi) = \int_{\Omega_u} \delta\phi(u) \int_{\Omega_v} \mathcal{B}(u, v) \cdot F(I(v), \phi(v))dvdu + \lambda \int_{\Omega_u} \delta\phi(u) \left\|\nabla(u)\right\|du. \quad (8)$$

By taking the first variation of this energy with respect to ϕ, the following evolution equation is obtained:

$$\frac{\partial\phi}{\partial t}(u) = \delta\phi(u) \int_{\Omega_v} \mathcal{B}(u, v) \cdot \nabla_{\phi(v)} F(I(v), \phi(v))dv + \lambda\delta\phi(u)div(\frac{\nabla\phi(u)}{|\nabla\phi(u)|}) \left\|\nabla\phi(u)\right\|. \quad (9)$$

It is certainly noteworthy that, any region-based segmentation energy can be put into this framework.

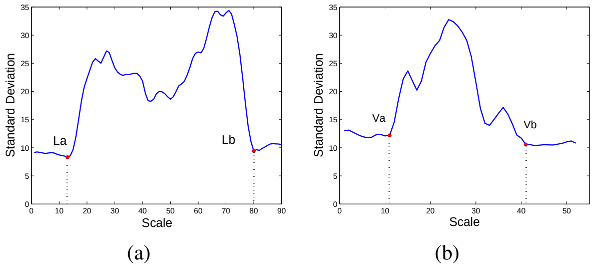


Figure 3. Standard deviation of columns and rows.

3. The Proposed Algorithm

Our proposed automatic lip contour extraction system includes an initialization step and a lip contour extraction process.

3.1 Initialization

Empirical studies have found that a lip shape is usually close to an rectangular region [5] [7]. Furthermore, it can also be approximatively surrounded by various of semi ellipses. How to find a combined semi-ellipse as the initial evolving curve is of crucial importance to extract the lip contours in our method.

For the purpose of finding the a combined semi-ellipse of lip region, the detection of lip corner dots is needed. Specifically, $I(x, y)$ represents a pixel value at coordinate (x, y), m, n express the maximum values of rows and columns. The left corner, right corner, up corner, down corner are denoted as La, Lb, Va, Vb.

We project the RGB-based lip image into the gray-level one, from the practical viewpoint, it is inevitable that there exists the noises or uneven illumination affects. Hence, each lip image is performed with a 3 × 3 mean filter and a contrast stretching adjustment.

According to the statistical methods, the really lip regions are usually of the big standard deviation value. Therefore, the lip corner dots of horizontal or vertical can be detected by computing the first and last value of standard deviation , which is changing obviously compared with the adjacent ones. For example, in order to find the horizontal lip corner columns quickly, we can use the following equations:

$$mean_j = \frac{1}{m}\sum_{i=1}^{m}(I(i,j)), j = 1, 2, \cdots n, \qquad (10)$$

$$Std_j = \left(\frac{1}{m}\sum_{i=1}^{m}(I(i,j) - mean_j)^2\right)^{\frac{1}{2}}, \qquad (11)$$

$$|Std_j - Std_{j+2}| < \Delta S, |Std_j - Std_{j-2}| < \Delta S, \qquad (12)$$

where ΔS is a changing threshold. Fig. 3 and Fig. 4gives an example. We can easily obtain the coordinate value of La_x, Lb_x. Afterwards, the value of La_y and Lb_y can be computed through the mean coordinate with the minimum gray value of $I(La_x, j), i = 1, \cdots, n$. The other approximative corner dots can also be computed through the above method.

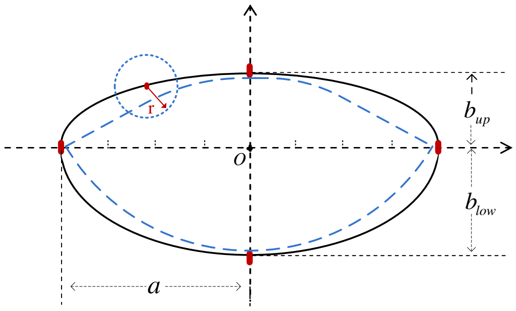


Figure 4. A combined semi-ellipse around the lip.

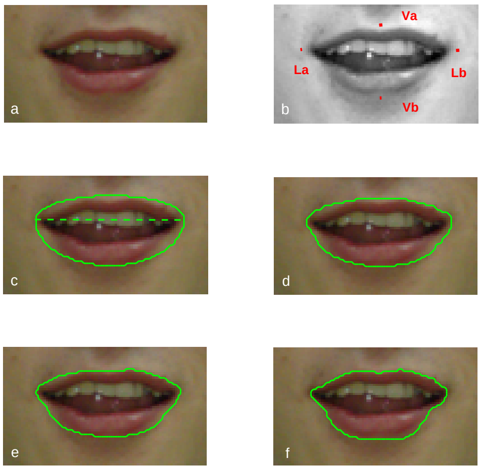


Figure 5. The extracting procedure. (a) lip image, (b) lip corner dots, (c) the combined semi-ellipse, (d) after 10 iterations, (d) after 20 iterations, (d) after 30 iterations.

Let the (x_c, y_c) be the origin center of the combined semi-ellipse, through which the mathematical equations are as follows:

$$x_c = \frac{1}{2}(La_x + Lb_x), y_c = \frac{1}{2}(La_y + Lb_y),$$

$$\theta = \arctan\left(\frac{Lb_y - La_y}{Lb_x - La_x}\right),$$

$$a = \frac{1}{2}((Lb_x - La_x)^2 + (Lb_y - La_y)^2)^{\frac{1}{2}},$$

$$b_{up} = ((Va_x - x_c)^2 + (Va_y - y_c)^2)^{\frac{1}{2}},$$

$$b_{low} = ((Vb_x - x_c)^2 + (Vb_y - y_c)^2)^{\frac{1}{2}},$$

$$X = (x - x_c) \cdot \cos\theta + (y - y_c) \cdot \sin\theta,$$
$$Y = (y - y_c) \cdot \cos\theta - (x - x_c) \cdot \sin\theta,$$

$$\frac{X_{up}^2}{a^2} + \frac{Y_{up}^2}{b_{up}^2} = 1, \frac{X_{low}^2}{a^2} + \frac{Y_{low}^2}{b_{low}^2} = 1, \qquad (13)$$

where a is the semi-major axes, b_{up} and b_{low} are the up and low semi-minor axes, respectively. θ is the inclined angle, which is positively defined in the counter-clockwise direction. θ is the inclined angle , and it is defined to be positive in the counter-clockwise direction.

3.2 Contour Extraction

After successfully finding the combined semi-ellipse of the lip region, we can let it be the evolving curve represented the zero level set C just as is stated in part 2, which can be fitted well in the LACM. Subsequently, local neighborhoods of the points can be split into local interior and local exterior by the evolving curve.

By computing the local energies at each point along the curve, the evolving curve will deform so as to minimizing the local energies in order to produce the desired lip segmentation. See Fig. 5 for the details.

The concise extraction steps are as follows:

- Locate the lip region, preprocess;
- Obtain the combined semi-ellipse;
- Evolve with iteration;
- Extract the lip contours.

The radius r selected by the function $\mathcal{B}(u, v)$ is an important parameter in LACM. By rule of thumb, $r = \frac{r_b}{2}$ is appropriate in most cases.

4. Experimental Result

We have applied our approach to the 500 frontal face images with the different mouth shapes. The database consists of 200 face images from the CVL face database [11], 200 face images from the GTAV face database [1] and 100 lip images from our laboratorial database. In our experiments, we set the parameter λ is equal to 0.3.

Examples of lip contour extraction are shown in Fig. 6 and Fig. 7. It can be clearly seen that, the accurate lip contours can be extracted using our algorithm. Table 1 presents the details in comparison with other two existing methods, in which, we define the extracted performance as $\frac{n_c}{n}$, where the n_c is the correct extracted numbers, n is the total numbers of the test database.

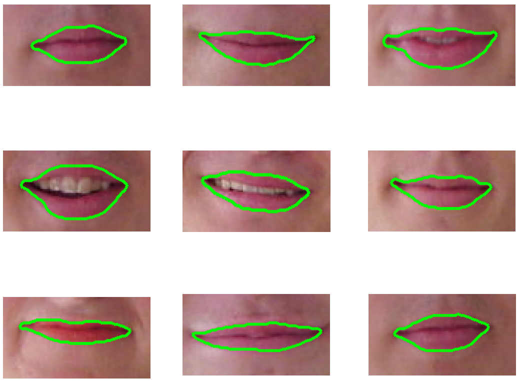


Figure 6. The extracted results of lip images form the CVL database by our proposed approach.

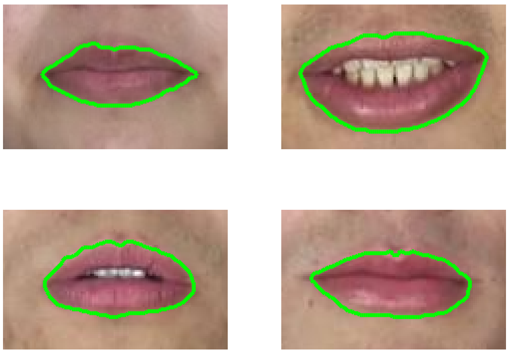


Figure 7. The extracted results of lip images from our laboratory database by our proposed approach.

Table 1. Compared with the existing methods.

Method	Lip model	Automatic	Performance
ACM	No	No	78.6%
DT	Yes	No	92.6%
Our approach	No	Yes	96.4%

As is shown above, the deformable or irregular lip contours can be extracted successfully using the proposed method, which also has a better extracting performance. Furthermore, the proposed approach is not limited to a fixed lip model, meanwhile , it is usually more tolerant to the affects of noise, rotation and the deformation. More important, it is also worth noting that our proposed algorithm can reach completely automation, though which the combined semi-ellipse can be found as the initial evolving curve in LACM.

We have also examined the unsatisfactory results (less than 4%), and found that they all have the very poor contrast between the lip and skin region, or have obvious beard effects around the lips.

5. Conclusion

This paper has proposed an automatic and robust lip contour extraction algorithm using localized active contour model. We obtain a combined semi-ellipse as the initial evolving curve through which an optimum extraction of the lip image into lip and non-lip regions can be found. This algorithm is robust against the the noise, rotation , deformation and the appearance of teeth.

References

[1] http://gps-tsc.upc.es/gtav/researchareas/upcfacedatabase /gtavfacedatabase.htm.

[2] G. Chiou and H. Jenq Neng. Lipreading from color video. *IEEE Transactions on Image Processing*, 6(8):1192–1195, 1997.

[3] T. Coianiz, L. Torresani, and B. Caprile. 2d deformable models for visual speech analysis. *NATO Advanced Study Institute: Speechreading by Man and Machine*, 150:391–398, 1995.

[4] S. Lankton and A. Tannenbaum. Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039, 2008.

[5] M. Li and Y. M. Cheung. Automatic lip localization under face illumination with shadow consideration. *Signal Process*, 89(12):2425–2434, 2009.

[6] J. J. Lien, T. Kanade, J. F. Cohn, and L. Ching-Chung. Automated facial expression recognition based on facs action units. In *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 390–395, 1998.

[7] A. W. C. Liew, S. H. Leung, and W. H. Lau. Lip contour extraction from color images using a deformable model. *Pattern Recognition*, 35(12):2949–2962, 2002.

[8] P. Liu and Z. Wang. Visual information assisted mandarin large vocabulary continuous speech recognition. In *Proc. of International Conference on Natural Language Processing and Knowledge Engineering*, pages 72–77, 2003.

[9] J. Luettin, N. Thacker, and S. Beet. Visual speech recognition using active shape models and hidden markov models. In *Proc. of ICASSP*, volume 2, pages 817–820, 1996.

[10] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, 2002.

[11] F. Solina, P. Peer, B. Batagelj, S. Juvan, and J. Kovac. Color-based face detection in the "15 seconds of fame" art installation. In *Proc. Mirage, Conf. Computer Vision/Computer Graphics Collaboration for Model-Based Imaging, Rendering, Image Analysis and Graphical Special Effects,Rocquencourt, France, Mar.10-11*, pages 38–47, 2003.

[12] A. Yezzi, A. Tsai, and A. Willsky. A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13:195–216, 2002.

[13] X. Zhang, R. M. Mersereau, M. Clements, and C. C. Broun. Visual speech feature extraction for improved speech recognition. In *Proc. of ICASSP*, volume 2, pages 1993–1996, 2002.

Automatic Segmentation of Color Lip Images Based on Morphological Filter

Meng Li
Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China

Abstract

This paper addresses the problem of lip segmentation in color space that is a crucial issue to the success of a lip-reading system. We present a new segmentation approach to lip contour extraction by taking account of the color difference between the lip and skin in the color spaces. Firstly, we obtain a lip segment sample via the distinction between the lip and skin in 1976 CIELAB color space. Secondly, we establish a probability model in HSV color space and make use of the segment sample so that the membership of lip and non-lip region is calculated. Thirdly, we employ a morphological filter to obtain the lip counter candidate based on the two memberships. Finally, we extract the lip contour via convex hull algorithm with the prior knowledge of the mouth shape. Experiments show the efficacy of the proposed approach in comparison with the existing lip segmentation methods.

1. Introduction

In the past decade, the lip segmentation has received considerable attention from the community because of its wide applications in audio-visual speech recognition, biometric person identification, lip synchronization, human expression recognition, and so forth [1, 2, 3, 4]. In general, lip segmentation is a non-trivial task because the color difference between the lip and the skin regions is not so noticeable sometimes. In particular, it becomes more challenging when the illuminations in the environment are complex.

In the literature, a few image segmentation techniques have been proposed. One class of methods is based on the clustering with color features [1, 5, 6] provided that the number of clusters (e.g. the clusters of skin and lip) is known in advance. Unfortunately, the hair, moustache and the visibility of teeth and tongue in the mouth opening generally require that the number of clusters is selected adaptively. Consequently, such a method is unable to operate fully automatically [7]. Another class of widely-used methods is model-based ones, such as Active Shape Model, Active Appearance Model, Active Contour Model (Snake), and so forth [1, 8, 9, 10, 11]. Specifically, they build a deformable model for lip by learning the patterns of variability from a training set of correctly annotated images. The shape of model can be adjusted by a parameter set so as to match and locate the lip in test images. Empirical studies have shown the success in their application domain, but they need to label some landmarks manually for training. Recently, some automatic segmentation approaches to lip images have been developed. For example, [12] and [13] utilize a color transformation or color filter to enlarge the difference between the lip and skin. Paper [7] utilizes the multi-scale wavelet to detect the edge of lip.

In this paper, we will present a new method for the automatic segmentation of lip images provided that the lower part of a face (i.e. the part between nostril and chin) has been available. The proposed method employs the color transformations so as to enlarge the difference between the lips and the skin like the existing methods, but the new method extracts a segment of lip only rather than the whole lip contour. Specifically, we firstly obtain a lip segment sample via the distinction between the lip and skin in 1976 CIELAB color space. Secondly, we establish a probability model in HSV color space and make use of the segment sample so that the membership of lip and non-lip region is calculated. Thirdly, we employ a morphological filter to obtain the lip counter candidate based on the two memberships. Finally, we extract the lip contour via convex hull algorithm with the prior knowledge of the mouth shape. Since our method is based upon the color diversity between skin and lip region rather than “absolute” color [5], its performance is stable and accurate on the images with different color temperatures and the testers with different skin or lip colors. Experiments have shown the efficacy of the proposed approach in comparison with the existing lip segmentation methods.

The remainder of this paper is organized as follows. We describe the calculation of lip membership in Section 2, and show the lip contour extraction in Section 3. In Section 4, we will conduct the experiment to empirically compare the proposed method with the existing ones. Finally, we draw a conclusion in Section 5.

2 Lip Membership Based on Color Space Transformation

It is desirable to work in a color space (a sample of original image is illustrated in Figure 1), in which the lip color (i.e. relative red) out of others can be highlighted. Since the value of a^* channel in 1976 CIELAB color space can determine the color component between red/magenta and green, i.e. the small values indicate green while the large values indicate magenta, we therefore convert the source image into 1976 CIELAB color space and normalize the a^* component to the range of $[0, 255]$, denoted as I_{a^*}. Furthermore, we utilize Eq.(1) as proposed in [12] to convert the source image to the range of $[0, 255]$ with equalization, denoted as $I_{G/R}$:

$$I_{G/R} = \begin{cases} 256 \times \frac{G}{R} & R > G \\ 255 & otherwise. \end{cases} \quad (1)$$

Let $I_{sub} = I_{a^*} - I_{G/R}$.[1] Subsequently, we can establish a Gaussian model for I_{sub} based on the gray-level value for each non-zero pixel with the mean $\hat{\mu}_{sub}$ and the standard deviation $\hat{\sigma}_{su}$. The candidate of lip segments can be obtained by

$$I_{candidate} = \begin{cases} 0 & I_{sub} \leq \hat{\mu}_{sub} - 2\hat{\sigma}_{sub} \\ I_{G/R} & otherwise. \end{cases} \quad (2)$$

In Eq. (2), it is found that most non-black points are in the lip region. Hence, to eliminate those non-black parts outside the lip region, we utilize Eq.(3) and Eq.(4) to calculate the gravity center, denoted as (g_x, g_y):

$$g_x = \frac{\sum_{x=1}^{col} \sum_{y=1}^{row} x I_{candidate}(x, y)}{\sum_{x=1}^{col} \sum_{y=1}^{row} I_{candidate}(x, y)} \quad (3)$$

$$g_y = \frac{\sum_{x=1}^{col} \sum_{y=1}^{row} y I_{candidate}(x, y)}{\sum_{x=1}^{col} \sum_{y=1}^{row} I_{candidate}(x, y)}, \quad (4)$$

where row and col denote the vertical and horizontal size of the image in pixel, respectively. We extract the nearest non-black part to the gravity center, which corresponds to the lip segment as shown in Figure 2. Please note that it is enough to extract a part of lip rather than the whole lip region because the extracted lip segment is used for sample data so as to establish a probability model.

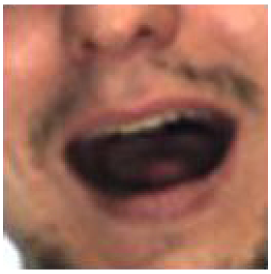


Figure 1. The original lip image, which is the source image of the subsequent figures.

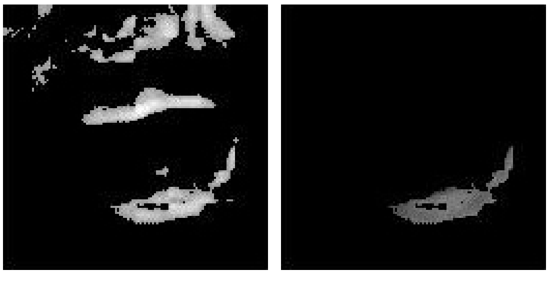


Figure 2. A sample image of (a) $I_{candidate}$, and (b) the extracted lip segment. In $I_{candidate}$, most non-black parts correspond to lip segments in the source image. The extracted lip segment is a part of the lip.

Subsequently, pixels in source lip image belonging to the lip segment area extracted above are converted into HSV color space. For each pixel, we let:

$$C_1 = H \cdot cos(2\pi \cdot S) \quad (5)$$

$$C_2 = H \cdot sin(2\pi \cdot S). \quad (6)$$

As a result, we obtain a 2-dimensional sample vector denoted as $C_{Seg} = (C_1, C_2)$.

We establish a probability model as follows:

$$M_{lip} = \frac{1}{2\pi\sqrt{\hat{\Sigma}}} \cdot exp(-\frac{(C_{Src} - \hat{\mu})\hat{\Sigma}^{-1}(C_{Src} - \hat{\mu})^T}{2}) \quad (7)$$

[1] In this paper, all equations are employed in positive area. That is, as long as a result is negative, it will be set at 0 automatically.

where C_{Src} is a 2-dimensional value for arbitrary pixel in source lip image. The parameters $\hat{\mu}$ and $\hat{\Sigma}$ can be estimated via the following equations:

$$\hat{\mu} = \frac{\sum_{i=1}^{n} C_{Seg}^{i}}{n} \tag{8}$$

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^{n} (C_{Seg}^{i} - \hat{\mu})(C_{Seg}^{i} - \hat{\mu})^{T}. \tag{9}$$

Based on the probability model of Eq. (7), we can calculate the lip membership for each pixel in a source image.

Similarly, based on the black area of $I_{candidate}$, we can also establish a probability model to calculate the non-lip membership denoted as $M_{non-lip}$. Figure 3 gives an example of the two membership maps, in which the high membership corresponds to the light area, and vice versa. Moreover, considering the convenience of visibility, we project the membership from $[0, 1]$ to $[0, 255]$, i.e., the gray scale.

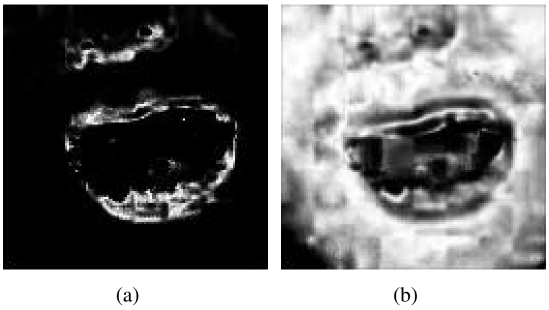


Figure 3. A sample image of (a) lip membership map, (b) non-lip membership map. The high membership corresponds to the light area, and vice versa.

3 Lip Counter Extraction

Obtain a mask by letting

$$Mask = 255 - M_{non-lip} - I_{G/R}. \tag{10}$$

Moreover, the lip membership is considered as marker. Thus, the morphological reconstruction operation proposed in [14] can be employed. Figure 4 (a), (b) and (c) illustrate the marker, mask and result, respectively.

We utilize a gray-level threshold selection method proposed in [15] to transform the reconstruction result into a binary image denoted as B_{RT} (see Figure 4 (d)), and mark the biggest continued foreground block by B_{lip_1}. In the case of mouth closing, B_{lip_1} can represent the whole lip region accurately. However, in most cases of mouth opening, the blocks corresponding to upper and lower lips are usually separate. It is hard to extract the whole lip region via selecting the biggest block. Thus, some refinements are needed.

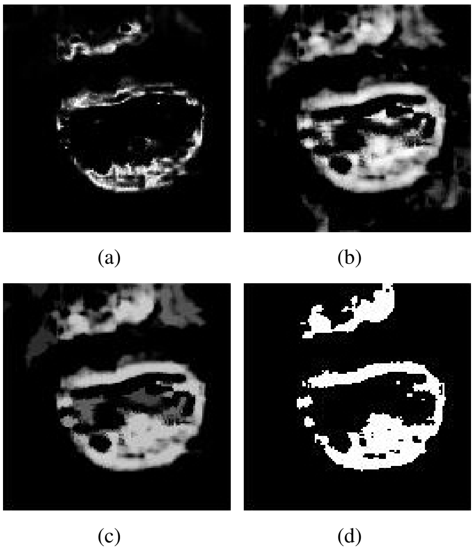


Figure 4. A sample image of (a) marker, (b) mask, (c) the result of morphological reconstruction, and (d) the result of gray-level threshold processing.

Considering the primary reason for discontinuity between upper and lower lip is that the teeth and tongue are eliminated via the above steps. Hence, we utilize the following equation

$$I_{TTM} = I_{G/R} - norm(I_{G/R} - (I_{G/R} - I_{a^*})) \tag{11}$$

to transform the image $I_{G/R}$ to I_{TTM}, where $norm(.)$ denotes the normalization in [0,255]. Subsequently, we can obtain the region covering the teeth, tongue and some parts of oral cavity approximately. Please note that, as we stated in Section 2, the computation is employed in the positive area, i.e. each negative results are set at zero, thus $I_{G/R} - (I_{G/R} - I_{a^*})$ is not equal to I_{a^*}.

We further transform I_{TTM} into a binary image denoted as B_{TTM} by the threshold selection method. Then, the morphological closing is employed to $B_{RT} \cup B_{TTM}$ by performing a 5×5 structuring element operation. We select the biggest foreground block denoted as

B_{lip_2} in the closing operation result. Hence, the binary image $B_{lip_1} \cup B_{lip_2}$ can represent the whole lip region even in the case of mouth opening. Furthermore, we can utilize the morphological opening with 3×3 structuring element so as to make the edge more smooth. The result is denoted as B_{lip}.

For the foreground pixels in B_{lip}, the corresponding positions, i.e. the row and column indices, are recorded and compose an $M \times 2$ matrix, denoted as P, where M denotes the number of foreground pixels in B_{lip}. We calculate the eigenvectors and eigenvalues of the covariance matrix of P. Subsequently, we can obtain an ellipse whose position and inclination are defined by the eigenvectors, and the length of major/minor axis are defined by the 1.5 times eigenvalues, respectively. Consequently, two horizontal lines crossing the highest and lowest point of the ellipse are obtained. The continued objects on the outside of the two lines are masked out from the stage of lip contour extraction.

Finally, the quickhull algorithm proposed in [16] is employed to draw the counter of lip (e.g. see Figure 5).

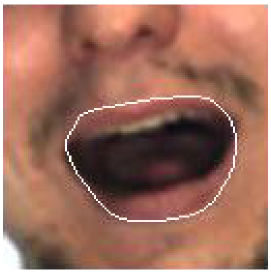


Figure 5. The final result of lip counter extraction.

4 Experimental Results

To demonstrate the performance of the proposed approach in comparison with the existing methods denoted as: Liew03 proposed in [5], and Guan08 in [7]. We utilized the four databases to test the robustness in different capture environments: (1) AR face database (126 people with 26 images for each) [17], (2) CVL face database (114 persons with 7 images for each) [18], (3) GTAV face database (44 persons with 27 images for each), (4) a database established by ourselves, including 19 persons (10 male and 9 female) with 15 pictures per person corresponding to different mouth shapes. We randomly selected 900 images in total (400 images from AR database, 200 images from CVL database, 200 images from GTAV database, 100 images from our database) and manually segmented the lip to serve as the ground truth. Moreover, in AR database, the images with the feature number 11, 12, 13, 24, 25, 26 (wearing scarf which covers the whole mouth) are not used for this experiment. Some segmentation results can be found in Figure 6.

Algorithm	Liew03	Guan08	Proposed
average OL, %	80.73	45.10	**89.27**
average SE, %	20.15	55.21	**9.32**

Table 1. The segmentation results across the four databases.

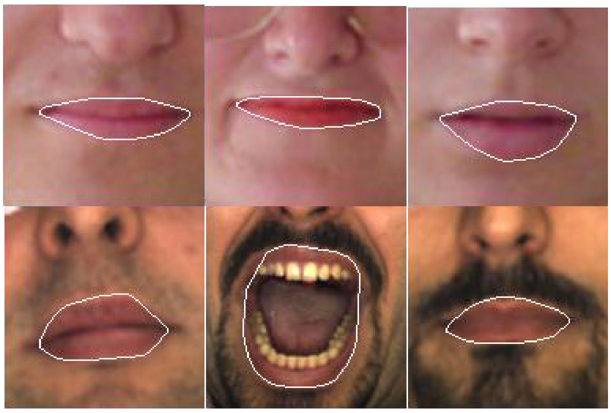


Figure 6. Some samples of lip counter extraction in different databases.

Two measures defined in [5] are used to evaluate the performance of the algorithms. The first measure determines the percentage of overlap (OL) between the segmented lip region A_1 and the ground truth A_2:

$$OL = \frac{2(A_1 \cap A_2)}{A_1 + A_2} \times 100\%. \qquad (12)$$

The second measure is the segmentation error (SE) defined as

$$SE = \frac{OLE + ILE}{2 \times TL} \times 100\%, \qquad (13)$$

where OLE is the number of non-lip pixels classified as lip pixels (i.e. outer lip error), ILE is the number of lip-pixels classified as non-lip ones (inner lip error), and TL denotes the number of lip-pixels in the ground truth.

Table 1 shows the segmentation results on the four different databases. It can be seen that the proposed method outperforms the Liew03 and Guan08 in both of the two measurements.

5 Conclusion

In this paper, we have proposed a new approach to automatic lip segmentation via the probability model in color space and morphological filter. This approach features the high stability of lip segmentation and robust performance against the different capture environment and different skin color (white and yellow). Experiments have shown the promising result of the proposed approach in comparison with the existing methods.

References

[1] I. Matthews, T.F. Cootes, and J.A. Bangham. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:198–213, 2002.

[2] W. Gao, Y. Chen, R. Wang, S. Shang, and D. Jiang. Learning and synthesizing mpeg-4 compatible 3-d face animation from video sequence. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1119–1128, 2003.

[3] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. In G. Bailly, E. Vatikiotis-Bateson, and P. Perrier, editors, *Issues in Visual and Audio-Visual Speech Processing*. MIT Press, 2004.

[4] H.E. Cetingul, Y. Yemez, E. Erzin, and A.M. Tekalp. Discriminative analysis of lip motion features for speaker identification and speechreading. *IEEE Transaction on Image Processing*, 15(10):2879–2891, 2006.

[5] Alan W.C. Liew, S.H. Leung, and W.H. Lau. Segmentation of color lip images by spatial fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 11(4):542–549, 2003.

[6] W.C. Liew S.H. Leung S.L. Wang, W.H. Lau. Robust lip region segmentation for lip images with complex background. *Pattern Recognition*, 40(12):3481–3491, 2007.

[7] Y.P. Guan. Automatic extraction of lips based on multi-scale wavelet edge detection. *Computer Vision, IET*, 2(1):23–33, 2008.

[8] J. Luettin, N.A. Thacker, and S.W. Beet. Speechreading using shape and intensity information. In *Proceedings of IEEE International Conference on Spoken Language Processing*, pages 58–61, Philadelphia, USA, 1996.

[9] B. Dalton, R. Kaucic, and A. Blake. Automatic speechreading using dynamic contours. In D.G. Stork and M.E. Hennecke, editors, *Speechreading by Humans and Machines: Models, Systems, and Applications*, Berlin, 1996.

[10] D. Chandramohan and P.L. Silsbee. A multiple deformable template approach for visual speech recognition. In *Proceedings of IEEE International Conference on Spoken Language Processing*, pages 50–53, Philadelphia, USA, 1996.

[11] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, 2(3):141–151, 2000.

[12] M. Lievin and F. Luthon. Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video. *IEEE Transactions on Image Processing*, 13(1):63–71, 2004.

[13] N. Eveno, A. Caplier, and P.Y. Coulon. A new color transformation for lips segmentation. In *Proceedings of the 4th IEEE Workshop on Multimedia Signal Processing*, pages 3–8, Cannes, France, 2000.

[14] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2(2):176–201, 1993.

[15] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[16] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[17] A.M. Martinez and R. Benavente. The ar face database. *CVC Technical Report No.24*, June 1998.

[18] F. Solina, P. Peer, B. Batagelj, S. Juvan, and J. Kovac. Color-based face detection in the '15 seconds of fame' art installation. In *Proceedings of Conference on Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, Image Analysis and Graphical Special Effects*, pages 38–47, Versailles, France, 2003.