# PROCEEDINGS

**The 12th HKBU-CSD Postgraduate Research Symposium**

# PG Day 2010

**Department of Computer Science**
**Hong Kong Baptist University**

**September 6 & September 7, 2010**

# The 12th HKBU-CSD Postgraduate Research Symposium (PG Day) Program (Updated)

| September 6th 2010, Monday | |
|---|---|
| **Time** | **Sessions** |
| 09:30-09:40 | **On-site Registration (Room OEW 802)** |
| 09:40-10:00 | ***Welcome Address* (Room OEW 802)**<br><br>**12th HKBU Postgraduate Research Symposium (PG Day) Welcome Address.**<br>***Prof. Jiming Liu,***<br><br>**Chair Professor and Head of Computer Science Department,**<br>**Hong Kong Baptist University** |
| 10:00-11:00 | ***Distinguished Lecture* (Room OEW 802)**<br>**(Chair: *Prof. Jiming Liu, Head of Department of Computer Science, HKBU*)**<br><br>**Prepare Your Computer Science and Engineering Career in a Non-linear World.**<br>***Prof. C.L. Philip Chen***<br><br>**Chair Professor and Dean, Faculty of Science and Technology**<br>**University of Macau** |
| 11:00-13:30 | **Noon Break** |
| 13:30-16:00 | **Session I (Chair: Mr. Xia Shang, T909)**<br>• ***Secure Proximity Monitoring in Mobile Geo-Social Services***<br>Mr. Li Hong Ping<br>• ***Of Acquiring a Motion Field in the Compressed Domain***<br>Mr. Cheung Quan Jia<br>• ***Detecting, Locating, and Tracking Hacker Activities within a WLAN Network***<br>Mr. Shun Chin Yiu<br>• ***Speeding up Scoring Module of Mass Spectrometry Based Protein Identification by GPUs***<br>Mr. Li You<br>• ***Automatic Segmentation of Color Lip Images Based on Morphological Filter***<br>Mr. Li Meng |
| 16:00-16:10 | **Tea Break** |
| 16:10-18:10 | **Session II (Chair: Mr. Zou Weiwen, T909)**<br>• ***Robustness and Entropy for Directed Networks without Loops***<br>Mr. Shi Benyun<br>• ***A Computational Study on the Impact of Human Traveling Behaviors on Infectious Disease Dynamics***<br>Mr. Xia Shang<br>• ***Semantic Indexing for Music Search with Adaptive Recommendation***<br>Mr. Deng Jie<br>• ***A Robust Lip Tracking Algorithm using Localized Active Contours and Deformable Models***<br>. Mr. Liu Xin |

# The 12th HKBU-CSD Postgraduate Research Symposium (PG Day) Program

| September 7th 2010, Tuesday | |
|---|---|
| **Time** | **Sessions** |
| 09:20-10:50 | **Session III (Chair: Mr. Shi Benyun, T909)**<br>• ***Learning the Relationship between High and Low Resolution Images in Kernel Space for Face Super Resolution***<br>Mr. Zou Weiwen<br>• ***Cooperative and Penalized Competitive Learning for Clustering Analysis***<br>Ms. Jia Hong<br>• ***A Linear-chain CRF-based Learning Approach For Web Opinion Mining***<br>Mr. Qi Luole |
| 10:50-11:00 | **Tea Break** |
| 11:00-12:30 | **Session IV (Chair: Mr. Li You, T909)**<br>• ***Incremental Maintenance of Minimal Bisimulation of Cyclic Graphs***<br>Mr. Deng Jintian<br>• ***Selectivity Estimation of Twig Queries on Cyclic Graphs***<br>Mr. Peng Yun<br>• ***A Tag Means a Lot Than It is in a Folksonomic System***<br>Mr. Tsoi Ho Keung |
| 12:30-14:00 | **Noon Break** |
| 14:00-15:00 | **Session V (Chair: Mr. Xia Shang, T909)**<br>• ***Delay Cascades in a Queueing Network of Cardiovascular Care***<br>Ms. Tao Li<br>• ***A Rotation-invariant Script Identification based on BEMD and LBP***<br>Mr. Pan Jianjia |
| 17:00-17:30 | **Best Paper & Best Presentation Awards Announcement (Room T909)** |
| **Closing** | |

# Table of Papers

# Secure Proximity Monitoring in Mobile Geo-Social Services

Li Hong Ping
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
hpli@comp.hkbu.edu.hk

## Abstract

*Nowadays, Location Based Services (LBS) become more and more popular. According to the influence of the social network, users usually join social groups with their friend. In some group, user may want to be notified, if the user is geographically close to any users within the social group. Proximity Detection enable us to attain the goal. However, we are required to bear the risk of disclosure of location information, while we are enjoying the LBS. This privacy threat reduces the attractiveness of this kind of services. Previously, some papers contribute ideas to deal with this problem. Nevertheless, most of them assume that there is a trusted central server within their system, which is impractical in the real world. The problem under our study is to continuously monitor if any two mobile users in a social group are within a distance of D. Meanwhile, the exact location of a mobile user is not disclosed to any third party. This paper propose a computationally feasible solution in this problem, which only disclose the approximate location of the user to the authorized party. Untrusted third party (including centralized server) is not able to know the users' location.*

## 1 INTRODUCTION

As the technology advance, more and more mobile phones and PDAs are equipped with geo-positioning capabilities (e.g. GPS). At the same time, the rise of social networking sites to narrow the gap among people. Friend-locator services (e.g., Google Latitude), which enable user to know their friends' locations, is also becoming popular. Nevertheless, friend-locator services users usually expect certain level of privacy protection rather than completely expose their position to their friends.

The existing research work in proximity detection can only protect users' location privacy in a certain degree. They are not enough to satisfy the requirement of requesting completely location privacy.

In this paper, we can completely protect the users' location privacy, because we do not require any trusted third party. Under the protection of secure communication, server can just receive the hash value. By using those value to judge where there is any user nearby each other. Server only announces the users, when they are locating within a pre-defined distance of another user. In general, users have different location privacy requirement to other users according to different social group. Such as user Alice allow her family member to know her location when they are in the same district and allow her friend to know her location only when they are within (e.g. fifty meters) from each other.

The design of proposed solution gives three contributions at the same time. Firstly, it can preserve users' location privacy even there is no trusted third party. It outperforms many of the previous solution, which requires the existence of a centralized anonymity server. Secondly, using hash function with a set of time-varied salt gives us a better protection, because it is pretty hard for us to find the user location from the hashed value. We prevent unwanted party to know where we are and let the permitted party to know our approximate location. Thirdly, this solution employed grid base layered structure, which is similar to some of the previous approach. The major different between previous solution and our solution, is previous solution always quad-tree structured grid based layer and our solution use a nona-tree structured grid based layer (see Figure 1). This modification reduces the number of required layer.

Our solution not only gives a good protection in user location privacy, but also requires a low communication cost, which is directly proportional to the number of user. The paper is organized as follows. We briefly review related work in Section 2 and then give a problem definition in Section 3. System Operation is presented in Section 4. Section 5 presents the experimental results of our proposed solution. At last, we conclude our paper in Section 6.

## 2 RELATED WORK

In this section, we review the development of the location privacy technology and show the contribution of this paper to this topic. In the literature, there are many research efforts in this topic. Most of them adopt one of the three techniques (including cloaking, dummies and encryption) when handling the user location privacy problem.

The earliest proposal for location privacy protection is spatial cloaking, which is proposed by Gruteser and Grunwald [1].

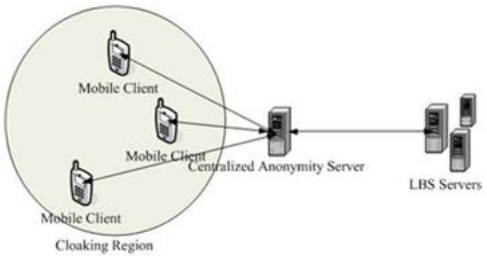



**Figure 1. General Structure of Location Cloaking**

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, see Figure 1. However the quality of service is highly depend on the density of users' distribution. Also, it is time consuming for searching nearby users to form a cloaking region.

Later, location cloaking algorithms advanced from cloaking of snapshot locations to continuous location updates [3, 4]. 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). Except the most common k-anonymity cloaking, there are other types of cloaking method, such as Hilbert curve [5] and Casper[6]. Both of Hilbert curve [5] and Casper[6] employ grid-based cell as their cloaking region, which also the idea that this paper has employed. The major advantage of grid-based cell is it requires less time for us to locate ourselves, comparing with the k-anonymity cloaking which require location of k nearest neighbor to find out our cloaking region. Also, grid-based cell can have a better resistant to path-tracking, as the locations of all cells have been predefined by the system. A. Khoshgozaran and C. Shahabi [5] guarantees the query anonymity even location information is disclosed to the adversary.

However, each client needs to maintain complex data structure and communication protocol as well as long range communication among peers. Therefore additional computation and communication cost may be quite costly for practical use. For the Casper [6] solution, it blurs a user's exact location information into a grid-based cloaking spatial region based on user specified privacy requirements. This framework uses a quad-tree data structure that maps the location information into grids with different levels and resolutions. Due to the limitations of the quad-tree structure, the calculated cloaking region is often larger than required, which may cause lower service quality.

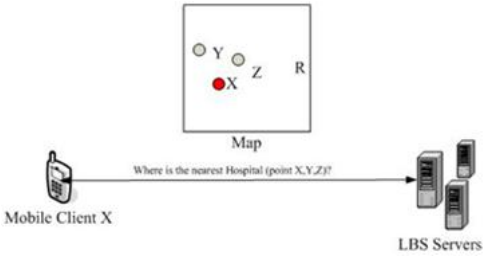



**Figure 2. Protect location privacy by using faked dummies**

On the other hand, [2] suggest that we can protect our location information by faked dummies. Just like Fig. 2 show that when client X send his location service request to the LBS server, he will also send out the k faked dummies (Y,Z,....) simultaneously, so as to diversify the risk of the discovery of his actual location. Although [2] has tried to user some movement simulation technique, we cannot prevent the threat of path tracking, because the exact location must contain in the set of dummies. Therefore, it is not difficult for us to find out the user location by using the assistance of the data mining technique.

The previous solutions only provide some protection for the location privacy. However, most of them require a trusted third party (e.g. trust anonymity server) to process the users' location data. It is not practical for us to request for a trusted third party in the real world.

Cryptography solution in location privacy can help us to protect the location privacy, while without the existence of the trusted third party. Yao's [7, 8] present how to exchange secret by using some comparison method. More recently, G. Ghinita and P. Kalnis [9] use

Private Information Retrieval (PIR) implementation to build up a location privacy protection framework which does not require any anonymizers 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 required storing large amount of different content. Also, it is not easy to find the optimal size of grid partition that makes the computation and communication cost becomes huge.

Many papers have already been published for the topic of finding k-nearest neighbor (kNN).

Except the research of finding kNN, proximity detection is another important topic in location privacy application. The defination of proximity detection is the capability of a location-based service (LBS) to automatically detect when a pair of targets approaches each other closer than a predefined proximity distance. It is not efficient for us to do the proximity detection, if we solely use the solution of kNN. That may give us too (less/much) information when the point of interest (POI) are unevenly distributed. Ruppel [10] applies a distance-preserving coordinate transformation. By using centralized proximity detection method to detect the proximity among the transformed locations. However, Liu et al. Liu [11] show that distance preserving coordinate transformations is not safe enough, as it is easy for attacker to derive the secret mapping function. Mascetti present a solution - Hide&Crypt, presented in [12], is a privacy preserving solution which employs a filter-and-refine paradigm. Server uses the specified thresholds and computed distances to determine whether friends are in proximity. However users may need to directly communicate with their friend to check their proximity status, if they are defined as "possibly in proximity". Hide&Crypt use secure multi-party computation (SMC) protocol, which can protect the users' location privacy, but it also brings a choice between the service quality and the communication cost.

More recently, FRIENDLOCATOR [13] and VICINITYLOCATOR [14] both track users in a sparse grid while they are far away from their friends, in order to reduce the communication cost. Changing to finer grid, only when they come closer with their friend.

The limitation of the FRIENDLOCATOR is the proximity detection accuracy of it is low and uncontrollable. For VICINITYLOCATOR [14] give a solution that allow user to choose the "area of interest". In VICINITYLOCATOR, client requires to find all granules contained in his vicinity. We still have to make a tradeoff between the service quality and performance. It is because if the granule size is big, the function of the granules will become meaningless and it also scarifies the accuracy of the result. On the other hand, if the granule size is small, the computation and computation will be high. Also, VICINITYLOCATOR require disclosing the encrypted location to server, because encryption is revisable. So it still not a complete safe solution.

Similar to FRIENDLOCATOR [13] and VICINITYLOCATOR [14], our solution employs an adaptive position-update policy, which use different density layer, in order to reduce the communication cost. Comparing with traditional grid layer structure, our solution use non-tree structure rather than quad-tree structure. Therefore our solution requires less number of layers to solve the problem in some resolution. Also, our solution use hash function instead of encryption. Hash function is irreversible because there are many different numbers map to the same value. At the same time, because the weak collision probability of hash function, it is rare to find a pair of value, which has some hash value. Furthermore, every time before our location information undergo the hash function, it will combine will a random generated salt. Users can refresh the random salt as they wish. As a result, even users stay in the same place. The message they send to the server is always different. Moreover, comparing with proximity detection, proximity monitoring is a continuous work, we are able to show whenever two users are nearby once the system gets started.

## 3 PROBLEM DEFINATION

This section describes the system model under our study. We assume users within the social group have reached a consensus on the acceptable distance of proximity detection. Figure 3. show us a sample of user have reached a consensus with all users in different social group. This gives system users flexibility to adjust the proximity distance, according to their need. In our system, there are 2 types of

| Social Group | Proximity Distance (m) |
|---|---|
| Colleague | 20 |
| Friend | 50 |
| Family | 100 |

**Figure 3. Table of acceptable distance of proximity detection for different social group**

entities, client and server. Client is the users within the people within the same social group and server is just used for comparison of hashed value. Figure 4 display our system framework. The key idea to preserve the location privacy is separating the information sharing into 2 parts. Traditional centralized anonymity server act as a system center, which responsible for all client requests and analysis work. Therefore, those solutions require making an assumption, that the centralized server must be trusted. It is impractical to expect centralized server to be trusted, because the users' information is valuable. Even if centralized server is non-colluded, nobody can guarantee it will not be broken in by the attacker. Inspired by the solution of Yao's millionaire problem [7,8,15,16], our solution require users to share some standard with the other users directly and the server duty is to analyze the secret value.

Figure 4 show us the framework of our system. Firstly, one of the users in that social group shares a data processing

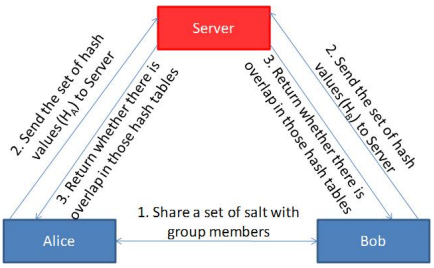



**Figure 4. System Framework**

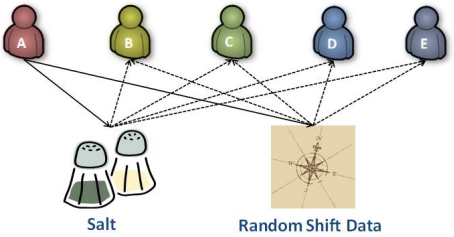



**Figure 6. System Initialization**

standard (e.g. the shifted location of different table, proximity distance) with other group member. Then, all users will transform their location information to a secret value, according to the given standard. After the transformation finish, all users will send their hashed location to the server. Finally, server will check whether all user hashed location to find whether they are within the proximity distance of another user.

# 4 MECHANISM OF SYSTEM WORK

In this section, we give the detail explanation of our system's operation. Probably, our system workflow can be divided into three stages. The first stage is initialization, which has mentioned in the previous section (Figure 5 step 1), group member establish a communication standard among them in order to achieve the aim of secure multi-party computation (SMC). The following stage is the system operation, in this stage users require to send the hash value to the server. Then server is able to determine whether there is a positive result in detection. The previous stages have already helped us to finish the proximity detection. However, there is a room of improvement for us to minimize the communication cost. In order to reduce the communication cost, we employed non-tree structured grid based layer (see Figure 11). It helps us the filter out the higher probability candidate, eliminate all the unnecessary updates.

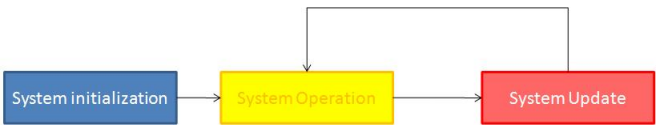



**Figure 5. System Workflow**

## 4.1 System Initialization

In the initialization stage (Figure 6), one of the users in the social group generate the random salt and shifted data and share them to other users within the group. The usage of the salt is to make the hashed location become unpredictable to the third party.

### 4.1.1 Use of Salt

If there is no random salt and system only hashes their location directly, it may be suffer from the brute-for attack. On the contrary, if we combine the location with a randomly generated salt, the hashed location becomes unpredictable.

### 4.1.2 Use of Random Shifted Data

Our solution use grid based layer (cell size $D^2$) as a foundation, where D is a proximity distance of the social group. Every user locates in different grid cells of the layer. After that we can have proximity detection by checking whether there are users in the same grid cell. If they are in the same grid cell, we can sure their distance must be within the range of D.

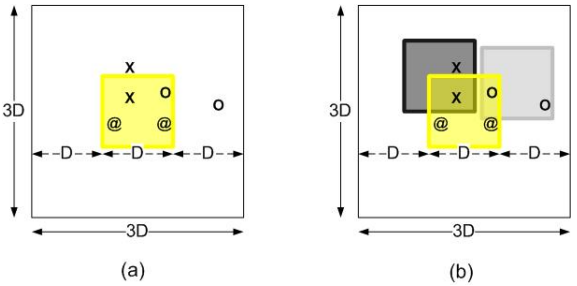



**Figure 7. Single Grid Layer vs Multi-Grid Layer**

However, if we only use one table to check the distance, we miss too many positive results even they are within a distance D. In Figure 7a, "O", "X", "@" pairs are within distance D with each other. However only "@" is allocated in the same cell, other pairs locate near the edge of the cell. Therefore even their distance between each other is less than D; they are still treated as unqualified candidate in proximity monitoring. The solution of this problem is add more same size grid based layer and shift randomly to any direction within distance D. The randomly shifted data

is only a set of simple random numbers, which range is between -(cell length) to (cell length). That mean if the grid layer is in Layer 0, the range of randomly shifted data is between -D to D. For Layer 1, the range is -3D to 3D, etc. Figure 7b show that after adding more layers "O" pair fall in the gray color grid cell and "X" pair fall in the black color grid cell. In order to simply the demonstration, after we add gray color layer and black color layer we find that "O" and "X" are also within distance D from each other. In fact if we just solely add a few layers is not enough for us to stable our service quality.

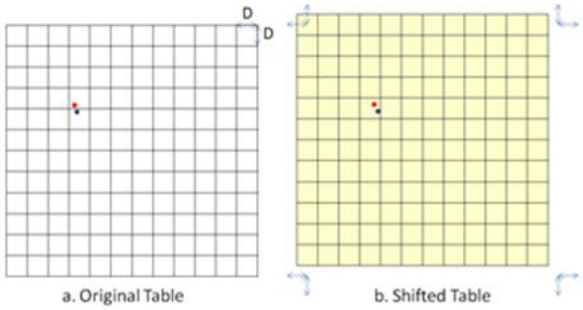



**Figure 8. Original Table and Shifted Table**

Figure 8. show us how randomly shifted cell can help us to solve the problem of missing case. As we can see even there are 2 dots (represent 2 users) in the original table, which are very close to each other. They have not fallen in the same cell.After add a shifted table, which shift to the left up direction. Both users fall in the same cell again and they will be treated as distance D beside his friend.

$$P = (0.75)^N \tag{1}$$

| No. of Mappings(N) | Prob. of missing report(P) |
|---|---|
| 1 | 75% |
| 2 | 56.25% |
| 3 | 42.19% |
| 4 | 31.64% |
| 5 | 23.73% |
| 10 | 5.63% |
| 15 | 1.34% |
| 20 | 0.32% |

**Table 1. Relationship between the probability of missing report and number of layers**

Table 1 shows us the relationship between the probability of missing report and number of layers. For a user who require for distance D proximity monitoring, its coverage area of its D distance will be $(2D)^2$. For one cell size $(D)^2$ only cover 25% of the coverage area. That's mean there have 75% chance of missing report. However, the missing rate can be reduced by adding more randomly shifted layer. As we can see when there are 20 layers, the rate of missing is only 0.3% which is only a very small chance.

As a result, we see the usage of the randomly shifted data, which solve the problem of missing report by using only one grid cell.

## 4.2 System Operation

After the standard is shared, the next step is using those data to achieve our major purpose - secure proximity monitoring. For the ease of illustration, we use only one table for the explanation, which is one of the tables from the bottom grid layer.

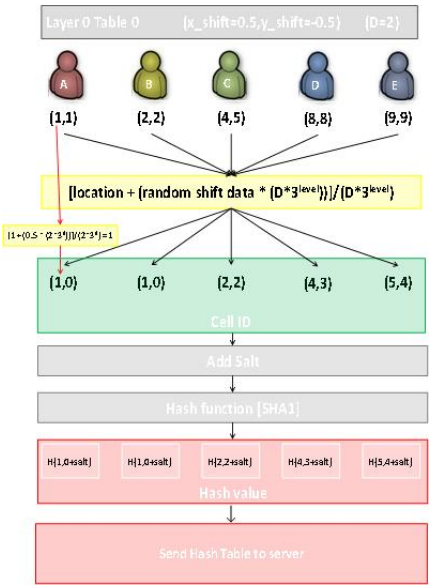



**Figure 9. Workflow of System Operation**

In Figure 9, we can see there is 5 users A,B,C,D,E. After A share the salt and random shifted data (x = 0.5, y = -0.5) in the initialization stage (Figure 7), other users are able to use those information to find cell ID and hide their location under the same standard. As the cell size is (D*D), if two users fall in the same cell, they must be the positive candidate in the proximity monitoring. Then, all user find the cell ID they belong to. The following step users are going to combine their cell ID and the shared salt and undergo the hash function (e.g. SHA1). Finally, all users send their hash value to the server. Then, server checks whether there is same hash value among users. In this case, server find user A,B and user D,E are in the same cell. So server announce user A, B they are nearby each other and hide the location of other users, because they are outside the bound

of proximity monitoring. User D,E will also have the same arrangement as user A,B.

### 4.3 System Update (Minimize Communication Cost)

It is sure that we finish our work by using one layer of hash values. Nevertheless, we need to check all users hash value frequently, which consume much more resource than we actually require.

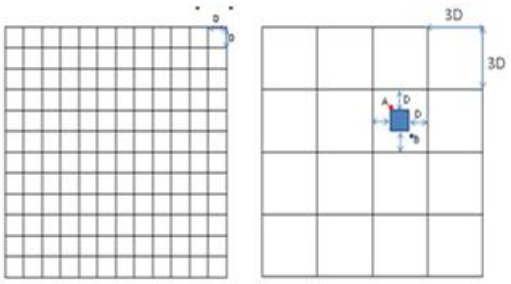



**Figure 10. Sample of $D^2$ Layer**

In the Figure 10, we show an example to explain the reason of frequent update, when only one layer has been used. In this figure, it shows us the distance between user A and B is 1.1 D. Although they are quite nearby each other, it is impossible for them to fall into the same grid cell. Therefore they are treated as away from each other. In this case, the problem come if A and B do not update frequently. It is because when there is no immediate reaction when user A and B come within distance, that show the service quality of system is not good enough. On the other hand, other users also require updating frequently, even they are far away from each other. We need some efficient update approach to improve quality of service and reduce communication cost.

There are some solutions [6, 11] suggest to create the layer based map, so as to reduce the communication by including the sparse layer. However, there are some differences between our solution and existing solution.

In existing solutions they use quad-tree structure fixed location lay, our solution use non-tree structured randomly-shifted layers. Attackers are more difficult to find the exact location of the users. Also, our solution hash the location into hash value, rather than directly transfer the grid based location to server.

Figure 11 shows the vertical view of our vertical hash structure. Our system is setup in a bottom-up manner, which starts at the bottom layer (cell size $D^2$) and end until the whole map is covered by a single cell.

$Max(L, W) = 3^n D$

That means the number of layer (n) in the system depends on the proximity distance (D) and the lager one of length (L) or width (W) of the map.

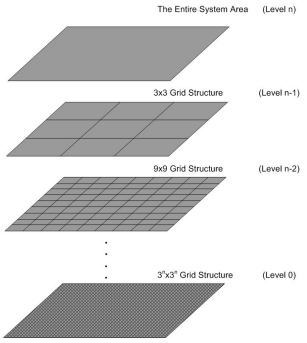



**Figure 11. The Non-Tree Structured Grid Based Layer**

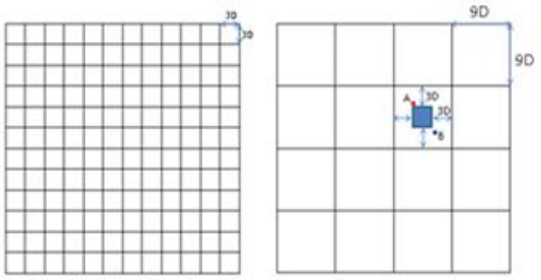



**Figure 12. Sample of $(3D)^2$ Layer**

We use Figure 10 and 12 as an explanation model of our work. In Figure 10, the distance between user A and B is 1.1 D, no matter how we shift the size D2 hashed table. They 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. The similar concept can also be used in Figure 12. , the distance between user A and B is 3.1 D, no matter how we shift the size D2 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 3D. The concept of expand layer of Figure 10 and 12 can be expanded. As the cell width and length expand a constant times per layer, so the size is also expand 9 times per layer. According to speed of expansion, a single cell in upper layer can cover a large place. That mean player require longer distance to escape the cell, thus less update is needed.

### 4.4 System Update in 2 Users Situation

In order to facilitate explanation, we first illustrate the system update of 2 users. After that, we will talk about the system update in practical multiuser situation. Our system do things in the initialization stage, so as to reduce the communication cost in the later stage. Therefore, in the initial-

ization stage server collect all the hashed value from users and users require to update only if necessary. In the following subsection, we will explain how our system work in different situation when there is 2 users.

#### 4.4.1 2 Users Nearby Initially

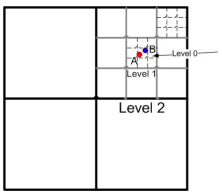



**Figure 13. 2 users nearby each other**

In this case, user A and B distance are less than D. That's mean they are overlap at the Level 0. So if they are required to do update when they leave any of the mappings in level 0.

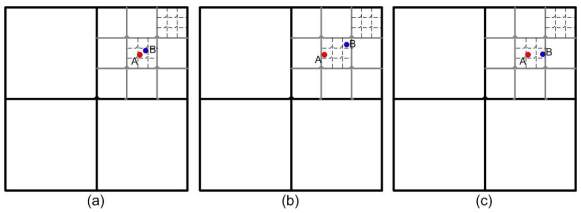



**Figure 14. Different situation of 2 users in Time 1**

Figure 14 shows us 3 different situation user A and B will face in later stage. Case (a) is user A and B remain same mapping as before. We do not need to do anything because nothing is changed. For the case (b), A and B move away from each other when there is change in the mappings. Update is required to be done, as the change of mapping for both users is only occurring in Level 0. Therefore both user just need to announce the server, there is change in certain layer. Then, both users need to update all layers at or below the certain layer. The solution of case C is similar to case B. The major difference is only user A requires to send the update information to the server. As user A quit the previous mapping while user B is remaining unchange.

#### 4.4.2 2 Users Away Form Each Other Initially

In this case, the distance between user A and B is 7.5 D. That's mean they only overlap in the higher level - level 2. Both users do not require to update their mappings if there is no change in Level 2. That's mean even there is change in level 1 and level 0 for both users, if the mapping in level

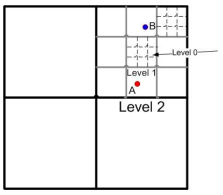



**Figure 15. 2 users away from each other**

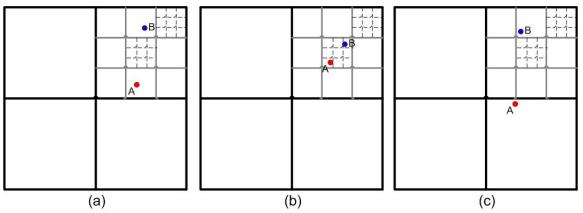



**Figure 16. Different situation of 2 users in Time 1**

2 is unchanged, no update is needed. So in case (a), both user A and B do not require to do any update. In case (b), both users quit their mappings in level 2, but not quit their mapping in level 3. Therefore we only need to update the mappings at or below level 2. After both users update their mappings to server, server find that both users overlap in a closer level - level 1. Then the server will focus on the update of level 1 of both users just like what is done in level 2 previously. On the contrary, case (c), both users quit their mappings in level 2 and no longer overlap in level 2. Server find their minimum overlap level is level 3. After that server will focus on the update of level 3, if there is no change in mapping of level 3, no update is required.

### 4.5 System Update in Multiuser Situation

All users have undergone the update process since they start the location monitoring. Multiuser solution is similar to the 2 users version. They are both share all sets of mapping during the system initialization. The major difference of two solutions is in the 2 users version update depend on the minimum overlap level (MOL) of 2 users, while multiuser version focus on a few near neighbor which overlap in lower level.

| User\User | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | 0 | 3 | 5 | 6 |
| 2 | 0 | | 4 | 2 | 2 |
| 3 | 3 | 4 | | 5 | 3 |
| 4 | 5 | 2 | 5 | | 4 |
| 5 | 6 | 2 | 3 | 4 | |

**Figure 17. User Overlap Matrix**

In the multiuser solution, we construct a user overlap matrix in the server, in order to keep track with the change in mappings. Figure 14 show us a user overlap matrix, if the value in the matrix is 0 that mean that two users have some hash table overlap at level 0 (cell size = $D^2$), then they must be their distance must within D. The level higher, the distance longer. Minimum overlap level (MOL) is the overlap level of the nearest neighbor. We use MOL as the update check because the lower level is the more likely to change and also more important in proximity monitoring.

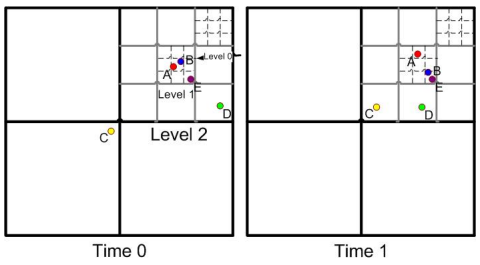



**Figure 18. Users Location from Time 0 to 1**

| User/Use | A | B | C | D | E |
|---|---|---|---|---|---|
| A | | 0 | 3 | 2 | 1 |
| B | 0 | | 3 | 2 | 1 |
| C | 3 | 3 | | 3 | 3 |
| D | 2 | 2 | 3 | | 2 |
| E | 1 | 1 | 3 | 2 | |

→

| User/Use | A | B | C | D | E |
|---|---|---|---|---|---|
| A | | 1 | 2 | 2 | 1 |
| B | 1 | | 2 | 2 | 0 |
| C | 2 | 2 | | 2 | 2 |
| D | 2 | 2 | 2 | | 2 |
| E | 1 | 0 | 2 | 2 | |

**Figure 19. Matrix Change from Time 0 to 1**

Figure 18 show us the location of user A,B,C,D,E from time 0 to time 1 and figure 19 is the overlap matrix of 5 users, which construct in time 0. Each time user may need to send an update to server, then server use this matrix to determine whether the user is required to update. Therefore even in the worst case only 3N communications is required, where N is the number of users.

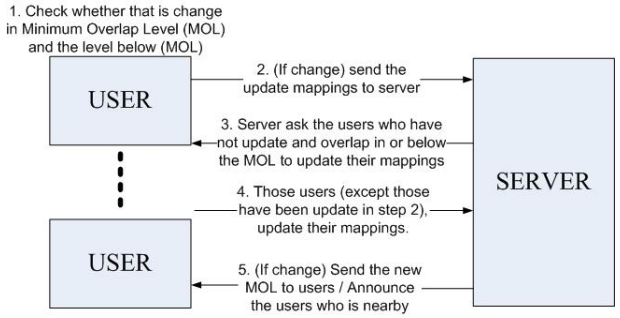



**Figure 20. Update Process in Every Period**

In figure 20, we have a more detail explanation in our solution and use the users from figure 18 as our sample. All users require to do step 1, check whether there is change in or below the MOL. Check the change of one level below MOL can help us to detect the other users in the MOL come closer. In our system update process, there are 3 cases with different communication cost will occur in the process.

Firstly, for the worst case which require 3N communications occur when overlap with new users in or below the MOL (step 1,3,4,5), just like the case of user E. Even he does not exit his mappings in MOL, but user B overlap with user E in level 0, which is a level lower then the original MOL. Therefore user E also requires to update.

The second case is the user leaves its mapping in or a level below the MOL (step 1, 2,5) just like user A,B,C,D. They exit the mapping of their MOL. Therefore they are all required to update their new mappings to the server. Under the consideration of cost saving, users only transfer the changed level mappings rather than update all mappings blindly.

The third case is the best case which do not require any communication occur when user do not exit any mappings in or one level below the MOL and also no new user entry or old user exit the MOL. Therefore, after (step 1) is finished, it will directly jump to (step 5) and see whether there is change in MOL.

The cost of update of step 2 and 4 is depending on number of levels of mapping change. For example, if there is change in mapping of level 0,1,2 between the update period. Then 3 levels are required to update.

## 5 EXPERIMENT

Our experiment employ T. Brinkho[17] solution as the base of sample collection. We use the paper[17] provided city Oldenburg as our test case. In order to have a better understand of our system performance, we analyze it in three aspect number of users, proximity distance (D) and effect of speed. For the size of social group, we use (10, 20, 40, 80) as our test case. It is a reasonable size of a social group, while we can also observe the relationship between the number of users and the cost clearly. For the proximity distance (D), according to M.Gruteser and D. Grunwald [1] suggestion, they propose 100m is a suitable segment distance for Driving Conditions Monitoring. So we use (25m, 50m, 100m, 200m) as our test cases and handle the case of both walking and driving monitoring. Not only concern on different need of groups, but also the difference of performance in various speed.

The graphs in figure 21 show us, how the performance of using the multi-layer grid is based layer approach instead of the single layer approach. For the single layer version, we require to update frequently, because if one the hashed value change, we need to do an immediate update, so as to preserve the data accuracy. Just like the situation we have mentioned in Section 4.3. It is different from the multi-layer version, which separate users into sparse level. From the

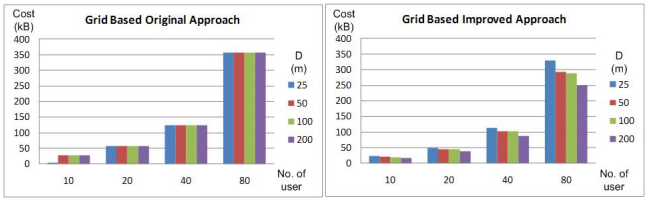



**Figure 21. Comparison between Single-Layer and Multi-Layer Grid Based Approach in Average Communication Cost of Server in a Second**

above graph, it shows us the performance is similar if the proximity distance (D) is small. However, proximity distance (D) lager, more cost can be save in the multi-layer approach. Normally, if proximity distance (D) is larger, users have higher chance to overlap. Thus communication cost is also higher. However, in our system the relationship between proximity distance (D) and communication cost is reversed. For D = 25, we use 7 layers to cover the whole map, but when D = 200, we only require 5 layers to do this. Multi-user approach also performs better in more users. So multi-user approach is more suitable in practical application.

In order to have a better understanding of our system performance, we make a comparison between our system and a native cloaking method.

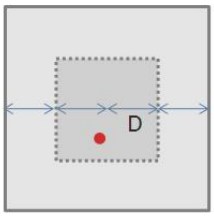



**Figure 22. Cloaking Region of the Native Approach**

Figure 22 show the structure of the native approach. First, we randomly generate a size $(2D)^2$ rectangle which contain the user's current location. Based on this rectangle, we will extend the size of the cloaking region to a size $(4D)^2$ rectangle, according to the center of the original $2D^2$ rectangle. When the system starts, all users send their cloaking region to the server. Then, server checks whether they is overlap in users' cloaking region. For the users who are nearby, we will continue ask the user to generate a new cloaking region everytime of update, until they are no longer nearby each other. For the users who are not nearby, server ask the user to update cloaking region, only if they have exit the original $(2D)^2$ area, otherwise no update is required.

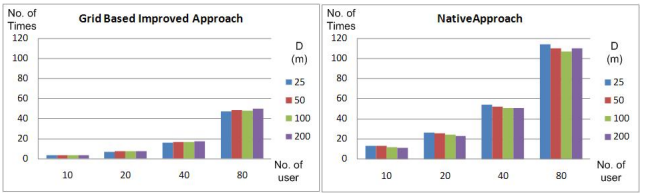



**Figure 23. Comparison in Times of Location Update**

Graphs in Figure 23 show number of times of location update required within a minutes for different number of users. Our system require less update than the native approach, because of the contribution of the higher level sparse layer. Users which have high minimum overlap level are far from other users. Therefore they need less update, because they require longer time to move out a grid based cell comparing with the lower level cases.

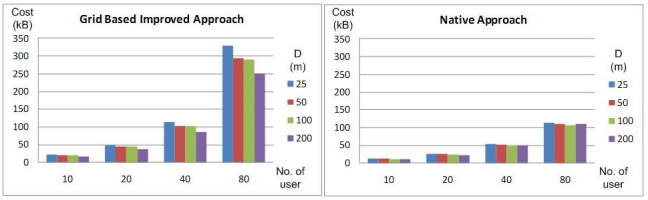



**Figure 24. Comparison between Grid Based and Native in Average Communication Cost of Server in a Second**

Despite less update is required. The communication cost of grid-based approach is still higher. It is because native approach only need to send the center of its cloaking region, while our system require sets of hash value for secure proximity monitoring. One time communication cost of native approach is around 1000 Bytes, including the communication overhead, but grid-based approach much more. Therefore even our system requires much less update, the cost of our system is still much higher than the native approach. However, the cost can be reduced by adjusting the number of layers and numbers of hash values per layer, but it is clearly a tradeoff between service quality and the load of communication cost.

The following graph show us the communication cost in one minute. Communication cost of the native approach increase when the speed is increasing and grid-based approach do not have direct relationship between communication cost and speed. Even the users are moving very fast. In grid-based approach, they require longer time for users to leave a large grid cell. On the contrary, for the native ap-

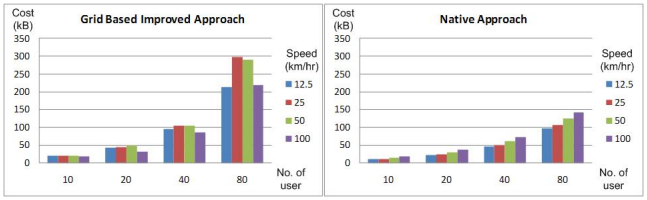



**Figure 25. Comparison between Our Approach and Native in Different Speed**

proach as the speed of the users increase, they always leave the cloaking region and overlap with new user. Therefore the communication cost of native approach will continue to increase and stop only when it reaches the equilibrium (every user required to update every time).

Although our approach requires more communication cost, it outperforms the native approach in better accuracy.

| GRID BASED APPROACH | | | |
|---|---|---|---|
| | Distance within D | | |
| | | T | F |
| System | T | 232904 | 0 |
| Display | F | 336 | 7373512 |

| NATIVE APPROACH | | | |
|---|---|---|---|
| | Distance within D | | |
| | | T | F |
| System | T | 231724 | 12548 |
| Display | F | 1516 | 7360964 |

**Figure 26. Comparison between the Report Accuracy**

Our experiment do more than 7 million comparisons in case of different number of users, speed and proximity distance (D). We come out the result, which show in Figure 26. Grid-based approach is more accurate than Native approach. Grid-based approach never gives wrong signal when user is not within the proximity distance D and gives less missing signal when there is user nearby.

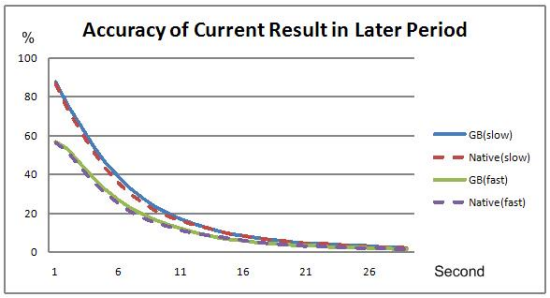



**Figure 27. Accuracy of Current Result in Later Period**

Figure 27 show us the degradation of accuracy for different approach in different speed by using current result in later period. In this experiment each period time is set as 1 second. We see that the performance of both approach is similar and grid based approach perform a bit better. The fast speed is set at 100km/hr slow speed is set at 12.5km/h. The result shows us fast speed have lower accuracy, especially in the easier stage.

## 6 CONCLUSION

In this paper, we investigate the problem of secure proximity monitoring. We propose a solution, the worst case of the solution is O( $n^2$ ). Our solution can completely protect the users' location privacy with a reasonable cost.

For the future work, for communication cost we think there is still some room of improvement. For example developing some more effective location transformation. Then communication cost can be reduced.

## References

[1] 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

[2] Hidetoshi Kido, Yutaka Yanagisawa, Tetsuji Satoh, An Anonymous Communication Technique using Dummies for Location-based Services, Pervasive Services, 2005. ICPS '05. Proceedings. International Conference

[3] T. Xu and Y. Cai. Location Anonymity in Continuous Location-based Services. In ACM GIS'07, pages 300–307, November 2007.

[4] 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

[5] A. Khoshgozaran and C. Shahabi. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In Proc. SSTD, 2007.

[6] 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.

[7] A.C. Yao. Protocols for secure computations. In Proceedings of the 23nd Annual IEEE Symposium on Foundations of Computer Science, 1982.

[8] A.C. Yao How to generate and exchange secrets. In Proceedings 27th IEEE Symposium on Foundations of Computer Science.

[9] 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.

[10] P. Ruppel, G. Treu, A. Kpper, and C. Linnhoff-Popien, "Anonymous User Tracking for Location-Based Community Services," in LoCA, 2006, pp. 116-133.

[11] K. Liu, C. Giannella, and H. Kargupta, An Attackers View of Distance Preserving Maps for Privacy Preserving Data Mining, in PKDD, 2006, pp. 297308.

[12] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, Privacy-aware proximity based services, in MDM, 2009, pp. 3140.

[13] L. Łiknys, J. R. Thomsen, S. Łaltenis, M. L. Yiu, and O. Andersen, A Location Privacy Aware Friend Locator, in SSTD, 2009, pp. 405410.

[14] L. Łiknys, J. R. Thomsen, S. Łaltenis, M. L. Yiu, Private and Flexible Proximity Detection in Mobile Social Networks, Proceedings of the 11th International Conference on Mobile Data Management (MDM), Kansas City, Missouri, May 2010.

[15] Artak Amirbekyan and Vladimir Estivill-Castro. Privacy-preserving k-nn for small and large data sets. In Proceedings of the ICDM Workshops, 2007.

[16] Processing Private Queries over Private and Indexed Data

[17] T. Brinkho. A Framework for Generating Network-Based Moving Objects. GeoIn- formatica, 6(2):153C180, 2002.

# On Acquiring a Motion Field in the Compressed Domain

Cheng Quan Jia

## Abstract

*Video object segmentation in the Compressed Domain has gained research attention due to its reduced complexity. Without the need to decode a compressed bitstream to the pixel domain, this approach makes application in real-time feasible. However, the motion vectors obtained in the motion compensation step is not intended to capture real object motion, and a measure is needed to ensure the validity of the said vectors. Since the motion compensation step in the encoding process resembles that of obtaining the 2-D optical flow field, optical flow techniques could be applied to the Compressed Domain. This paper is an attempt to draw a connection between optical flow and the information available in the Compressed Domain.*

## 1 Introduction

Image Change Detection is in essence a classification problem - to determine whether a change had occurred between images [11]. Video object segmentation is a subset of the problem - to distinguish different moving objects from the static background. For this end, motion information for each of the moving video objects must acquired before segmentation is performed. 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.

However, the processing and storage overhead in decoding every frame from an encoded video bitstream prevents these methods from application in real-time applications. Video Object Segmentation in the Compressed Domain has gained interest because of its reduced computational and storage complexity compared to Pixel domain algorithms, making them applicable to real-time applications such as surveillance systems.

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. The products of the two processes an array of DCT coefficients and predicted motion vector(s) associated with each macroblock respectively. The DCT coefficients denoting the value of vertical and horizontal frequencies and the motion vectors the approximation of image motion, both of which could be easily obtained from a parsed bitstream [10].

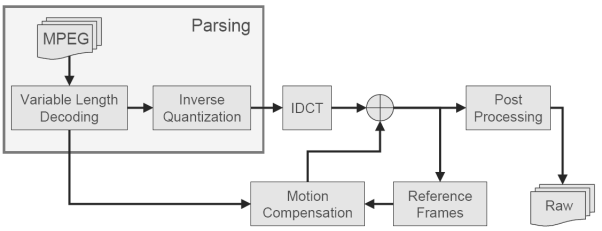



**Figure 1. Parsing from MEPG bitstream[10]**

For the following discussions, relatively small motion should exist in the input video bitstream, otherwise most of the macroblocks would be intracoded instead of intercoded because the encoders' inability to find an acceptable match in the search window.

## 2 Motion Accumulation and Median Filtering

The difficulty with the use of predicted motion vectors in video object segmentation is that the motion vectors are obtained to be the best-match of the reference frame rather than video object motion, and therefore not representative. A common approach to remove the outlying motion vectors is to accumulate the motion vectors from the bitstream over a few video frames then apply median filtering, used in [2],

[8], and [4]. However, the accumulation-and-filtering approach has two problems.

The first problem, addressed by Chen and Bajic [3], is that for repetitive motion (such as the bouncing motion of a ball) the motion vectors cancel out each other causing the accumulated motion to be of small magnitude and possibly undetected. This also leads to the discussion of the appropriate frame interval for accumulation. In [5], the accumulation interval is reduced to one (i.e. only using the motion vectors in the succeeding frame) if the average motion is estimated to be higher than a threshold. The second problem is that, while median filtering applied after motion accumulation would give a smooth motion field, we do not know if the accumulated motion field is contaminated by inaccurate motion vectors in the first place.

Porikli et al.'s investigation further suggests the of ineffectiveness of using motion accumulation alone for video segmentation. Porikli et al. [10] experimented with almost all of the information present in the Compressed Domain. The experimental results show that a slight over segmentation using DCT coefficients followed by aggregated motion based clustering produces more accurate boundaries than single stage joint segmentation. 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. Ironically, 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 algorithm favours moderate motion since spatial-temporal volumes would be disjoint in the presence of motion larger than the area of segmented 2-D object.

As the accumulation-and-filtering approach is ineffective as it is unable to identify the validity of motion vector, some measure is required to ensure that the motion vector from the Compressed Domain is reliable to be used in video object segmentation. The motivation of carrying out this investigation comes from the publication by Coimbra and Davies [4] that draws the connection between information from the Compressed Domain and Lucas and Kanade optical flow method. The result is an accurate motion estimation scheme that is independent of GOP structure and approximates closely to a Lucuas-Kanade optical flow method.

Coimbra and Davies' discovery triggered the interest to find the connection between optical flow and the Compressed Domain, in particular the use of confidence measure.

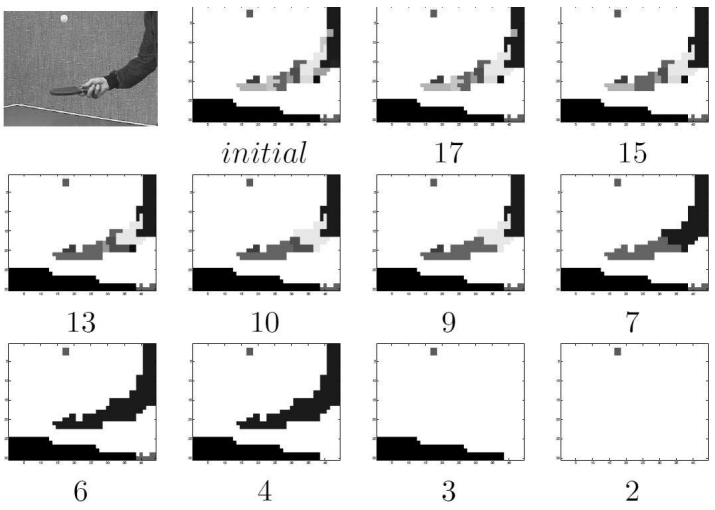



**Figure 2. Porikli et al.'s segmentation results at the corresponding clustering levels. Note the volume growing process could not blend the lower part of the arm into other regions since its DCT coefficients were also significantly different.[10]**

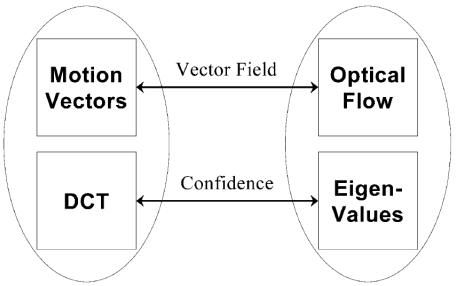



**Figure 3. Comparison of the LK and MPEG-2 system[4]**

## 3 Optical Flow

The first formal definition of optical flow is found in the publication by Horn and Schunck [6], in that "optical

flow is the distribution of apparent velocities of movement of brightness patterns in an image". According to [6], the optical flow problem is formulated as follows.

Let a brightness value at point (x,y) at time t be E(x,y,t), and the x- and y-component of optical flow be u and v respectively. For small motion (such that a point in the moving brightness pattern remains constant),

$$\frac{dE}{dx}u + \frac{dE}{dy}v + \frac{dE}{dt} = 0 \quad (1)$$

or,

$$\nabla I^T \cdot (u, v) = -\frac{dE}{dt} \quad (2)$$

where $\nabla$ I is the gradient of image intensity.

Since the constraint that a point in the moving brightness pattern is constant is not enough to derive the value of (u,v) (often referred to as the Aperture Problem in literature), additional constraints has to be applied in addition to the above equation. For example, Horn and Schunck [6] introduced a smoothness constraint in that the velocity field of the brightness patterns in the image varies smoothly.

Following the classification in Barron et al.[7], methods of approximating optical flow is divided into four approaches:

1. Differential Techniques
2. Region-based Matching
3. Energy-based Methods
4. Phase-based Methods

In comparing the performance of optical flow techniques [7] emphasized on the accuracy of the optical flow measurements. They found out that in general the local differential approaches gives the most accurate results, with the method proposed by Lucas and Kanade [9] being the most accurate and least expensive. In addition, Barron et al.'s assessment [7] point out the importance of confidence measures and thresholds in their publication, stating that the use of confidence ensures the accuracy of the approximated optical flow fields.

The optical flow method proposed by Lucas and Kanade [9] aims to find the disparity vector h that minimizes the difference between the original image F(x) and the translated image G(x). Their generalized algorithm, which can register translation as well as rotation, scaling and shearing, can be expressed as

$$G(x) = F(xA + h) \quad (3)$$

where A is a matrix of linear transformations for each pixel inside the region of interest R, in order to find the disparity which minimizes the sum of squared differences, i.e.

$$\sum_x [F(xA + h) - G(x)]^2 \quad (4)$$

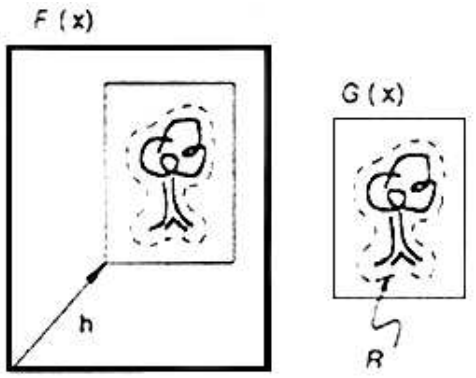



**Figure 4. Lucuas and Kanade formulate the problem as the search for the disparity vector h which minimizes the difference between F(x + h) and G(x), for x in some region of interest R.[9]**

## 4 Compressed Domain and Optical Flow

The process of Motion Compensation in video compression approximates optical flow calculation. The block-matching step, in that the encoder looks for the motion vector that gives the least difference between the reference and predicted macroblocks, resembles the optical flow approximation by Anandan [1] and Lucas and Kanade [9], in which both methods searches for the displacement with least error in the search window.

As a consequence, any confidence measure used in optical flow techniques could apply to the Compressed Domain. Coimbra and Davies [4] associates MPEG motion vectors and horizontal- and vertical-frequency DCT coefficients with optical flow and eigenvalues for confidence measure.

Unfortunately, this implies that the limitations in the optical flow methods also applies to Video Object Segmentation in the Compressed Domain. Aperture problem is present in all optical flow algorithms, and optical flow from less-textured image ares tends to be inaccurate. More importantly, sharp changes in intensities, such as occlusions and opening/closing of background lights. Such problems could be perhaps addressed separately, as in the Wallflower algorithm [12].

## 5 Conclusion

This paper presents the findings that are related to the acquisition of accurate Video Object Motion from the Compressed Domain. Many Compressed Domain video object segmentation algorithms involves the use of Motion Vectors in the input bit stream, which resembles the approximation of optical flow methods. While using these Motion Vectors saves the work of finding the optical flow for segmentation, it introduces the difficulties introduced by the fact that the motion vectors in the video bitstream does not necessarily reflect true motion.

Remedies have been introduced to deal with inaccurate Motion Vectors. A common approach is to accumulate Motion Vectors over a few picture frames then perform filtering to remove outliers. This approach, however, introduces problems such as motion cancelation and inclusion and inaccurate motion vectors. A better approach is to introduce confidence measures to remove potentially inaccurate Motion Vectors.

The application of optical flow methods in video object segmentation in the Compressed Domain, in particular the use of confidence measure, is an interesting topic and deserves further investigation.

## References


[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, January 1989.

[2] 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.

[3] 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.

[4] 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.

[5] K. Hariharakrishnan and D. Schonfeld. Fast Object Tracking Using Adaptive Block Matching. *IEEE Transactions on Multimedia*, 7(5):853–859, October 2005.

[6] B. K. P. Horn and B. G. Schunck. Determining Optical Flow. *ARTIFICAL INTELLIGENCE*, 17(1-3):185–203, August 1981.

[7] B. J.L., F. D.J., B. S.S., and T. Burkitt. Performance of optical flow techniques. pages 236–242, June 1992.

[8] 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.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. pages 674–679, 1981.

[10] 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.

[11] 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.

[12] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance, 1999.

# Detecting, Locating, and Tracking Hacker Activities within a WLAN Network

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

*When it comes to positioning technology, people usually think about the Global Positioning System (GPS). However, GPS, although mature enough to be used for navigation and tracking of goods, it is not effective in indoor environment. On the other hand, with the advance in WLAN technology and the popular adoption of wireless equipments and in particular, the IEEE 802.11 family, almost everyone in the community has a Wi-Fi enabled device integrated into their everyday life. With a good location estimation integrated into a Wi-Fi surveillance system, system administrator can closely monitor the network traffic as well as the behavior of the mobile users. Hence, there is a growing demand to have a quick and efficient way to indentify a specific group of people, or devices or asset within a controlled wireless network. In our proposed system, all the Wi-Fi traffic or information especially the MAC addresses and RSSI from the mobile clients (i.e. Wi-Fi devices) can be sniffed by an open-source Wi-Fi router or access point with custom-made embedded software program without pre-loading any client program on the mobile user devices. These sniffed information is then analyzed and stored in a database which will help network administrator to monitor the wireless network for surveillance purpose and security concerns. In summary, this paper proposes a wireless LAN system that can detect, locate and track down wireless communication within the system by modifying the embedded software in off-the-shelf WLAN routers or access points. Experiment results have shown that abnormal wireless activities can be detected and by our signal strength based localization algorithm, positions of these wireless mobile devices can be identified and be tracked within meters inside our WLAN system.*

## 1. Introduction

Under the Government's Digital 21 Strategy [1] to build Hong Kong into a wireless city, the HKSAR government had put forward a Wi-Fi Programme (GovWiFi) [2] as an initiative to create a wireless infrastructure to facilitate Internet access by citizens and businesses for enhancing quality of living and business operations. Other than the intended value-added services on the Internet, this programme also raises serious concern about hacking activities within the WLAN networks. In view of the heavy usage and progressively growing coverage of WLAN within the indoor environment, and the relatively few research studies on effective ways to do WLAN positioning and tracking, there is a need to investigate the feasibility of using the WLAN to locate a mobile user as well as tracking hacking activities in an indoor environment.

Wi-Fi becomes a much more valuable asset within a computer network; every mobile user can be granted a permission to use those Wi-Fi accesses within a permitted area. Consequently, mobile hackers and unauthorized access are possible within a mobile network, and no good tools are available to monitor the wireless network easily. In this paper, we proposed a back-end network surveillance system making use of a popular off-the-shelf wireless router by Linksys – (Model: WRT54G).[3]

It is important for us to consider a network which is capable of monitoring the access of the valuable wireless assets, and can be able to locate key personnel as well as mobile users. Every mobile device has a unique Media Access Control (MAC) address, and almost all the smart phones developed are equipped with a Wi-Fi chipset. When a mobile user turn-on the Wi-Fi access in his smart phone and entered into WLAN covered area, the message and packet exchanges with its corresponding signal strength become the fingerprint or the identifier for the mobile user within the network. Hence, there is a need to conduct a research to build an efficient and effective surveillance network to improve the system administrator's sensitivity to detect, locate, and to track hackers within a Wi-Fi network.

Besides, the estimation of the location of the hackers, or harmful devices can be useful for system administrator, as well as policeman to determine the location of those users, as to minimize the time for them to stop the un-authorized access to the network. As wireless networks are installed everywhere, the maintenance and fine tuning of the network become very complicated especially when hacker's activities are involved.

We very much relying on the existing intrusion detecting algorithm for detecting hackers' presence in a WLAN, but for detecting the hacker's location, there are some constraints when constructing our location estimation algorithms. First of all, we cannot assume that we can plant a client program in the hacker's device for data acquisition, that is for collecting the signal strength information at the mobile device for location estimation. Furthermore, we cannot assume that the hacker devices will behave like an ordinary mobile device, because they can change their SSID at will, tune up and tune down the antenna power for bigger fluctuation of signal strength in order to avoid being detected and located [4]. Hence, our approach is to enhance and rewrite embedded software at the Access Point (AP) side to "listen" to "conversations" among all the APs and wireless devices within the coverage of the WLAN network. In more technical terms, we will write software and re-program the AP and wireless Routers such that our software can collect signal strength readings when wireless devices, especially those hacker devices, that are communicating with any of the surrounding APs. [5] With these signal strength readings collected in real-time and stored in the data store at our data centre, together with the known positions of the APs, we can reuse our previous algorithms [6-12] to find out the whereabouts of the hacker devices.

Besides tracking hacker's activities within the WLAN, while some of the Wi-Fi devices are attached to the network all day, some of the Wi-Fi devices are only attached to the network in the daytime, but not at night. Thus, the ability to monitor the health and status of major components and key devices, such that we can receive prompt notification when changes appear, also serve an important role for network maintenance.

Finally, there are a number of ways to locate a mobile user based on RSSI. Some of the fundamental position techniques

are Location fingerprinting (LF), Propagation Loss Model (RF), Tri-lateration, Tri-angulations and Radio Maps with pattern recognition approaches. [13-17]. As yet another location estimation algorithms is not our main focus in this paper, for easier deployment of our system, and with the limited computation power of our system servers, a center of gravity (CG) method for location estimation is adopted in our system just for demonstrating the effectiveness for network surveillance purpose. [18]

## 2. Related work

In previous years, our research group had done quite a number of work on mobile phone positioning and results are astounding [19-30]. Although theoretically we can use the same or similar location estimation algorithms for the mobile phone network to be applied to the WLAN, there is relatively few researches on WLAN positioning [31-34] in the literature. The problems are mainly due to the differences in penetration power, signal fading, signal attenuation, the layout of access points (regularity vs. randomness), and the more serious body effect within the WLAN as compare to the mobile phone network. Although many researchers have proposed methods in providing location services using the mobile phone networks, few projects have actually been implemented. The RADAR system [32] was one of the early WLAN-based location estimation systems. Based on the FreeBSD distribution and WaveLAN WLAN network, the RADAR system can locate a user who carries a notebook with an accuracy of 5 meters. Y. Wang et al [31] did have a study on the feasibility for making use of the WLAN to locate a mobile device. They had done an empirical study inside their department building and labs and reported their findings and simulation results. However, there is no systematic way to generalize these approaches and in fine tuning the necessary environment parameters. Furthermore, the target environment is critical to the quality of positioning. Radio characteristics in an open environment are never static and there is no single methodology that can fine tune data from time-to-time to adapt with the changing environment. It is only recently that they starts to have studies on post-deployment adaptation issues. These include installing special hardware which monitors radio characteristics at different position and rebuild the propagation model from time-to-time [33, 34]. Among the commercial products that are available for WLAN positioning, ekahau is the company that is taking a leading position in in-door positioning. Their positioning system relies on building an accurate radio map according to the layout of the access points as well as the model and specific characteristics of these access points. Regardless that it is a costly system, it also subject to environmental changes and post-deployment adaptation problems.

## 3. Proposed System and Technologies Involved

Electronics, computers and wireless devices are becoming more in-expensive, low-power usage and multifunctional. And recent open-source wireless router has fast data processor that can handle real-time data acquisition for wireless network surveillance as well as location estimation for wireless devices.

In our experiments, a programmable router - Linksys WRT54G, which is burned with an open-source custom-made firmware can act as a wireless sensor to obtain information from data packets within the wireless environment. Service Set Identifier (SSID), Extended Service Set Identifier (ESSID), Received Signal Strength (RSSI), Noise Level, Traffic rate and Traffic Frequency… etc can be collected by a custom-made wireless data acquisition application written for the Linksys WRT54G WLAN router. In particular, the data acquisition application is a cross-compiled program for the 32-bit MIPS architecture processors manufactured by Broadcom, and this embedded piece of software plays an important role to achieve our goals in locating the hacker devices.

In short, we proposed a surveillance system to let the network administrator to monitor and analysis the network traffic and behavior. By making use of multiple wireless routers (Linksys WRT54G) burned with our custom-made cross-compiled program. Useful information is extracted, communications among all wireless devices are sniffed and being stored into our database server at the data centre [35]. With this information, locations of the mobile users can be estimated and suspected hacker's activities can be detected, identified, located and tracked with our WLAN environment.

Since we have to re-program the wireless router for packet sniffering, we have done lots of compatibility test on different open-source wireless routers and test out the feasibility. We found out that most wireless routers are using chipset from two major chipset manufacturers – Atheros and Broadcom.

We found out that for those who had chipset from Atheros, we can make use of the public open-source system call to switch the router into the monitor-mode as well as to the deep-monitor mode which can sniff all Wi-Fi traffic from the air.

On the other hand, for those routers who adopted the chipset from Broadcom, like the Linksys WRT54G, we cannot obtain the correct monitor mode to obtain necessary information for location estimation by using the usual system calls from iwlib.lib, wl.lib, or iwlist.lib… etc. So we have to rewrite the code from the pcap library, the prism library and code from wlioctl [36-38] to extract the Wi-Fi packet one by one in order to obtain the necessary information for location estimation.

Our custom-made program for Linksys WRT54G (AP) thus turned this wireless router into a wireless capture device which requires no authentication or access privileges in order to help us to do security auditing and to estimate the location of the hacker. And thus, we have made the AP to support monitor mode, such that we can disclose and decode the 802.11 frame information.

Wireless measurement can be done at both AP and mobile device level. For mobile device like a notebook, a D-Link PCMCIA Wi-Fi card with Atheros chipset can do the best job in extracting Wi-Fi frame information. Besides, we use the Nokia N96 mobile phone which is running Symbian 3.2 OS, to be programmed to extract the WLAN information. When the mobile device is associated to AP, both sides can obtain the RSSI data from each other, so that the upstream and downstream RSSI value can be obtained simultaneously.

In our system, all the APs sniffed data packets over the channels and each of these packets is being analyzed and stored in our data store. When a mobile user entered our monitoring areas (i.e. the WLAN), the MAC address, Received Signal Strength (RSSI) and relevance information were sniffed by the modified APs. Any data packet in the air will be captured not only by one AP, but also be captured by any APs that can detect and decode these packets.

The value of RSSI and the inter-distance between the mobile device and the AP varies with the inverse square law and is

somehow affected by interference, noise…etc. Indoor positioning technology or the technique for locating a mobile user is the base technology for indoor location-aware computing. In general, there are a large variety of location-based applications and services that rely on an accurate and stable location estimation system such as warehouse management, point-of-interest, infotainment and customer/consumer flow analysis within an enclosed area like a shopping mall or exhibition centre. Such location-based services play a crucial part in enabling e-commerce, and m-commerce, and eventually a critical part to bring the community to the era of ubiquitous/pervasive computing. [39]

Furthermore, at the mobile device, we had tried to formulate a RF signal propagation Loss Model based on the free space loss equation Lp(db)=20*LOG(f)+20*LOG(d)-function(fx) [40], where d is the inter-device distance in meters, f being the carrier frequency in MHz (i.e. at about 2400MHz according to the 802.11 standard due to difference channel), and function(fx) being the signal loss function due to obstacles like office furniture, book shelves and file cabinets. Calibration for function(fx) is needed especially for Multi-path fading and interference.

Thus, in summary, each AP in the WLAN is running a custom-made program to sniff data packet in the open air within the coverage of the WLAN network. Useful information is extracted from these data packets, including MAC address of the mobile device and the signal strength received by the APs are transferred to a control server, and then stored into a database together with the current timestamp for logging purpose. Later on, these data is analyzed by the control server and locations of the mobile users are estimated in real-time by our location estimation algorithm. In this paper, we used the center of gravity algorithm to estimate the mobile location within our test-site.

For better visualization, a program is written such that it will show all the mobile users at their estimated locations within the detectable area of our WLAN network. On the loading of our system, according to the database, less than 300 mobile users and Access Points can be detected and monitored within an hour during the daytime working hours.

## 4. Experiments and Results

### 4.1 Experiment 1

Figure 1 shows the layout of APs for Experiment 1 with obscured objects like tables, chairs, bookshelves and benches in a lab of the Research Centre for Ubiquitous Computing (RCUC) at HKBU. During the experiment, our system can detect the presence of about 25 APs. Within the coverage area and we looked into the inter-device distance and the corresponding RSSI under signal attenuation, multipath, reflection, refraction, and signal interferences. Within the WLAN, we obtain signal (data) mainly from three APs, which is BUAP7, BUAP9 and BUMAIN, others APs are just treated as wireless devices. The number in the middle is the distance unit between the AP and the wireless device. For example, distance from BUAP7 to BUAP1 is 3.28m. In this experiment, about 50000 samples were taken for each wireless device. The inter-device distance, that is, the distance between the sender and the receiver, and the received RSSI value is plotted out for investigation.

From Figure 1, BUMAIN and BUSTAFF1 were placed at same location, but only BUMAIN is used to grab packets from the others APs. On the other hand, BUAP7 and BUAP9 were also placed at same location, and both of them were used to grab packets from the other APs. This setup used to demonstrate intra-device error, the error reading on RSSI when the same equipment with the same distance to others APs is used.

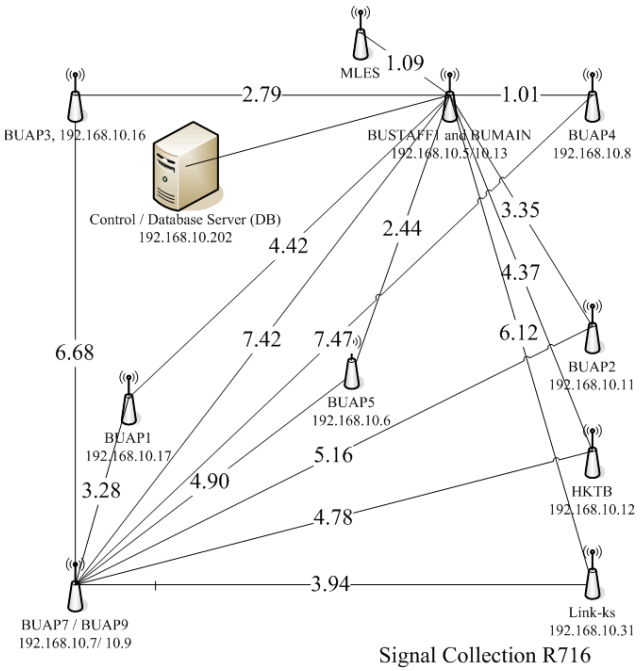



Figure 1. The Setup for Experiment 1 at a Lab in RCUC.

### 4.2 Experiment 1 Result and Analysis

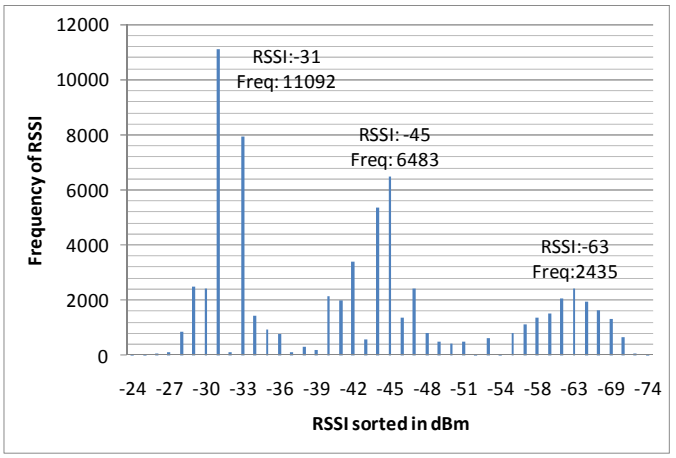



Figure 2. Experiment 1

Figure 2 shows the RSSI received at BUAP1 from BUAP7. The distance between the two routers is fixed at 3.28m. From Figure 2, there is a good indication that three peaks occurred at different RSSI readings, that is, at-32dbm, -45dbm, and -63dbm. This clearly shows that signal transmissions are influenced by multi-path, reflection, refraction, absorption, constructive as well as destructive interferences within this test-site. This raises an important question on how we should relate RSSI readings with the inter-device distance. And thus, we look into the average, the mode and the average of the top 10% readings of the RSSI.

| Marker ID | Distance (meter) | Avg. RSSI (dbm) | Mode. RSSI (dbm) | Top 10% (dbm) |
|---|---|---|---|---|
| BUAP4 | 1.01 | -36.36 | -41 | -28.08 |
| BUAP5 | 2.44 | -37.94 | -37 | -31.55 |
| BUAP3 | 2.79 | -42.89 | -43 | -33.74 |
| BUAP2 | 3.35 | -45.05 | -41 | -36.49 |
| BUAP1 | 4.42 | -45.53 | -43 | -37.71 |
| link-ks | 6.12 | -49.23 | -50 | -45.4 |
| BUAP7 | 7.42 | -47.31 | -43 | -41.24 |
| BUAP9 | 7.42 | -49.77 | -50 | -42.98 |

Table 1, Sniffer AP at BUMAIN and RSSI Measurements.

Table 1 and Figure 3 show the signal strength data (RSSI) and a plot of RSSI against different inter-device distance. From Figure 3, all three curves show some fluctuations of signals across the range of inter-device distances. The curve on the "Mode of RSSI" fluctuates the most, followed by the curve with the "Average RSSI", and the Top 10% RSSI curve shown to be most stable, except the point at 6.12m where all three measurements do not follow its own trend. We highly suspected that there are strong destructive interference of signals at this marker. Anyhow, we find out that the Top 10% RSSI readings is a good indicator for the RSSI readings against the inter-device distances between the sender and receiver.

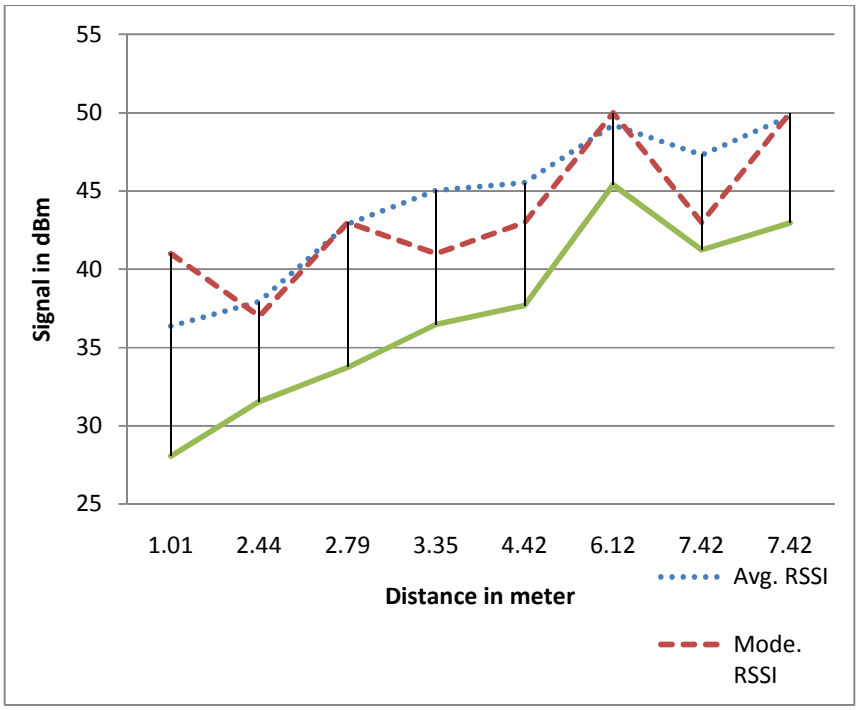



Figure 3. Sniffer AP at BUMAIN, RSSI Measurements vs. Inter-device Distances.

| Marker ID | Distance | Avg. RSSI | | | Mode. RSSI | | | Top 10% | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BUAP7 | BUAP9 | %Diff | BUAP7 | BUAP9 | %Diff | BUAP7 | BUAP9 | %Diff |
| BUAP9 | 0 | -24.37 | -21.00 | 16% | -18 | -25.42 | 29% | -12.28 | -11.80 | -4% |
| BUAP1 | 3.28 | -43.13 | -40.23 | 7% | -31 | -31 | 0% | -29.43 | -28.20 | -4% |
| link-ks | 3.94 | -47.86 | -50.92 | -6% | -38 | -39 | 3% | -34.78 | -37.72 | 8% |
| BUAP5 | 4.90 | -41.84 | -39.10 | 7% | -34 | -34 | 0% | -32.25 | -28.71 | -12% |
| BUAP2 | 5.16 | -47.71 | -42.78 | 12% | -42 | -34 | -24% | -38.96 | -31.84 | -22% |
| BUAP3 | 6.68 | -44.80 | -50.61 | -11% | -50 | -45 | -11% | -35.47 | -42.50 | 17% |
| BUMAIN | 7.42 | -49.78 | -48.65 | 2% | -53 | -45 | -18% | -41.57 | -42.88 | 3% |
| BUSTAFF1 | 7.42 | -50.67 | -47.92 | 6% | -51 | -45 | -13% | -44.62 | -40.67 | -10% |
| BUAP4 | 7.47 | -47.42 | -46.20 | 3% | -42 | -43 | 2% | -38.22 | -36.97 | -3% |
| | | | Average | 4.00% | | Average | -3.56% | | Average | -3.00% |

Table 2. Sniffer AP at BUAP7/BUAP9, RSSI Measurements vs. Inter-device Distances.

Table 2 shows the intra-device discrepancy on receiving signals. Signals are collected by both BUAP7 and BUAP9 where they are located at the same position during the experiment. Table 2 shows that of the three methods in measuring the RSSI readings, the top 10% RSSI reading produces the smallest differences with an average difference at -3.00%. And thus, Experiment 1 has shown that multi-path and interferences do exist and affect the signal readings and that we should use the "Top 10% RSSI" readings for estimating the inter-device distance between sender and receiver and the average differences in RSSI readings between two wireless routers varies between -3.56% to 4%

## 4.3 Experiment 2

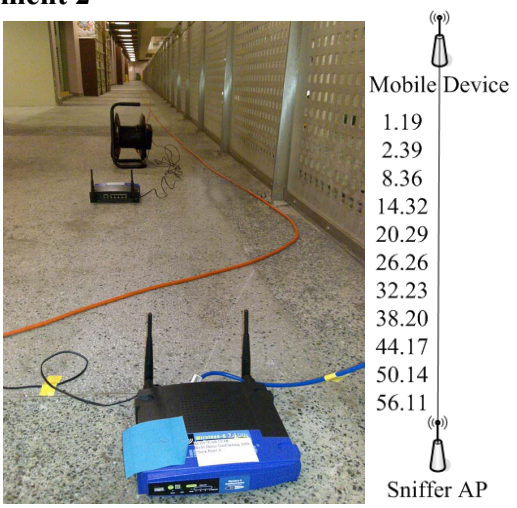



Figure 4. Signal Attenuation along a corridor.

Figure 4 shows the setup for Experiment 2 - the investigation on signal attenuation and distance between mobile devices in a semi-outdoor environment (outside corridor). Only 4 APs are used and we marked down 11 markers along a straight line. At each marker position, that is at positions shown below (1.19, 2.39, 8.36, 14.32, 20.29, 26.26, 32.23, 38.20, 44.17, 50.14, 56.11). 100 Samples are collected for our later analysis.

## 4.4 Experiment 2 Result and Analysis

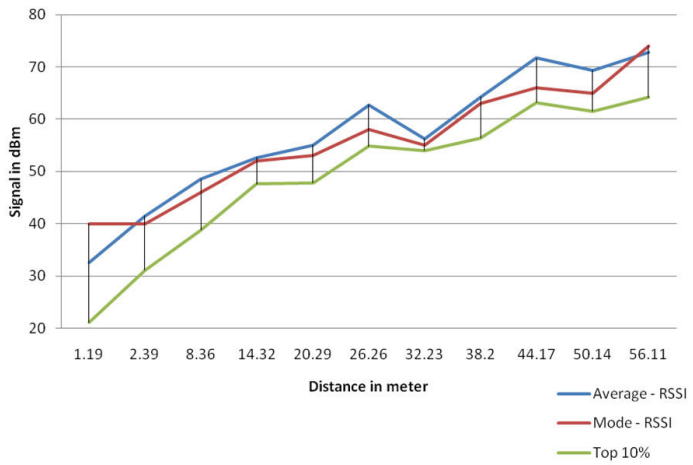



Figure 5. RSSI vs. Inter-device Distances in Experiment 2.

Unlike Experiment 1 which is conducted in an enclosed area, Experiment 2 is an investigation on the signal readings in a semi-open area – an open corridor outside our lab. When comparing Figure 3 and Figure 5, signals from Figure 5 (i.e. Experiment 2) are less fluctuating and more stable. There is a nice correlation between signal strength and inter-device distance indicating that a strong signal relates to a shorter inter-device distance and a weak signal correlates to a longer inter-device distance. Furthermore, there is no good indication of multi-path and interferences of signals throughout the range of distances in the experiment which varies from 1.1m to 56.11m. This range of distance is much wider than that in our Experiment 1.

| Distance in meter | Average - RSSI | Mode - RSSI | Top 10% |
|---|---|---|---|
| 1.19 | -32.59 | -40 | -21.1 |
| 2.39 | -41.49 | -40 | -31.1 |
| 8.36 | -48.50 | -46 | -38.8 |
| 14.32 | -52.66 | -52 | -47.6 |
| 20.29 | -55.07 | -53 | -47.8 |
| 26.26 | -62.72 | -58 | -54.8 |
| 32.23 | -56.23 | -55 | -54.0 |
| 38.2 | -64.28 | -63 | -56.4 |
| 44.17 | -71.73 | -66 | -63.2 |
| 50.14 | -69.38 | -65 | -61.5 |
| 56.11 | -72.82 | -74 | -64.2 |

Table 3. RSSI Measurements and Inter-device Distances in a semi-outdoor environment.

Table 3 shows the three proposed methods in measuring the RSSI, and similar to what is observed in Experiment 1 (Table 1), the Top 10% RSSI reading is the best among the three indicators for estimating the inter-device distance between the sender and receiver based on RSSI in this experiment.

## 4.5 Experiment 3

Figure 6 shows a similar experiment setup as in Experiment 1. It is used to demonstrate the accuracy of a location estimation method called Center of Gravity (CG). In this experiment, the distance between AP1 and AP3 is 6.68m, and the distance between AP1 and AP2 is 3.94m. Under the CG algorithm, the (x,y) co-ordinates are calculated as follows:

$$x = \frac{x_1 s_1^{-b} + x_2 s_2^{-b} + x_3 s_3^{-b} + \ldots + x_n s_n^{-b}}{s_1^{-b} + s_2^{-b} + s_3^{-b} + \ldots s_n^{-b}}$$

$$y = \frac{y_1 s_1^{-b} + y_2 s_2^{-b} + y_3 s_3^{-b} + \ldots + y_n s_n^{-b}}{s_1^{-b} + s_2^{-b} + s_3^{-b} + \ldots s_n^{-b}}$$

where (x,y) is the estimated location of the mobile user, $(x_1,y_1)$, $(x_2,y_2),\ldots,(x_n,y_n)$ are the locations of n receiving APs, and $s_1,s_2,s_3\ldots,s_n$ are the corresponding RSSI from each AP. The CG

algorithm has proven to be very effective and can provide outstanding performance in metropolitan area during our mobile location estimation experiments using the mobile phone network. However, the down side is that it can only estimate a mobile device inside the convex hull as defined by the APs involved.

## 4.6 Experiment 3 Result and Analysis

For Experiment 3, we assume a mobile user is carrying a wireless device and is performing some hacker's activities. Nine marker positions are defined and the mobile user will visit each marker in turn and signal information from the handheld device will be collected through the four access points - AP1, AP2, AP3, and AP4 as indicating in Figure 6.

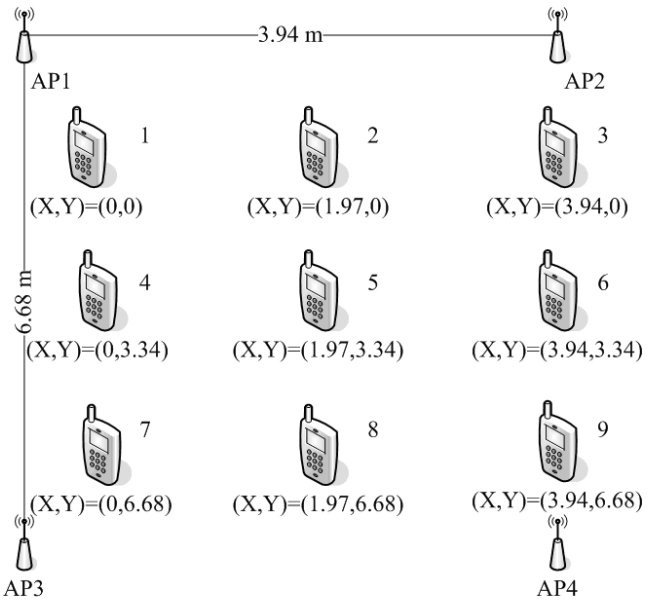



Figure 6. Localization of mobile devices within a lab in RCUC.

| Marker ID | Actual Position (X,Y) | | Estimated Position (X,Y) | | Error in meter |
|---|---|---|---|---|---|
| | x | y | x | y | |
| 1 | 0.00 | 0.00 | 0.45 | 0.70 | 0.83 |
| 2 | 1.97 | 0.00 | 1.70 | 3.27 | 3.28 |
| 3 | 3.94 | 0.00 | 2.94 | 1.25 | 1.60 |
| 4 | 0.00 | 3.34 | 2.04 | 2.66 | 2.15 |
| 5 | 1.97 | 3.34 | 2.09 | 3.29 | 0.13 |
| 6 | 3.94 | 3.34 | 1.78 | 2.47 | 2.33 |
| 7 | 0.00 | 6.68 | 0.86 | 5.21 | 1.71 |
| 8 | 1.97 | 6.68 | 2.34 | 3.91 | 2.80 |
| 9 | 3.94 | 6.68 | 2.57 | 4.44 | 2.63 |

Table 4. Accuracy of the CG localization algorithm.

Knowing the physical location and coordinates of the four APs, and together with the signal information collected through these four APs, we will use the "Center of Gravity" location method to estimate the mobile user's location. Table 4 shows the actual positions of the nine markers as well as the estimated location of the mobile user at the corresponding marker. With the actual position and the estimated position of the mobile user, the relative error is calculated and listed in Table 4.

One can observe that the error in meter ranged from 0.13m at Marker 5 to 3.28m at Marker 2. The system can estimate the hacker's position when it is near the centroid of the WLAN network, and the accuracy deteriorates towards the rim of the convex hull as defined by the four access points.

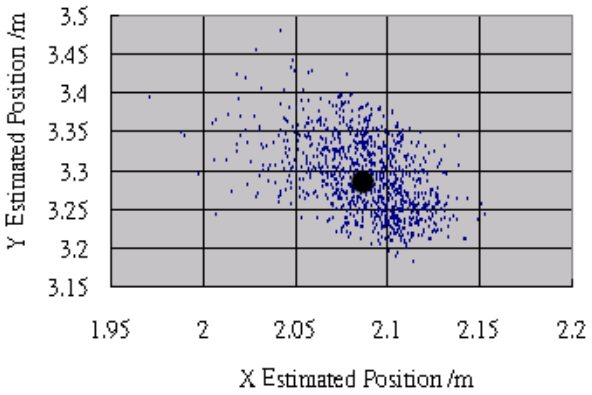



Figure 7, Marker #5 at (1.97,3.34)

Figure 7 shows the plot of the estimated position at various time instances at Marker 5 (1.97, 3.34). The plot itself shows the accuracy as well as the distribution of the estimated positions by the "Center of Gravity" algorithm with respect to the actual position of the mobile user at Marker 5. And the overall average error for the CG algorithm is 0.13m as shown in Table 4.

## 5. Conclusion and Future Work

The goal of the proposed project is to investigate the feasibility of using the WLAN to locate a mobile user as well as locating and tracking hacking activities in an indoor environment to enhance information security and to enable location-aware computing.

In the future, we are going to enhance the existing signal strength based location estimation methods for indoor location estimation base on the method we have done on our previous research paper, such as integrates two or more location estimation methods, and make use of RSSI collected from mobile terminals and/or from base-stations (WLAN access points) for a more stable location estimation. By the way, in order to use the Multi-fingerprinting method to nullify the body effect, we need to investigate how signals are collected, organized & stored, recognized & retrieved from location server and to reduce the cost of data acquisition and the cost of maintaining the signal data in the database up to date. Furthermore, we have to study and provide practical solutions for locating and tracking hackers' activities within the wireless network, and to investigate the feasibility and practicality of integrating our methods into the GovWiFi project so as to provide location based services and extra security services for better location-aware computing in Hong Kong. With such a development platform, application builders can develop numerous location based services and applications ranging from warehouse and resource management, indoor workers deployments, infotainment, and personal safety

Finally, we are entering a new era of computing – the era of ubiquitous/pervasive computing where we can retrieve any data

from any device using any network at anytime and any place, and this is the technology that will bring us a step closer to a more secure and more convenient ubiquitous computing society!

## 6. References


[1] "Hong Kong Government's Digital 21 Strategy" at http://www.info.gov.hk/digital21/eng/index.htm

[2] "Government Wi-Fi Programme (GovWiFi)" at http://www.gov.hk/en/theme/wifi/program/index.htm

[3] "Linksys WRT54G series" at http://en.wikipedia.org/wiki/Linksys

[4] 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.

[5] "Open Source Fireware for Access Point" at http://www.opertwrt.org

[6] William H. Wong, Joseph K. Ng, and Wilson M. Yeung, "Wireless LAN Positioning with Mobile Devices in a Library Environment", Proceedings of ICDCS-MDC 2005 Workshop, pp. 633-636, June 6-10, 2005, Columbus, Ohio, USA.

[7] Zhili Wu, Chun-hung Li, Joseph K. Ng, "Improvements to RADAR Location Classification", in Proceedings of the 2008 International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2008)

[8] Joseph K. Ng, Junyang Zhou, Kenneth M. Chu, and Karl R.P.H Leung, "A Train-Once Approach for Location Estimation using the Directional Propagation Model (DPM)", IEEE Transactions on Vehicular Technology, 57(4), pp. 2242-2256, July 2008.

[9] Junyang Zhou, Wilson M. Yeung and Joseph K. Ng, "Enhancing Indoor Positioning Accuracy by utilizing signals from both the mobile phone network and the Wireless Local Area Network", in Proceedings of the IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), pp 138-145, March 25-28, 2008, GinoWan, Okinawa, Japan. IEEE Computer Society Press.

[10] Wilson M. Yeung, Junyang Zhou and Joseph K. Ng, "Enhanced Fingerprint-based Location Estimation System in Wireless LAN Environment", in Proceedings of the 1st International Workshop on System and Software for Wireless SoC (WSOC 2007), pp 273-284, December 17-20, 2007, Taipei, Taiwan, Springer.

[11] Wilson M. Yeung and Joseph K. Ng, "Wireless LAN Positioning based on Received Signal Strength from Mobile device and Access Points", in Proceedings of the 13th International Conference on Embedded and Real- Time Computing Systems and Applications (RTCSA 2007), pp. 131-137, August 21-23, 2007, Daegu, Korea 2007, IEEE Computer Society Press.

[12] Wilson M. Yeung, and Joseph K. Ng, "An Enhanced Wireless LAN Positioning Algorithm based on the Fingerprint Approach", Proceedings of IEEE TENCON 2006, IEEE CS Press, 14-17 November 2006, Hong Kong, China.

[13] Kenneth M. Chu, Karl R. Leung, Joseph K. Ng, Chun Hung Li, "A Directional Propagation Model for Locating Mobile Stations within a Mobile Phone Network", International Journal of Wireless and Mobile Computing (IJWMC): Special Issue on Applications, Services and Infrastructure for Wireless and Mobile Computing, 3(1/2) pp. 12-21, August 2008, Inderscience.

[14] William H. Wong, Joseph K. Ng, and Karl R.P.H. Leung, "Large-Scale Location Estimation over GSM networks: the GEAR Approach", Proceedings of the 24th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS-MDC 2004), pp. 574 --- 579, March 23-26, 2004, Hachioji, Tokyo, Japan.

[15] William H. Wong, Joseph K. Ng, and Karl R.P.H. Leung, "Large-Scale Location Estimation over GSM networks: the Gear Approach", to appear in International Journal of Wireless and Mobile Computing.

[16] Zhi-li Wu, Chun-hung Li, Joseph K. Ng, and Karl R.P.H. Leung, "Location Estimation via Support Vector Kernel Regression", IEEE Transactions on Mobile Computing, Vol. 6, No. 3, pp. 311-321, March 2007.

[17] P. Bahl and V.N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system”, in Proceedings of INFOCOM 2000, (2): 775-784. Available online at citeseer.ist.psu.edu/bahl00radar.html

[18] Junyang Zhou, Enhanced Signal Propagation Models and Algorithm Selector for Providing Location Estimation Services within Cellular Radio Networks, Hong Kong Baptist University 2007

[19] Melvyn Wong and Joseph K. Ng, "Fleet Management System", Innovation and Technology Fund (UIM), Innovation and Technology Comission, HKSAR Government.

[20] Duncan Lau, Joseph Ng, Karl Leung, Lawrence Cheung, "Develop an accurate low-cost Mobile Location Estimation System (MLES) for fleet management applications using existing mobile phone infrastructure ", Innovation and Technology Fund (ITSP), Innovation and Technology Comission, HKSAR Government.

[21] Stephen K. Chan, Kenny K. Kan, and Joseph K. Ng, "A Dual-Channel System for Providing Location Estimation in Mobile Computing", Journal of Interconnection Networks (JOIN), Volume 4, Number 3, pp. 271 --- 290, September 2003, World Scientific Publishing Company.

[22] Kenny K.H. Kan, Stephen K.C. Chan, and Joseph K. Ng, "A Dual-Channel Location Estimation System for providing Location Services based on the GPS and GSM Networks", Proceedings of The 17th International Conference on Advanced Information Networking and Applications (AINA 2003), pp. 7 - 12, March 27-29, 2003, Xi'an, China.

[23] Joseph K. Ng, Stephen K. Chan, and Kenny K. Kan, "Location Estimation Algorithms for Providing Location Services within a Metropolitan Area based on a Mobile Phone Network", Proceedings of the 5th International Workshop on Mobility Databases and Distributed Systems (MDDS 2002), pp. 710 --- 715, Aix-en-Provence, France, September 2-6, 2002

[24] Karl R.P.H. Leung, Joseph K. Ng, Tim K. Chan, Kenneth M. Chu, and Chung Hung Li, "Network Based Mobile Station Positioning in Metropolitan Area", Proceedings of the International Conference on Parallel and Distributed Computing (Euro-Par 2003), pp. 1017 --- 1026, Springer-Verlag, August 26-29, 2003, Klagenfurt, Austria.

[25] Ka Ho Kan, "Location Estimation System Based on the GSM Network", M.Phil. Thesis, Hong Kong Baptist University 2004.

[26] Kenneth M. Chu, Karl R.P.H. Leung, Joseph K. Ng, and Chun H. Li, "Locating Mobile Stations with Statistical

Directional Propagation Model", Proceedings of the 18th International Conference on Advanced Information Networking and Applications (AINA 2004), pp. 230 --- 235, March 29-31, 2004, Fukuoka, Japan.
[27] Kenneth M. Chu, Karl R. Leung, Joseph K. Ng, Chun Hung Li, "A Directional Propagation Model for Locating Mobile Stations within a Mobile Phone Network", International Journal of Wireless and Mobile Computing (IJWMC): Special Issue on Applications, Services and Infrastructure for Wireless and Mobile Computing, 3(1/2) pp. 12-21, August 2008, Inderscience.
[28] William H. Wong, Joseph K. Ng, and Karl R.P.H. Leung, "Large-Scale Location Estimation over GSM networks: the GEAR Approach", Proceedings of the 24th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS-MDC 2004), pp. 574 --- 579, March 23-26, 2004, Hachioji, Tokyo, Japan.
[29] William H. Wong, Joseph K. Ng, and Karl R.P.H. Leung, "Large-Scale Location Estimation over GSM networks: the Gear Approach", to appear in International Journal of Wireless and Mobile Computing.
[30] Zhi-li Wu, Chun-hung Li, Joseph K. Ng, and Karl R.P.H. Leung, "Location Estimation via Support Vector Kernel Regression", IEEE Transactions on Mobile Computing, Vol. 6, No. 3, pp. 311-321, March 2007.
[31] Y. Wang, X. Jia, H.K. Lee, and G.Y. Li, "An indoor wireless positioning system based on wireless local area network infrastructure", Proceedings of the 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services (SatNav 2003), Melbourne, Australia, 22-25 July 2003.
[32] P. Bahl and V.N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system”, in Proceedings of INFOCOM 2000, (2): 775-784. Available online at citeseer.ist.psu.edu/bahl00radar.html
[33] P. Krishnan, A.S. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu, “A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks”, in Proceedings of INFOCOM 2004, 2004.
[34] S. Ganu, A.S. Krishnakumar, and P. Krishnan, “Infrastructure-based location estimation in WLAN networks”, in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2004), 2004.
[35] Wilson M. Yeung, Wireless LAN Positioning in Indoor Environment, Hong Kong Baptist University 2007
[36] "Broadcom 802.11abg Networking Device Driver" at https://dev.openwrt.org/browser/trunk/openwrt/package/openwrt/include/wlioctl.h?rev=375,
[37] "pcapsource from Kismet" at http://www.google.com.hk/codesearch/p?hl=zh-TW#ACmMdBt9LZs/Firmware_Alchemy-pre5.4a.src.by.TheIndividual.tar.bz2|g0iVX931-Yk/Firmware_Alchemy-pre5_4/src/router/kismet/pcapsource.cc&q=WLC_GET_MONITOR
[38] "Partial driver for WAVELAN" at http://www.google.com.hk/codesearch/p?hl=zh-TW#p-OcbD5DbwM/packages/1.2/kernel/patches/49-3rdparty/MC16_vt_ar5k-20030509.tar|xlsugco9EAs/3rdparty/vt_ar5k/include/vt_wlan.h&q=prism_hdr_t
[39] Junyang Zhou, Kenneth M. Chu, and Joseph K. Ng, "An Improved Ellipse Propagation Model for Location Estimation in facilitating Ubiquitous Computing", Proceedings of the 11th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2005), pp. 463-466, August 17-19, 2005, Hong Kong.
[40] Junyang Zhou, Enhanced Signal Propagation Models and Algorithm Selector for Providing Location Estimation Services within Cellular Radio Networks, Hong Kong Baptist University 2007

*Gene Expression*

# Speeding up Scoring Module of Mass Spectrometry Based Protein Identification by GPUs

You Li, Kaiyong Zhao, Xiaowen Chu*, Jiming Liu
Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

Received on XXXXX; revised on XXXXX; accepted on XXXXX
Associate Editor: XXXXXXX

**ABSTRACT**

**Motivation:** Database searching is a main method for protein identification in shotgun proteomics, and till now most research effort is dedicated to improve its effectiveness. However, the efficiency of database searching is facing a serious challenge, due to the ever fast increasing of protein and peptide databases resulting from genome translations, enzymatic digestions, and post-translational modifications. On the other hand, as a general-purpose and high performance parallel hardware, Graphics Processing Units (GPUs) develop continuously and provide another promising platform for parallelizing database searching based protein identification to increase its efficiency.

**Results:** In this paper, we propose to systematically research on speeding up database search engines by GPUs for protein identification. Considering the scoring module is the most time-consuming part, we mainly utilize GPUs to speed it up. We choose two popular scoring method: firstly, SDP based method, which is chosen by X!Tandem, reaches a speedup of thirty to one hundred; secondly, KSDP, which is adopted by pFind, achieves a speedup of five to ten.

**Contact:** Xiao-wen Chu, chxw@comp.hkbu.edu.hk

## 1 INTRODUCTION

Protein identification is the basis of proteomics, the main character of which is high-throughput, with tandem mass spectrometry based shotgun approach as the technique of choice. Compared with other data analysis methods, database search engines have been the most stably and widely utilized, such as Mascot (Perkins et al., 1999), SEQUEST (Eng et al., 1994), pFind (Fu et al., 2004; Li et al., 2005; Wang et al., 2007), X !Tandem (Craig and Beavis, 2004), OMSSA (Geer et al., 2004), and Phenyx (Colinge et al., 2003). While most of research effort targets to improve effectiveness by the designing new scoring and validating algorithms, the efficiency of the database search engines are facing a serious challenge, due to the following reasons:

Firstly, the number of entries in protein sequence database is keeping increasing. Take IPI.Human for example, from v3.22 to v3.49, the count of the protein has increased nearly by 1/3 times. Moreover, together with the evolution of genome sequencing technologies, proteogenomic research wishes to adopt genome translated protein sequences to identify more protein. As an example, the EST database (Human.12.06) will be translated into 8,163,883 protein sequences, over 100 times larger than the human proteome IPI.Human.v3.49, which has only 74,017 protein sequences..

Secondly, increasing importance of considering semi- or non-specific digestion leads to 10 or 100 times more digested peptides respectively than specific digestion, as is shown in Table 1.

**Table 1.** The scale of peptide sequences under tryptic digestion

| Database | Proteins | Peptides (fully specific) | Peptides (non-specific) |
|---|---|---|---|
| Yeast | 6,717 | 741,476 | 120,464,808 |
| IPI-Human | 74,017 | 7,412,821 | 1,230,715,950 |
| Swiss-Prot | 398,181 | 34,764,218 | 5,605,491,572 |

peptide mass = 800 Da–6000 Da, peptide length = 4–100 amino acids, non-specifically digested peptide length = 4–50 amino acids, max missed cleavage sites = 2.

**Table 2.** The number of post-translationally modified peptides

| Modification sites | Num. modified peptides |
|---|---|
| 0 | 3,309,085 |
| 1 | 25,197,765 |
| 2 | 133,063,810 |
| 3 | 477,180,661 |
| 4 | 1,361,747,010 |
| 5 | 3,395,725,099 |
| 6 | 7,823,314,004 |
| 7 | 17,606,043,889 |
| 8 | 41,148,061,489 |
| 9 | 99,244,365,518 |

Note: database = IPI-Human V3.49, fully tryptic digestion, peptide mass = 800 Da–6000 Da, peptide length = 4–100 amino acids, max missed cleavage sites = 2. Ten modifications are specified: Oxidation (M), Phosphorylation (S, T, Y), Methylation (K, R), di-Methylation (K, R), tri-Methylation (K), and Acetylation (K).

Thirdly, post-translational modifications (PTMs) generate exponentially more modified peptides. Till now, over 500 types of PTMs exist in Unimod database (http://www.unimod.org). If we choose ten common varia-

*To whom correspondence should be addressed.

ble PTMs and limit the number of modification sites in a peptide to no larger than five, the number of tryptic peptides of the human proteome will be increased over 1000 times, as is shown in Table 2. At the same time, the generation speed of the mass spectrometer increases steadily.

One of the direct results from the above four increase is the large scale number of scoring between peptide and spectrum, which is the most compute intensive and time consuming part in the whole flow of protein identification. Profiling analysis shows that scoring module takes more than 90% of total identification time with both pFind and X!Tandem. Thus, speeding up scoring module is a promising method to increase the efficiency of protein identification, which could be conducted by parallelizing the scoring function.

Recently, Graphics Processing Units (GPUs), which has become a general-purpose and high performance parallel hardware, develop continuously and provide another promising platform for parallelizing scoring function. GPUs are dedicated hardware for manipulating computer graphics. Due to the large demand for computing real-time and high-definition 3D graphics, the GPUs have evolved into highly parallel many-core processors. The advantages of computing power and memory bandwidth in GPUs have driven the development of general-purpose computing on GPUs (GPGPU).

To the best of our knowledge, no research has ever attempted to parallel database search engine by GPUs, which could work as a small cluster or a much higher performance node inside a cluster. Thus, in this paper, we choose two popular scoring methods: SDP based method, chosen by X!Tandem; KSDP, adopted by pFind, and conduct systematic research on parallelizing the scoring function by using a general-purpose parallel programming model, namely Compute Unified Device Architecture (CUDA) [9, 10]. Our first contribution is firstly applying GPUs to speed up the protein identification. Our second contribution is the observation that the spectrum-peptide matching distribution is an important factor to be considered, based on which we design, implementation, and evaluation of two different strategies. For the spectrum which does not share matched peptides with other spectra, we mainly utilize the GPU on-chip registers and texture to minimize the memory access latency. For the spectra which share the same set of matched peptides, we design a novel and highly efficient algorithm that treats the scoring module as matrix multiplication, and then makes use of GPU on-chip shared memory together with on-chip registers. As a result, SDP gets a speedup of thirty to one hundred; KSDP achieves a speedup of five to ten.

The rest of this paper is organized as follows. Section II introduces some existing speedup methods, the GPU architecture and GPU application in bioinformatics. Section III presents our design of parallel scoring algorithm on GPUs. Section IV presents our experimental results, and Section V concludes the paper and presents some future work.

## 2 RELATED WORK

We firstly introduce the background knowledge for scoring method, present the existing speedup method, and illustrate the basic architecture of GPU.

### 2.1 Spectrum and Fragment ions

A peptide is a string of amino acid residues joined together by peptide bonds. In the mass spectrometer, peptides derived from digested proteins are ionized. Peptide precursors of a specific mass-charge ratio (*m/z*) are selected and further fragmented by collision-induced dissociation (CID). Product ions are detected. The measured *m/z* and intensity of the product ions form finally the peaks in the tandem mass spectrum (MS/MS spectrum), as shown in Fig 1. By CID, three kinds of backbone cleavages on peptide bonds can produce six series of fragment ions, denoted by N-terminal *a*, *b* and *c* type fragments and C-terminal *x*, *y* and *z* type fragments, as shown in Fig.2.

The scoring method computes the similarity between theoretical and experimental spectra, which could both be expressed as *N*-dimensional vectors, where *N* is the number of m/z values used. We use vector $\boldsymbol{c} = [c_1, c_2, ..., c_N]$ stand for the experimental spectrum and vector $\boldsymbol{t} = [t_1, t_2, ..., t_N]$ the theoretical one. $c_i$ and $t_i$ are binary values {0, 1} (or the intensity).

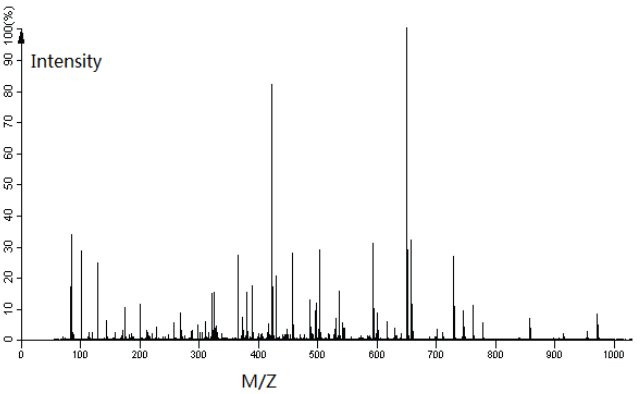



**Fig. 1.** An example of MS/MS Spectrum

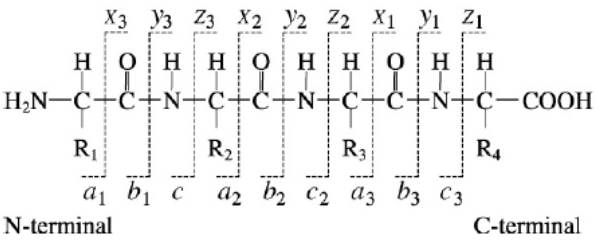



**Fig. 2.** Fragment ions from peptide bonds cleavage by CID

### 2.2 Speeding up methods

There are several researches on improving the design of classical database search engines, for example, Edwards & Lippert considered the problem of redundant peptides and peptide-spectrum matching (Edwards and Lippert, 2002), Tang et al. adopted peptide and b/y ions indexes (Tang et al., 2005), Dutta & Chen utilized the nearest neighbor search to improve peptide-spectrum matching (Dutta and Chen, 2007), and Roos et al. made use of hardware cache to speed up identification (Roos et al., 2007).

There are also various researches, based on tag, to improve the efficiency of protein identification. One of the most significant method is the peptide sequence tag (Mann and Wilm, 1994), and followed by GutenTag (Tabb et al., 2003), MultiTag (Sunyaev et al., 2003), InsPecT (Tanner et al., 2005), and Spectral Dictionary (Kim et al., 2008). In fact, extracting peptide tag or tags from the tandem mass spectrum is a very complicated process, due to the spectra resolution and accuracy, charge states, peptides sequence length. Consequently this method is still not as commonly adopted as traditional database search engines.

Obviously paralleling database search engines could achieve a high efficiency. pFind, X!Tandem, Sequest, and Mascot all have parallel version. In fact, all the above work could further increase the efficiency by GPUs. For the single PC based search engine, GPUs could work as a small cluster. For parallel version, GPUs could sharply increase the computing power of each node.

### 2.3 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-Thread (SIMT) 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, 12]. 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; second, if the number of threads in a block is not a multiple of warp size, the remaining instruction cycles will be wasted. Besides, when accessing the memory, *half-warp* executes as a group, which has 16 threads. If the half-warp threads access the coalesced data, the access operation will perform within one instruction cycle. Otherwise, the access operation will occupy up to 16 instruction cycles.

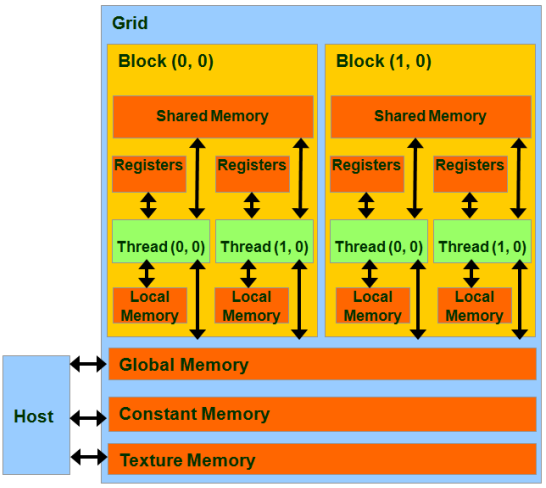



**Fig. 3.** Hardware architecture of the GPU

## 3 DESIGN OF PARALLEL SCORING MODULE

Profiling analysis shows that scoring module takes more than 90% of total identification time with both pFind and X!Tandem. Thus, in this paper, we mainly parallel scoring module by GPUs, and choose two widely used scoring methods from two popular search engines.

### 3.1 Spectral dot product in X!Tandem

The tandem mass SDP between the experimental and theoretical spectra is defined as

$$SDP = <c, t> = \sum_{i=1}^{N} c_i t_i \qquad (1)$$

The SDP-based cosine value of the angle between spectral vectors was adopted as a similarity measure (Wan *et al.*, 2002; Tabb *et al.*, 2003). In current peptide-scoring algorithms, the SDP also plays an important role, indirectly utilized in X!Tandem and SEQUEST.

Algorithm 1: CPU-based SDP

```
// C: the set of experimental spectrum
// c:  experimental spectrum
// T: the set of experimental spectrum
// t: experimental spectrum
1. for each c in C
2.     for each t in T
3.       if c.mass > t.mass-tol && c.mass < t.mass+tol
4.           for i from 1 to N
5.               SDP_Score += c_i t_i
6.           end of for
7.       end of if
8.     end of for
9.  end of for
```

Algorithm 2: GPU-based SDP

```
// C_i : the i_th spectrum in C
1.    i = threadId;
2.    for each t in T
3.      if C_i.mass > t.mass-tol && C_i.mass < t.mass+tol
4.          for j from 0 to N
5.              SDP_Score += C_ij t_i
6.          end of for
7.      end of if
8.    end of for
```

The algorithm of SDP is simple, as shown in Algorithm 1. Line 1~3, for each spectrum, find all the peptides whose precursor mass are in the spectrum's precursor mass window, assuming there are $m$ peptides; line 4~5 compute the SDP score between the experimental and theoretical spectrum. The computation complexity is $O(|C|\lg(|T|)Nm)$. Adopting GPUs, we can assign each peptide to one thread, scoring with its matched peptide, as shown is Algorithm 2. Obviously, the computation complexity decreases to $O(\lg(|T|)Nm)$.

Another widely used scoring method XCorr could also adopt the above parallel algorithm. XCorr is an important scoring part in SEQUEST, which a widely used protein identification software, but with a notorious searching speed. The process of XCorr is as equation 2 and 3. Obviously, calculating XCorr is like calculating SDP for 150 times.

$$R_\tau = \sum_{i=1}^{N} c_i t_{(i+\tau)} \qquad (2)$$

$$XCorr = R_{(\tau=0)} - \frac{1}{149} \sum_{-75<k<75} R_{(\tau=k)} \qquad (3)$$

### 3.2 KSDP in pFind

KSDP is a kernel based SDP scoring method, which significantly increase the effectiveness of SDP. The kernel trick is to compute directly the dot product in the correlative space with a proper kernel without an explicit mapping from the spectral space to the correlative space. Considering different kind of fragment ions, all the fragments are arranged in correlative matrix, as shown in Fig.4. All predicted fragments are assumed to possess unique m/z values so that all non-zero dimensions in the theoretical spectral vector, $\boldsymbol{t}$, can be extracted and rearranged into the matrix $\boldsymbol{T}=(t_{pq})_{m*n}$, where $m$ is the number of fragment types and $n$+1 is the residue number of peptide precursor. For example, $t_{2,3}$ corresponds to the fragment $b_3$ in Fig.4. The experimental spectral vector $\boldsymbol{c}$ could be organized in the same way.

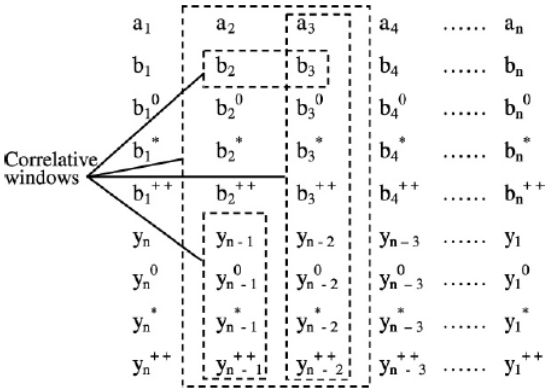



Fig. 4. Correlative matrix and correlative windows

$$K_{pep}(c,t) = \sum_{i=1}^{m}\sum_{j=1}^{n}[\sum_{k=j-l_1}^{j+l_2}(w_{|k-j|}(c_{ik}t_{ik})^{1/d})]^d \qquad (4)$$

Algorithm 5: CPU-based KSDP

```
//l: the size of correlative window
//l_1: ⌊(l−1)/2⌋, l_2: ⌈(l−1)/2⌉
//win: temp kernel value
//d: a parameter controls the degree of correlation
1.    for each c in C
2.       for each t in T
3.          if c.mass > t.mass-tol && c.mass < t.mass+tol
4.             K_ct = 0;
5.             for (i=1; i≤m; ++i)
6.                win_{i,1} = 0;
7.                for(j=1;j≤l_2; ++j)
8.                   win_{i,1} += (c_ij, t_ij)^{1/d};
9.                end of for
10.               K_ct += win_{i,1}^d;
11.               for (j=2; j≤n; ++j)
12.                  win_{i,j}=win_{i,j-1}+(c_{i,j+l2},t_{i,j+l2})^{1/d}-(c_{i, j-l1-1}, t_{i, j-l1-1})^{1/d};
13.                  K_ct += win_{i,j}^d;
14.               end of for
15.            end of for
16.         end of if
17.      end of for
18.   end of for
```

The process of KSDP is show as Equation 4 and Algorithm 5. Line 1~3 find the corresponding peptides for each spectrum; line 4~13 calculate the kernel function by traversing

the whole correlative matrix. The computation complexity is O(|*C*|lg(|*T*|)*mn*). Using GPUs, we can also assign each spectrum to one thread, scoring with all its corresponding peptides, as shown in Algorithm 6. And the computation complexity is O(lg(|*T*|)*mn*).

Algorithm 6: GPU-based KSDP

```
//
1.  i = threadID;
2.      for each t in T
3.        if C_i.mass > t.mass-tol && C_i.mass < t.mass+tol
4.           K_ct  = 0;
5.           for (i=1; i≤m; ++i)
6.              win_{i,1} = 0;
7.              for(j=1;j≤l_2; ++j)
8.                 win_{i,1} += (c_ij, t_ij)^{1/d};
9.              end of for
10.             K_ct  += win_{i,1}^d;
11.             for (j=2; j≤n; ++j)
12.               win_{i,j}=win_{i,j-1}+(c_{i,j+l2},t_{i,j+l2})^{1/d}-(c_{i, j-l1-1}, t_{i, j-l1-1})^{1/d};
13.               K_ct  += win_{i,j}^d;
14.             end of for
15.          end of for
16.       end of if
17.     end of for
18.  end of for
```

## 4 EXPERIMENT

We have implemented both CPU- and GPU based scoring algorithms using CUDA version 2.3. Our 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 calculate the time of the application after the file I/O, in order to show the speedup effect more clearly.

We compare the speed of CPU- and GPU based scoring algorithms, varying the number of the spectrum and peptide. To show the number of scoring and the time consumption more clearly, we let each spectrum score with a specific number of peptide.

As shown in Table 3, the speedup of SDP by GPUs is very favorable, from thirty to more than one hundred, mainly resulting from the distribution of each spectrum to each thread. Besides, owing to the simple calculation process of SDP, most of the work could be conducted on on-chip register without reading latency. We can also observe that the increase of the number of spectrum does not increase the time consumption, which means this simple parallel algorithm works well.

As is shown in Table 4, KSDP achieves a speedup of two to eight, which is not very favorable right now. The speedup mainly comes from the simple multi-thread without any optimization concerning memory usage.

**Table 3.** The speedup effect of SDP, in second.

| *Pep* | *Spec* | *On CPU* | *On GPU* |
|---|---|---|---|
| 1024 | 1024 | 3.74 | 0.11 |
| | 2048 | 7.49 | 0.14 |
| | 4096 | 14.97 | 0.16 |
| 2048 | 1024 | 7.51 | 0.14 |
| | 2048 | 15.02 | 0.18 |
| | 4096 | 30.01 | 0.22 |
| 4096 | 1024 | 15.07 | 0.18 |
| | 2048 | 30.26 | 0.23 |
| | 4096 | 60.28 | 0.31 |

**Table 4.** The speedup effect of KSDP, in second.

| *Pep* | *Spec* | *On CPU* | *On GPU* |
|---|---|---|---|
| 1024 | 1024 | 1.85 | 0.78 |
| | 2048 | 3.74 | 0.81 |
| | 4096 | 7.58 | 0.97 |
| 2048 | 1024 | 3.86 | 1.52 |
| | 2048 | 7.78 | 1.57 |
| | 4096 | 15.56 | 1.96 |
| 4096 | 1024 | 8.01 | 3.01 |
| | 2048 | 16.02 | 3.09 |
| | 4096 | 32.01 | 3.98 |

## 5 FUTURE WORK

As talked in section 4, KSDP has not got a very high speed-up effect, since we only use global memory. In the following work, we would:

Firstly, we would put the spectrum on texture, since each spectrum would score with multi peptides, so using texture with cache mechanism would decrease the reading latency.

Secondly, when the set of peptides grows, letting one thread deals with one spectrum is not surely a good idea. We need to make a new strategy: letting one thread dealing with one spectrum and a limited set of peptides, when the peptides grow, we will use multi thread to deal with one spectrum.

Thirdly, we find out that some spectrum with near precursor mass would score with the same set of peptides, which makes it possible to adopt shared memory to further optimize the algorithm.

## REFERENCES

Perkins, D. N.; Pappin, D. J.; Creasy, D. M.; Cottrell, J. S., Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **1999,** 20, (18), 3551-67.

Eng, J., An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry* **1994,** 5, (11), 976-989.

Fu, Y.; Yang, Q.; Sun, R.; Li, D.; Zeng, R.; Ling, C. X.; Gao, W., Exploiting the kernel trick to correlate fragment ions for peptide identification via tandem mass spectrometry. *Bioinformatics* **2004,** 20, (12), 1948-54.

Gronert, S.; Li, K. H.; Horiuchi, M., Manipulating the fragmentation patterns of phosphopeptides via gas-phase boron derivatization: determining phosphorylation sites in peptides with multiple serines. *J Am Soc Mass Spectrom* **2005,** 16, (12), 1905-14.

Gao, Y.; Wang, Y., A method to determine the ionization efficiency change of peptides caused by phosphorylation. *J Am Soc Mass Spectrom* **2007,** 18, (11), 1973-6.

Craig, R.; Beavis, R. C., TANDEM: matching proteins with tandem mass spectra. *Bioinformatics* **2004,** 20, (9), 1466-7.

Geer, L. Y.; Markey, S. P.; Kowalak, J. A.; Wagner, L.; Xu, M.; Maynard, D. M.; Yang, X.; Shi, W.; Bryant, S. H., Open mass spectrometry search algorithm. *J Proteome Res* **2004,** 3, (5), 958-64.

Colinge, J.; Masselot, A.; Giron, M.; Dessingy, T.; Magnin, J., OLAV: towards high-throughput tandem mass spectrometry data identification. *Proteomics* **2003,** 3, (8), 1454-63.

Mann, M.; Wilm, M., Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Anal Chem* **1994,** 66, (24), 4390-9.

Tabb, D. L.; Saraf, A.; Yates, J. R., 3rd, GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model. *Anal Chem* **2003,** 75, (23), 6415-21.

Sunyaev, S.; Liska, A. J.; Golod, A.; Shevchenko, A., MultiTag: multiple error-tolerant sequence tag search for the sequence-similarity identification of proteins by mass spectrometry. *Anal Chem* **2003,** 75, (6), 1307-15.

Tanner, S.; Shu, H.; Frank, A.; Wang, L. C.; Zandi, E.; Mumby, M.; Pevzner, P. A.; Bafna, V., InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. *Anal Chem* **2005,** 77, (14), 4626-39.

Datta, R.; Bern, M., Spectrum fusion: using multiple mass spectra for de novo Peptide sequencing. *J Comput Biol* **2009,** 16, (8), 1169-82.

Kim, S.; Gupta, N.; Bandeira, N.; Pevzner, P. A., Spectral dictionaries: Integrating de novo peptide sequencing with database search of tandem mass spectra. *Mol Cell Proteomics* **2008**.

Bafna, V.; Edwards, N. In *On de novo interpretation of tandem mass spectra for peptide identification*, RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology, 2003; ACM Press: 2003; pp 9-18.

Shilov, I. V.; Seymour, S. L.; Patel, A. A.; Loboda, A.; Tang, W. H.; Keating, S. P.; Hunter, C. L.; Nuwaysir, L. M.; Schaeffer, D. A., The Paragon Algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Mol Cell Proteomics* **2007,** 6, (9), 1638-55.

Yen, C. Y.; Russell, S.; Mendoza, A. M.; Meyer-Arendt, K.; Sun, S.; Cios, K. J.; Ahn, N. G.; Resing, K. A., Improving sensitivity in shotgun proteomics using a peptide-centric database with reduced complexity: protease cleavage and SCX elution rules from data mining of MS/MS spectra. *Anal Chem* **2006,** 78, (4), 1071-84.

Dutta, D.; Chen, T., Speeding up tandem mass spectrometry database search: metric embeddings and fast near neighbor search. *Bioinformatics* **2007,** 23, (5), 612-8.

Roos, F. F.; Jacob, R.; Grossmann, J.; Fischer, B.; Buhmann, J. M.; Gruissem, W.; Baginsky, S.; Widmayer, P., PepSplice: cache-efficient search algorithms for comprehensive identification of tandem mass spectra. *Bioinformatics* **2007,** 23, (22), 3016-23.

Park, C. Y.; Klammer, A. A.; Kall, L.; MacCoss, M. J.; Noble, W. S., Rapid and accurate peptide identification from tandem mass spectra. *J Proteome Res* **2008,** 7, (7), 3022-7.

# Automatic Segmentation of Lip Images Based on Markov Random Field

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 a successful lip-reading system. We present a new segmentation approach to lip contour extraction by taking account of the maximum a posterior - Markov random field (MAP-MRF) framework. We first examine various color models and select a simple color transform derived from LUX and 1976 CIELAB color space as an effective descriptor to characterize the lip region by its discriminative properties. Thus, the initial label set with respect to lip and skin region is available. Based upon the identified lip area, we further refine the lip region using both color and label information, as those two are combined within a Markov random field (MRF) framework. 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.*

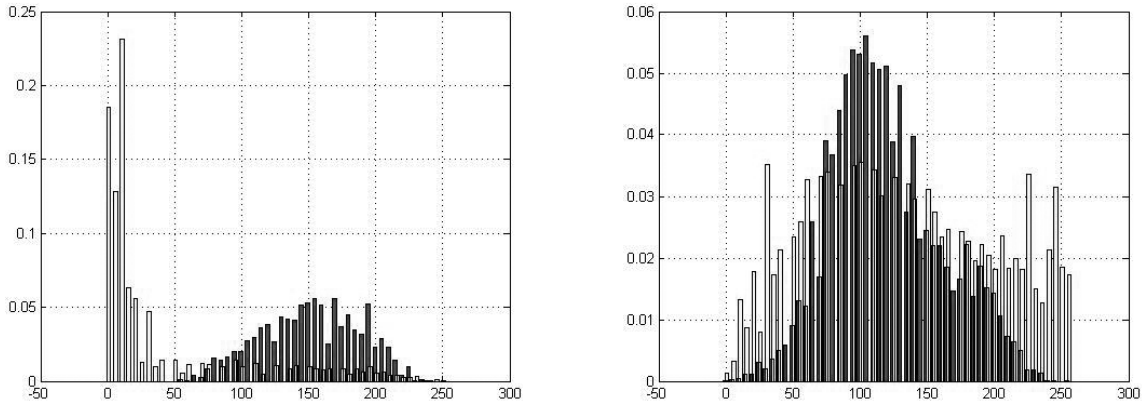
## 1. Introduction

Automatically segmenting out person's lip from face image is an active research area nowadays for its wide range of possible applications such as lip-reading for disabled people, audio-visual speech recognition in noisy environment, face detection, biometric person identification, lip synchronisation, facial expression recognition and so forth [1, 2, 3, 4, 5, 6].

Thus far, various segmentation techniques have been proposed. In general, these methods based on color space rather than gray level since color image can provide more useful clues. In [7, 8, 9, 10], they made an analysis of the original color space, and transformed its representation into a new space by intensity difference between lips and background. Clustering with color feature is an attractive preprocessing method as well. [11, 12, 13] utilize clustering based methods to conduct segmentation and achieve considerable high accuracy. Meanwhile, wavelet is another effective solution. [14] proposed a segmentation method which used wavelet multi-scale edge detection across the $C_3$ component of the discrete Hartley transform (DHT). Moreover, [10] employed Gaussian mixture model (GMM) to estimate the membership map of lips computed from the skin color distribution. Nevertheless, these methods only focus on color feature regardless of spatial characters which also convey important clues for segmentation. Thus, corresponding results seemed fragmented and so easily affected by noise, some of which are hard to overcome in postprocessing.

Nowadays, in image segmentation field, the assumption that "physical properties in a neighborhood of space present some coherence and generally do not change abruptly"[15] is utilized widely so as to overcome the segmentation errors arised from the intensity non-uniformity and local perturbations. In [16], a spatial fuzzy clustering algorithm was proposed. This method is able to take into account both the distributions of data in feature space (derived from 1976 CIELAB and 1976 CIELUV color space) and the spatial interactions between neighboring pixels during clustering. Furthermore, [5, 17, 18, 19] and so forth made advantage of the Markov random field (MRF) model to represent the spatial constraint.

Although empirical studies have shown their success, practical lip segmentation is a non-trivial task. The main difficulty lies in robustness and automation. In real world, the illumination condition and the complexion of speaker are multifarious. Thus, robustness is an important benchmark for a lip segmentation method. However, it is challenging for a color transform to achieve stable performance in different situations. It is because that the fluctuation range of hue of lip and skin region is considerable large under different illumination conditions, not to mention difficulties brought by testers (speakers) with totally different complexion. On the other hand, from the practical viewpoint, automation is also an important factor as well. Nevertheless, lots of common methods can not satisfy this requirement. For example, in the lip segmentation task, the fuzzy




(a) (b)

**Figure 1. The gray-level probability distribution of $I_{a^*}$ (dark color) and $I_U$ (light color) in (a) lip region, and (b) skin region. The x-axis represents the gray-level value, and the y-axis represents the probability.**

clustering based method [16] requires pre-assignment of the number of clusters, and the MRF based method [5] heuristically fixes the cluster number as 3. However, in application, the existence of moustache, the visibility of teeth and tongue generally require that the number of clusters is selected adaptively. Unfortunately, the performances of these methods depend on the knowledge of cluster number significantly.

In this paper, we will present a new method for the robust automatic segmentation of lip images provided that the part between nostril and chin has been available. Firstly, the proposed method employs a color transform derived from the LUX and 1976 CIELAB color space to obtain a lip segment via the distinction between the lip and skin. The result can be used to 1) initialize the MRF label map, and 2) estimate the parameters of likelihood energy function in MRF model. Secondly, a MRF model is established on the 4-neighborhood system. Thirdly, a deterministic algorithm called iterated conditional modes (ICM) is performed to minimize the cost function and obtain a robust labeling of lip and background, respectively. Finally, given the prior knowledge of human mouth shape, noise suppressing is performed based on morphological operation and taken as postprocessing procedure, together with the convex hull algorithm. 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. Section 2 describe the pre-processing (color transform), lip segmentation (MAP-MRF classification) and post-processing (lip boundary extraction) in turn. In Section 3, we will conduct the experiment to empirically compare our approaches with some existing methods. Finally, Section 4 draws a conclusion.

## 2. Method

### 2.1 Color Transform

It is desirable to work in a color space, in which the lip color (i.e. relative red) out of the others can be highlighted. Since the value of $a^*$ channel in 1976 CIELAB color space can determine the color component between magenta and green, i.e. the small values indicate green while the large indicate magenta. We therefore transform the source image into 1976 CIELAB color space and employ the histogram equalization[20] to map the $a^*$ component into the range of [0, 255], denoted as $I_{a^*}$. Furthermore,we utilize Eq.(1) as proposed in [5] to convert the source image to the range of $[0, 255]$ with equalization, denoted as $U$:

$$U = \begin{cases} 256 \times \frac{G}{R} \ if R > G \\ \\ 255 \qquad otherwise. \end{cases} \tag{1}$$

Then, we also map U component into the range of [0, 255] denoted as $I_U$.

We select 100 lip images from 4 databases randomly, label the lip region manually, and obtain $I_{a^*}$ and $I_U$ from each image. Then, calculate the histogram (normalized into [0, 1]) of data set composed by the in-

tensity of pixels belong to $I_{a^*}$ and $I_U$ which fall into lip and skin region, respectively (see Fig. 1). We further calculate the mean values of the four distribution denoted as $\mu_{lip}^{a^*}$, $\mu_{skin}^{a^*}$, $\mu_{lip}^{U}$ and $\mu_{skin}^{U}$. In this experiment, $\left|\mu_{lip}^{a^*} - \mu_{lip}^{U}\right| = 110.88$ is far more larger than $\left|\mu_{skin}^{a^*} - \mu_{skin}^{U}\right| = 15.64$.

Based on the analysis above, we believe that the difference between $I_{a^*}$ and $I_U$ in lip region is significant but inconspicuous in skin region. Thus, we can employ the following equation to get the lip segment roughly.

$$I_{sub} = I_{a^*} - I_U.^{1} \tag{2}$$

Subsequently, we establish a Gaussian model for $I_{sub}$ based on the gray-level value of each non-zero pixel with the mean $\hat{\mu}_{sub}$ and the standard deviation $\hat{\sigma}_{sub}$. The candidate lip segment can be obtained by

$$\tilde{I}^1_{candidate} = \begin{cases} 0 \ if \quad I_{sub} \leq \hat{\mu}_{sub} - 2\hat{\sigma}_{sub}, \\ \\ 1 \ otherwise. \end{cases} \tag{3}$$

Moreover, from a different aspect, we can calculate another representation of candidate lip segment. Firstly, the following equation is employed to get $U^{'}$:

$$U^{'} = \begin{cases} \frac{a^*}{G} \ if a^* > G \\ \\ 0 \quad otherwise. \end{cases} \tag{4}$$

where $a^*$ denotes the equalized $a^*$ component in 1976 CIELAB color space, and $G$ denotes the equalized $G$ component in RGB color space.

Then, a gray-level threshold selection method proposed in [21] is utilized to transform $U^{'}$ into a binary image denoted as $I^2_{candidate}$.

We assume the lip region is not connected to the border of input image. Thus, the morphological reconstruction based method proposed in [22] is performed to suppress noisy structures. For this operation, the mask is $\tilde{I}^1_{candidate}$, and the marker is an image which is all zero except along the border. The output image is denoted by $I^1_{candidate}$. Therefore, we use $I^1_{candidate} \cap I^2_{candidate}$ as lip segment, denoted by $I_{seg}$.

## 2.2 MAP-MRF Classification

In order to build a probability map of lip and skin region, each pixel $s$ will be attributed a label $l_s$ from the set $\Lambda$ reflects its feature class. In this paper, $\Lambda = \{0, 1\}$. For $s$ belong to lip class, $l_s = 1$, and $l_s = 0$ otherwise.

[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.

An realization of a set of labels is considered a configuration defined on a 2-D rectangular regular lattice $\mathscr{S} = \{i | 1 \leq i \leq N\}$ where $N$ is the number of pixels in the input image. An observed image in modified $HSV$ color space $c = \{c_i | i \in \mathscr{S}\}$, and a configuration $l = \{l_i | i \in \mathscr{S}\}$ are instances of each random field. The form of $c_i$ can refer to Eq. (5):

$$c_i = \{(H_i \cdot cos(2\pi \cdot S_i), H_i \cdot sin(2\pi \cdot S_i))^T | i \in \mathscr{S}\} \tag{5}$$

where $H_i$ and $S_i$ denote the $H$ and $S$ component value of pixel $i$.

A prior model should properly define the interactions between the labeled pixels. MRF are well suited for that purpose. Let us consider the spatial 4-neighborhood structure $\mathscr{N}_s$. The label field is supposed to verify the main MRF property related to that neighborhood, which means the label $l_s$ of the current pixel $s$ depends only on the labels $l_r$ of its neighbors $r \in \mathscr{N}_s$. Assuming our scene is a piecewise constant surface and the model is spatial homogeneous, the prior probability can be written as follows with the independent assumption in terms of the MRF-Gibbs equivalence:

$$p(l) = \prod_{i \in \mathscr{S}} \frac{e^{-V(i)}}{\sum_{j \in \mathscr{S}} e^{-V(j)}} \tag{6}$$

Based on Potts model, the prior energy can be defined as:

$$V(l) = \sum_{i \in \mathscr{S}} V(i) = \sum_{i \in \mathscr{S}} \sum_{i' \in \mathscr{N}(i)} (1 - \delta(l_i - l_{i'})) \tag{7}$$

where $\delta(\cdot)$ is the Kronecker delta function.

To establish the likelihood energy function, we further assume that the intensity $c_i$ for pixels with the same label follows the same bivariate Gaussian distribution. The likelihood probability can be written as:

$$p(c|l) = \prod_{i \in \mathscr{S}} \frac{1}{2\pi\sqrt{|\hat{\Sigma}^{l_i}|}} \cdot exp(-\frac{(c_i - \hat{\mu}^{l_i})(c_i - \hat{\mu}^{l_i})^T}{2\hat{\Sigma}^{l_i}}) \tag{8}$$

Given the label $l_i = \lambda \in \Lambda$, parameter estimation is made as follows:

$$\hat{\mu}^{\lambda} = \frac{\sum_{j=1}^{M^{\lambda}} c_j^{\lambda}}{M^{\lambda}}, \tag{9}$$

$$\hat{\Sigma}^{\lambda} = \frac{1}{M^{\lambda} - 1} \sum_{j=1}^{M^{\lambda}} (c_j^{\lambda} - \hat{\mu}^{\lambda})(c_j^{\lambda} - \hat{\mu}^{\lambda})^T, \tag{10}$$

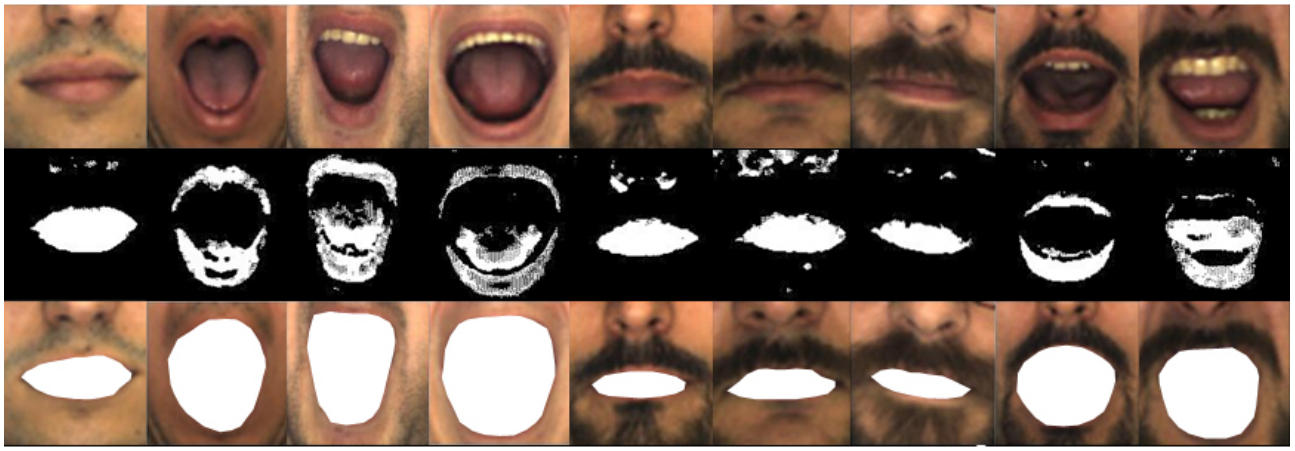


**Figure 2. Top row: some examples of source input images, in which several representative samples – normal situation, mouth opening (teeth existed), mustache existed, etc., are involved. Middle row: corresponding segmentation results obtained by ICM. Bottom row: final segmentation results.**

where $c_j^\lambda$ is obtained by Equ. (5) at the $j$th pixel with label $\lambda$ in the input image, and $M^\lambda$ denotes the number of these pixels.

Thus, the likelihood energy can be defined as:

$$V(c|l) = \sum_{i \in \mathscr{S}} (c_i - \hat{\mu}^{l_i}) {\Sigma^{l_i}}^{-1} (c_i - \hat{\mu}^{l_i})^T \qquad (11)$$

Based on MAP framework, the label can be selected for each pixel through the following optimal function:

$$l = \arg\max_{l_i \in \Lambda} p(l)p(c|l) = \arg\min_{l_i \in \Lambda} (\alpha V(l) + V(c|l)) \qquad (12)$$

where $\alpha$ is a positive weight which can be used to balance the dimensions of the two terms, say, $V(l)$ and $V(c|l)$. And the segmentation result won't be sensitive to this value. In our experiments, $\alpha$ was set to 2.

We choose the iterative deterministic algorithm ICM (Iterated Conditional Modes) to compute the minimum energy at each site for sake of its low computation cost. But the problem is it may converge towards local minima. In our experiments, however, a stable solution is always found in practice after a few iterations on the field (less than 5). The initial label set is derived from $I_{seg}$. The relative variation of global energy is used as termination condition: $\Delta E(l)/E(l) < \epsilon$ (typically, $\epsilon = 0.05$) where $E(l) = \alpha V(l) + V(c|l)$. Some segmentation results obtained by ICM can be found in middle row, Figure. 2.

### 2.3 Boundary Extraction

The result obtained by ICM can be considered as a binary image (the pixels with label 1 are foreground, and the others are background). Then, we suppress the boundary connected structures [22] in it and denoted as $B_{RT}$. The biggest continued foreground block is marked 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 connected block. Thus, some refinements are needed.

Considering the primary reason for disconnection between upper and lower lip is that the teeth and tongue are eliminated in the above steps. Hence, we utilize the following equation

$$I_{TTM} = I_U - I_{a^*} \qquad (13)$$

to obtain the region covering the teeth, tongue and some parts of oral cavity approximately.

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 $5 \times 5$ 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 utilize the morphological opening with $3 \times 3$ structur-

ing element so as to make the edge more smooth. The result is denoted as $B_{lip}$.

Finally, the quickhull algorithm proposed in [23] is employed to draw the contour of lip (e.g. see bottom row, Figure 2).

The proposed segmentation method can be summarized as follows:

---

input the RGB source image;

compute the binary $I_{seg}$ as the initial segmentation;

initialize the number of iteration $t = 0$;

update $l^t = \{l_i | i \in \mathscr{S}\}$;

do
{
update the variables: $\hat{\mu}^\lambda, \hat{\Sigma}^\lambda$;

get a label set which can reach the minimum of Equ.(12) via ICM;

perform the morphological filters and convex hull method to refine the label set;

update $l^{t+1} = \{l_i | i \in \mathscr{S}\}$ by the label set;

$t = t + 1$;

}until($\frac{d(l^t, l^{t-1})}{size(l^t)} \leq \xi$ )

output the final result;

---

where $d(l^t, l^{t-1})$ denotes the Hamming distance between the label set obtained by the $t$th and $(t-1)$th iteration, respectively; $size(l)$ denotes the number of elements belong to set $l$; $\xi$ is termination condition which can get 0.005 heuristically.

## 3. Experiment Results

To demonstrate the performance of the proposed approach in comparison with the existing methods denoted as: Liew03 proposed in [16], and Lievin04 in [5]. We utilized four databases to test the accuracy and robustness in different capture environments: (1) AR face database (126 people with 26 images for each) [24], (2) CVL face database (114 persons with 7 images for each) [25], (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 800 images in total (400 images from AR database, 200 images from CVL database, 100 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) were not used for this experiment.

| Algorithm | Liew03 | Lievin04 | Proposed |
|---|---|---|---|
| average OL, % | 78.73 | 87.46 | **92.48** |
| average SE, % | 35.15 | 25.01 | **7.10** |

**Table 1. The segmentation results across the four databases.**

Two measures defined in [16] 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 $A_2$:

$$OL = \frac{2(A_1 \cap A_2)}{A_1 + A_2} \times 100\%. \tag{14}$$

The second measure is the segmentation error ($SE$) defined as

$$SE = \frac{OLE + ILE}{2 \times TL} \times 100\%, \tag{15}$$

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 Lievin04 in both of the two measurements. Specifically, for the embarrassed problem in lip segmentation – the moustache existed cases, say, the tester in AR database with number 4, 5, 18, 26, 31, 38 and so forth, the $OL$ and $SE$ our method proposed is 91.15% and 10.14%, respectively.

## 4. Conclusion

In this paper, we have proposed a new approach to automatic lip segmentation via color transform and the MAP-MRF framwork. This approach features the high accuracy of lip segmentation and robust performance against diverse 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] T. Chen and R.R. Rao. Audio-visual integration in multimodal communication. *Proceedings of the IEEE*, 86(5):837–851, 1998.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] H.E. Cetingul, Y. Yemez, E. Erzin, and A.M. Tekalp. Discriminative analysis of lip motion features for speaker identification and speech-reading. *IEEE Transaction on Image Processing*, 15(10):2879–2891, 2006.

[7] 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.

[8] T. Wark, S. Sridharan, and V. Chandran. An approach to statistical lip modelling for speaker identification via chromatic feature extraction. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 123–125, Brisbane, Australia, 1998.

[9] N. Eveno, A. Caplier, and P.-Y. Coulon. A parametric model for realistic lip segmentation. In *Proceedings of International Conference on Control, Automation, Robotics and Vision*, pages 1426–1431, Singapore, 2002.

[10] C. Bouvier, P.-Y. Coulon, and X. Maldague. Unsupervised lips segmentation based on roi optimisation and parametric model. In *Proceedings of IEEE International Conference on Image Processing*, pages 301–304, San Antonio, USA, 2007.

[11] M. SADEGHI, J. KITTLER, and K. MESSER. Spatial clustering of pixels in the mouth area of face images. In *Proceedings of IEEE Internatioal Conference on Image Analysis and Processing*, pages 36–41, Palermo, Italy, 2001.

[12] S.L. Wang, W.H. Lau, S.H. Leung, and A.W.C. Liew. Lip segmentation with the presence of beards. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 529–532, 2004.

[13] 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.

[14] Y.P. Guan. Automatic extraction of lips based on multi-scale wavelet edge detection. *Computer Vision, IET*, 2(1):23–33, 2008.

[15] S.Z. Li. *Markov Random Field Modeling in Image Analysis (Third Edition)*. Springer, 2009.

[16] 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.

[17] Albert C.S. Chung. A segmentation model using compound markov random fields based on a boundary model. *IEEE Transactions on Image Processing*, (1):241–252, 2007.

[18] H. Gribben, P. Miller, G.G Hanna, K.J Carson, and A.R. Hounsell. Map-mrf segmentation of lung tumours in pet/ct images. In *Proceedings of 6th International Symposium on Biomedical Imaging: From Nano to Macro*, pages 290 – 293, Boston, USA, 2009.

[19] B. Scherrer, F. Forbes, C. Garbay, and M. Dojat. Distributed local mrf models for tissue and structure brain segmentation. *IEEE Transactions on Medical Imaging*, (8):1278–1295, 2009.

[20] C. Gonzalez. *Digital Image Processing (Third Edition)*. Pearson Eduation Eduaction, Inc., 2008.

[21] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[22] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 1999.

[23] 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.

[24] A.M. Martinez and R. Benavente. The ar face database. *CVC Technical Report No.24*, June 1998.

[25] 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.

# Robustness and Entropy for Directed Networks without Loops

Benyun Shi

## Abstract

*Robustness is one of the fundamental characteristics that allows a system to maintain its performance against internal and external perturbations and uncertainty. The components of a system can be represented as a network, where the structure (i.e., connectivity) represents the inter-relationship of the system components. The interactions among system components (e.g., cell-cell signal transduction) further control the performance of the network. In this paper, we consider two kinds of network robustness: (i) the structural robustness, which focuses on network connectivity, and (ii) the functional robustness, which focuses on network performance. Most existing studies about complex networks have focused on analyzing structural robustness of undirected networks. In this paper, we present a general entropic measurement (i.e., network entropy) for both structural and functional robustness of directed networks without loops. Theoretical analysis and empirical simulations have validated that the network entropy is positively correlated with network robustness. Finally, we outline some potential implementations of the defined network entropy in terms of resource distribution problems.*

## 1 Introduction

Robustness is the ability of a system to maintain its performance in the face of internal/external perturbations and uncertainty. The performance of a system is often associated with its functionality, such as protein metabolism in organisms. Many systems can be modeled as networks, where each node may represent a system component, and each edge may represent inter-relationship between two components. For example, in ecological networks [22], different species form a food web through predator-prey interactions. In biological network [11], many cellular components interact (e.g., activate or inhibit) with each other to form signal transduction pathways. By doing so, it is helpful for us to understand the robustness of a complex living system (e.g., Th cell differentiation [11]) by studying the properties of the modeled network.

Existing studies on complex networks [4] have shown that there are two kinds of network robustness, i.e., *structural robustness* and *functional robustness*. The structural robustness focuses on the ability of the network to maintain its connectivity properties under random failure or intentional attack [4]. For example, one may interest in the size of the giant component as a function of the fraction of nodes or edges that are deleted from a network [1]. The functional robustness focuses on the performance of the network under uncertainty. For example, the removal of nodes in a power transmission grid, either by random breakdown or intentional attacks, changes the balance of power flows and leads to a global redistribution of loads over all the grid, which might trigger a cascade of overload failures [20]. The main difference between structural and functional robustness is that when node/edge failure happens, structural robustness does not consider redistributing any quantity that is being transported by the network, while functional robustness must take into account the redistribution dynamics.

Although various studies ([1][5][6][8][12][18], just list a few) in complex networks field have been done both theoretically and empirically for analyzing network robustness, most of them focus mainly on undirected networks. Recent studies [11][15][16] in system biology have shown that biological systems can be modeled as directed networks. Some of them have quantitatively analyzed the relationship between functional robustness of biological networks and the number of feedback loops on the networks. Different with previous studies, in this paper, we will study the robustness (i.e., both structural and functional) of resource distribution systems. The resource distribution systems, such as natural gas distribution systems on pipeline networks, urban water distribution systems, can be modeled as directed networks, where each node represents a resource supplier or consumer, and each direct edge represents resource flows from one node to another. One important feature of this kind of networks is that there is no distribution loops, i.e., resources cannot flow back to its origin.

Most existing studies on system robustness analysis rely on either simulations (e.g., in the field of complex networks) or experiments (e.g., in the field of system biology). However, for some real-world systems (e.g., power grids), to test the system robustness is expensive, and hence is intractable. As argued by H. Kitano [15] that, to understand the robust-

ness at the system level, we still need a solid theoretical and methodological foundation. In this case, it will be extremely significant to present a general measurement that could identify both structural and functional robustness of a system. In this paper, we will present an entropic measurement for identifying the robustness of resource distribution systems. Given a distribution strategy for a specific distribution system, as the system evolving, the quantity of resources at each node will become relatively constant. In this case, we say that the system reaches a *stable* state. Under perturbation and uncertainty, the resources will be *redistributed* based on distribution strategies. We will focus on the ability of such a system to maintain its stable state under two kinds of perturbations: (i) structural perturbations, i.e., random and intentional node deletion, and (ii) functional perturbations, i.e., disturbance in resource supply from certain nodes.

The main objective of this paper is to identify the relationship between network robustness (both structural and functional) and network entropy (defined in Section 3). By both theoretical analysis and empirical simulation, we show that the network entropy is positively correlated with network robustness under both structural and functional perturbations. By treating network entropy as an indicator of network robustness, we further outline the following two questions for future studies: (i) how to find the critical components that are essential for the robustness of a system, and (ii) how to build a robust resource distribution system. This work extends the state of the art in the following ways:

1. Different with previous studies focusing on robustness of undirected networks, in this paper, we study the robustness of resource distribution systems, which can be modeled as directed networks without loops;

2. We present an entropic robustness measurement for resource distribution systems, which can reflect both structural and functional robustness of the systems;

3. We verify the positively correlation between network entropy and robustness (i.e., structural and functional robustness) by both theoretical analysis and empirical simulations.

The rest of this paper is organized as follows: In Section 2, we summarize the related work about network robustness for both undirected and directed networks; In Section 3, we present the network entropy for resource distribution networks, i.e., directed networks without loops; In Section 4, we define the robustness that we consider in this work; In Section 5, we theoretically analyze the relationship between network robustness and entropy; Empirical simulations in Section 6 validate the structural and functional robustness with respect to network entropy respectively; We conclude our work and outline some questions for future studies in Section 7.

## 2 Related Work

Network robustness has been well studied in complex network field, where most of them focus mainly on undirected networks. However, in reality, most living systems (e.g., food webs or metabolism systems) and artificial systems (e.g., resource distribution systems) can be modeled as directed networks, where each node in such networks plays as different roles, such as the predators and preys in food webs, activator and inhibitor in biological networks and resource suppliers and consumers in resource distribution networks. There are two kinds of network robustness, i.e., structural and functional robustness. The former focuses on the connectivity of the network, while the latter focuses on the functionality of the network. The network robustness studies on directed and undirected networks have different focuses, in this section, we will summarize the related work in the two research domains.

### 2.1 Network Robustness of Undirected Networks

For the last decade, researchers in statistical physics have proposed various approaches to study network tolerance to errors and attacks both by numerical simulations [5][1] and theoretical approaches [6][8]. The structural robustness to errors (or random failure) is the ability of the system to maintain its connectivity after the random deletion of a fraction $f$ of its nodes or edges. While an attack means that the deletion process is targeted to a class of particular nodes or edges. Once nodes or edges are deleted, several quantities allow the characterization of the network properties, such as the size of the giant components, centralities (e.g., betweenness centralities). The focus of statistical physics is to study the critical value $f_c$, above which a set of critical exponents characterizing the phase transition. Various work has been done to study the structural robustness of undirected networks with different properties, such as correlated networks [23], uncorrelated networks [8], random and scale-free networks [1]. For example, Albert et al. in [1] have anticipated by simulations that targeted deletion of nodes in uncorrelated scale-free networks are highly effective compared with random deletion.

Motivated by several blackout accidents (e.g., the blackout of 11 US states in 1996 [20]), functional robustness of networks has drawn extensive attentions. The functional robustness takes into account the redistribution of quantity that is being transported on the network. The studies on functional robustness focus on modeling the cascading failures [18] or congestion in communication networks [12] to try to understand the fundamental reasons or critical issues

that result in the accidents. Similarly, models for networks with different topological properties have been proposed. To have a more detailed survey about robustness of undirected networks, we refer to S. Boccaletti et al.'s work [4].

### 2.2 Network Robustness of Directed Networks

Recent studies in system biology [11][15][16] have shown that biological systems can be modeled as directed networks, where every node on the directed network may play different roles (e.g., activator or inhibitors in Th cell [11]). The dynamics on the networks are usually modeled by differential equations [11][9] or simulated by cellular automata (CA) [16]. Existing studies of robustness in biological systems focus mainly on the ability of of the systems to maintain its stable states under perturbations. Some of them [16][11] concentrate on investigating the role of feedback loops (i.e., a circular chain of interactions) in realizing the system robustness. For instance, the authors in [16] have found that networks with a larger number of positive feedback loops and a smaller number of negative feedback loops are likely to be more robustness against perturbations. However, a solid theoretical and methodological foundation, especially, a general robustness measurement, to study robustness of directed networks is still lacking. Different with existing work in system biology, in this paper, we focus on the robustness of resource distribution systems, i.e., the distribution dynamics on directed networks without loops.

### 2.3 Network Robustness and Entropy

Existing work shows two ways to define network entropy. Firstly, network entropy can be defined by structural observable (e.g., degree distribution [24], remaining degree distribution [21][2]). Secondly, the network entropy can also be derived from systematic approaches in the context of ergodic theory of dynamic systems [10]. Network entropy defined by degree distribution or remaining degree distribution has been validated to highly correlate with network heterogeneity, and further be analyzed to be robust against random failures. For example, in [21], the authors have defined the network entropy based on remaining degree distribution for undirected networks, they further validate by real-world networks that heterogeneous networks (e.g., scale-free networks) have higher entropy value than regular networks. For functional network entropy, Demetriusa et al. in [9][10] have adopted the Kolmogorov-Sinai (KS) entropy to define dynamical entropy for population dynamics on the life cycle network (i.e., directed networks), where the age-structured model is used to simulate the population dynamics. They further analyze that the KS entropy is positively correlated with network robustness of the life cycle system. With respect to resource distribution, so far as we know, very few studies have been focused on the robustness analysis for resource distribution on the networks. In this paper, we define an entropic measurement for measuring the robustness of resource distribution systems.

## 3 Network Entropy

The resource distribution systems can be modeled as directed networks without loops. Given a directed network $G(V, E, W)$ with $N$ nodes and $M$ edges to represent a distribution system, where $v_i(t) \in V$ represents the quantity of resources that flow through node $i$ at time $t$. Each directed edge $e_{ij} \in E$ represents the connectivity of nodes $i$ and $j$. If there are resources flowing from $i$ to $j$, then $e_{ij} = 1$; otherwise, $e_{ij} = 0$. Each edge is associated with a weight $w_{ij} \in [0, 1]$, where the weight set $S_i = \{w_{ij} | e_{ij} = 1, \sum_j w_{ij} = 1\}$ represents the distribution dynamics of $i$ that describe how the resources are distributed from $i$ to $\{j | e_{ij} = 1\}$. For instance, if $e_{ij} = 1$, $i$ will distribute $v_i \cdot w_{ij}$ quantity of resources to $j$. In this case, the value $v_i$ of each node will also be influenced by the status of its parents, i.e., $\{v_j | e_{ij} = 1\}$. In this section, we will present an entropic measurement for robustness of directed networks, which takes into account both network connectivity and dynamics on the networks.

### 3.1 An Entropic Measurement

The topological structure of network $G(V, E, W)$ can be described by an $N \times N$ adjacency matrix $A = \{e_{ij}\}_{N \times N}$, which is directed and Boolean, i.e., $e_{ij} \in \{0, 1\}$ represents whether there is flows from node $i$ to node $j$.

With respect to a resource distribution network, each node of the network is associated with a quantity of supply or demand, i.e., each node may represent a resource supplier, or an intermediator, or a consumer. For instance, in Figure 1, nodes 1 and 2 represent resource suppliers; nodes 3, 4 and 5 represent resource consumers; at the same time, nodes 3 and 4 also play as distribution intermediators. The distribution system is open because resources will be continuously supplied by suppliers and consumed by consumers. To facilitate analysis, we add a virtual node 0 to make the system close (see Fig. 1 for example), where all resources will be assumed to be supplied by 0, and consumed resources will flow back to 0 after being consumed by corresponding consumers (node 3, 4 and 5 in Fig. 1).

Each node $i$ ($i \neq 0$) in the newly formed network is associated with a quantity of inflows $I_i$ and outflows $O_i$, where $I_i = O_i$. For the virtual node 0, we have $I_0 = \sum_{e_{j0}=1} O_j$ and $O_0 = \sum_{e_{0k}=1} I_k$. In this paper, we treat the flows on the network as a stochastic process $P$, where $p_{ij} \in P$ represents the probability that a unit of flow which enters $i$ is distributed along edge $ij$ to $j$. By the definition of

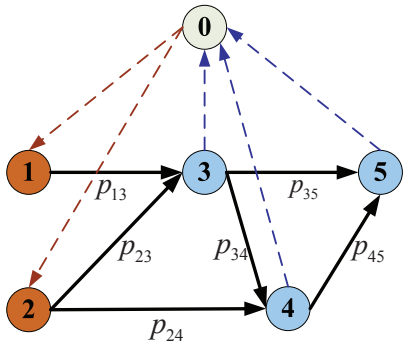



**Figure 1.** An example of energy flow network. Nodes 1 and 2 represent resource suppliers, and nodes 3, 4 and 5 represent resource consumers. A virtual node 0 is added to facilitate the analysis such that all resources is assumed to be initially supplied (respectively, finally consumed) by 0.

$G(V, E, W)$, we have $p_{ij} = w_{ij}$. Note that $p_{ij} = 0$ if and only if $e_{ij} = 0$. In this case, the expected flows from $i$ to $j$ will be $I_i p_{ij}$. We denote stationary state $\pi_i(t)$ the quantity of resources that flow through $i$ at time $t$, then we have

$$\pi_i(t) = \sum_{k=0}^{N} \pi_k(t) \cdot p_{ki} \tag{1}$$

It can be observed that the stationary state of each node is interdependent, that is, the value $\pi_i(t)$ is determined by stationary values of its parents, i.e., $\{k|e_{ki} = 1\}$. Furthermore, the stationary state distribution $\Pi$ also depends on the stochastic process $P$.

The author in [14] has given a form of Shannon entropy for mutually dependent probability distributions: for two mutually dependent probability distribution $Q$ and $R$, the entropy of the joint distribution $Q * R$ can be given by

$$H(Q * R) = H(Q) + H(R|Q) \tag{2}$$

Based on the result, we define the entropy of a resource distribution networks, i.e., a directed network without loops as follows:

$$H(\Pi * P) = H(\Pi) + H(P|\Pi) \tag{3}$$

Since the resource supply of each supplier is known in advance, for the energy suppliers $i$ (where $e_{0i} = 1$), we define its stationary state $\pi_i = \frac{I_i}{O_0} = \frac{I_i}{\sum_{e_{0j}=1} I_j}$. Given the Markov matrix $P$, the stationary states of other nodes can be calculated iteratively by Equation 1. Hence, we define the network entropy as

$$H(\Pi) = -\sum_{e_{0i}=1} \pi_i \log \pi_i \tag{4}$$

and

$$H(P|\Pi) = -\sum_{i,j=1}^{N} \pi_i p_{ij} \log p_{ij} \tag{5}$$

For nodes $i|e_{0i} = 1$ (i.e., suppliers), the value $\pi_i$ is predefined, in this case, the value of Equation 4 is constant. Therefore, the network entropy $H(\Pi * P)$ is mainly determined by $H(P|\Pi)$ in quation 5, i.e., the entropy of the Markov process on the network. That means the network entropy is also determined by both the network connectivity and the distribution dynamics on the network.

### 3.2 An Example

In the following, we take the network illustrated in Figure 1 as an example to show how to calculate the conditional entropy defined in Equation 5. Note that nodes 1 and 2 in Fig. 1 represent resources suppliers, and nodes 3, 4 and 5 represent resource consumers. A virtual node 0 is added to facilitate the analysis such that all resources is initially supplied (respectively, finally consumed) by the node 0. The adjacency matrix $A$ of the network is

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Suppose that supply of 1 and 2 is respectively $I_1$ and $I_2$. We have the stationary states $\pi_1 = \frac{I_1}{I_1+I_2}$ and $\pi_2 = \frac{I_2}{I_1+I_2}$. Based on Equation 1, the stationary states of node 3, 4 and 5 can be calculated as follows:

$$\begin{aligned} \pi_3 &= \pi_1 p_{13} + \pi_2 p_{23} \\ \pi_4 &= \pi_2 p_{24} + \pi_3 p_{34} \\ &= \pi_2 p_{24} + \pi_1 p_{13} p_{34} + \pi_2 p_{23} p_{34} \\ \pi_5 &= \pi_3 p_{35} + \pi_4 p_{45} \\ &= \pi_1 p_{13} p_{35} + \pi_2 p_{23} p_{35} + \pi_2 p_{24} p_{45} + \pi_1 p_{13} p_{34} p_{45} \\ &\quad + \pi_2 p_{23} p_{34} p_{45} \end{aligned}$$

In this case, the entropy of each node can be calculated as follows:

$$\begin{aligned} H_1 &= -\pi_1 p_{13} \log p_{13} \\ H_2 &= -\pi_2 p_{23} \log p_{23} - \pi_2 p_{24} \log p_{24} \\ H_3 &= -\pi_3 p_{30} \log p_{30} - \pi_3 p_{34} \log p_{34} - \pi_3 p_{35} \log p_{35} \\ H_4 &= -\pi_4 p_{40} \log p_{40} - \pi_4 p_{45} \log p_{45} \\ H_5 &= -\pi_5 p_{50} \log p_{50} \end{aligned}$$

Hence, the conditional entropy (Equation 5) for the Markov process on the network is determined, i.e., $H(P|\Pi) = \sum_{i=1}^{5} H_i$.

## 4 Network Robustness

Robustness reflects the ability of a system to maintain its performance under internal/external perturbations. For the resource distribution networks, in this paper, we consider two kinds of robustness: structural and functional robustness.

The structural robustness focuses on the ability of a network to maintain its connectivity in the face of structural perturbations, where resources redistribution will not be taken into account. We focus mainly on two types of structural perturbations for structural robustness, i.e., random error and intentional attack [4]. For a random error on network $G$, it means that a node of $G$ will be deleted with probability $1/N$, where $N$ is the total number of nodes in $G$. With the help of dominator trees, we define error sensitivity ($ES$, Equation 10 in Section 6.1.2) to measure the structural robustness, which considers the expected number of nodes (including the deleted one) that will loss resource supply under a random error. For an intentional attack, it means that the most important node will be deleted form $G$. We define attack sensitivity (AS, Equation 11 in Section 6.1.2) to measure the structural robustness of $G$ under intentional attack.

The functional robustness focuses on the ability of a distribution system to satisfy resource demand of consumers in the face of supply perturbations from resource suppliers. When the resource supply of a supplier node is perturbed, resources flows along its outgoing edges will be redistributed based on corresponding edge weights. During redistribution, the distribution strategy of each node $i$, i.e., the value of weight set $S_i$, will keep unchanged (however, the quantity of resources flowing through each edge may change). The functional robustness in this paper is measured by the variation of satisfactions of resource consumers in the system. The detailed description will be given in Section 6.2.

## 5 Theoretical Analysis

In this section, we will analyze the relationship between network robustness and entropy defined in Sections 3 and 4. The analysis is mainly based on existing work [9][10]. Demetriusa et al. in [9] have firstly adopted the Kolmogorov-Sinai (KS) entropy to characterize the macroscopic properties (i.e., functional robustness) of life cycle graph [9] (i.e., network) based on ergodic theory and statistical mechanics. They use age-structured model [1] to model the population dynamics on the life cycle network. They further verify in [10] that the KS entropy is also positively correlated with structural robustness of undirected networks (e.g., regular, random and scale-free networks). The major differences between Demetriusa et al.'s work and ours is that (i) a life cycle network is directed network with multiple loops, while in this paper, we consider directed networks without loops, (ii) we consider the resource distribution dynamics other than population dynamics on the networks, and (iii) the age-structured model is an abstraction model for population dynamics, the network entropy defined in this paper is directly based on physical meaning (i.e., distribution strategies) of the network.

KS-entropy describes the time-ordered sequences of nodes by an assumed Markov process on a network, which is represented by a Markov matrix $P = \{p_{ij} | p_{ij} \geq 0, \sum_j p_{ij} = 1\}$. The dynamical entropy of the stochastic process is defined as

$$H(P) = \sum_{i=1}^{N} \pi_i H_i, \quad \texttt{where} \quad H_i = -\sum_j p_{ij} \log p_{ij} \quad (6)$$

Here, $H_i$ is the Shannon entropy of the distribution $[p_{i1}, \cdots p_{iN}]$, and $\pi_i \in \Pi$ is the stationary state of $i$.

The network entropy is the entropy of the Markov process associated with the adjacency matrix $A = \{e_{ij}\}_{N \times N}$ of the netwrok. In this case, for the Markov matrix, they have $p_{ij} = 0$ if and only if $e_{ij} = 0$. Denote $\lambda$ the dominant eigenvalue of adjacency matrix $A$. The authors in [3] have proved that $\log \lambda$ satisfies a variational principle, where

$$\begin{aligned} \log \lambda = \quad & \sup_{P \in M_A} \Big\{ - \textstyle\sum_i \pi_i \sum_j p_{ij} \log p_{ij} \\ & + \textstyle\sum_i \pi_i \sum_j p_{ij} \log a_{ij} \Big\} \end{aligned} \quad (7)$$

where $M_A$ is the set of all possible Markov processes associate with adjacency matrix $A$. In terms of this paper, $M_A$ corresponds to the set of all possible distribution strategies associated with a network.

The important feature of KS entropy defined in [9][10] is that the stationary distribution $\Pi$ , which characterizes the long time invariant behavior of Markov process $P$, satisfies

$$\Pi P = \Pi \quad (8)$$

In terms of this paper, the stationary distribution defined in Equation 1 also satisfy the condition of Equation 8.

**Lemma 5.1** *The stationary states defined in Equation 1 satisfy* $\Pi P = \Pi$ *for any Markov process defined in Section 3.*

The authors in [9] have derived a fluctuation theorem to invoke the network entropy defined by Equation 6 as a measure of network robustness (measured by fluctuation decay rate $R = \lim_{t \to \infty} [-\frac{1}{t} \log P_\epsilon(t)]$, where $P_\epsilon(t)$ denote the probability that the sample mean deviates by more than $\epsilon$ from its unperturbed value at time $t$. We will give

a specific definition in Section 6.2.). The fluctuation theorem shows that the changes in fluctuation decay rate are positively correlated with changes in network entropy (i.e., $\triangle H \triangle R > 0$), which means that "an increase in entropy entails a great insensitivity of an observable to dynamic or structural perturbation of the network" [17].

In Section 3, similar with Demetriusa et al.'s work, we have already derived a Markov process associated with the network edge wights and the adjacency matrix $A$ (i.e., $e_{ij} = 0 \Leftrightarrow p_{ij} = 0$) on directed networks without loops. The dynamical entropy $H(P|\Pi)$ is similarly defined by mutually dependent stationary states $\Pi$ and Markov matrix $P$. According to the fluctuation theorem in [9], we can conclude that the dynamical entropy $H(P|\Pi)$ defined in this paper is positively correlated with the network entropy (measured by fluctuation decay rate). In this case, give a network adjacency matrix $A$, to form a robust resource distribution network, we only need to find the supremum over all possible Markov matrix $P \in M_A$ in Equation 7, which corresponds to a robust resource distribution strategy.

# 6 Simulation

In this section, we will do simulation-based experiments to study the relationship between network robustness and the entropy proposed in Section 3. The simulation objective is to validate the following two hypotheses.

**Hypothesis 6.1** *The network entropy defined in Section 3 has a strong positive correlation with structural robustness of directed networks without loops in the face of random error defined in Section 4.*

**Hypothesis 6.2** *The network entropy defined in Section 3 has a positive correlation with functional robustness of directed networks without loops in the face of supply perturbations defined in Section 6.2.*

The simulations for structural and functional robustness will be done separately in this section because if we can validate the above two hypotheses separately, we can conclude that the network entropy is positively correlated with both structural and functional robustness.

## 6.1 Structural Robustness

To validate the Hypothesis 6.1, in this section, we will calculate the robustness and entropy values for 100 randomly generated directed networks, each of which have the same number of nodes and edges but different connectivity. There are three important issues that need to be addressed:

- How to generate directed networks without loops? Existing studies in complex networks focus on generating undirected networks with different topologies (e.g., random, small-world and scale-free networks). To generate directed networks without loops, we will introduce the cascade model [7] in Section 6.1.1, which is extensively used to model ecological systems [13];
- How to measure the structural robustness under random error and intentional attack? In Section 6.1.2, we will introduce two kinds of measurements, i.e., error sensitivity and attack sensitivity, with the help of dominator trees;
- How to calculate network entropy? Because structural robustness does not consider resource redistribution after perturbations, in this paper, we assume all outgoing edges of each node $i$ have the same weight.

### 6.1.1 Directed Network without Loops

To generate directed networks without loops, we borrow ideas from the cascade model [7], which explains some important properties of natural webs, such as the number of species within a trophic level. The basic idea is that the trophic level structure is taken into account, where every node $i = 1, 2, \cdots, n$ is assigned by a random $l_i$ number distributed evenly within the unit interval. Nodes are ordered according to $l_i$. If $l_j < l_i$ then $i$ has a probability of zero to link to $j$.

Algorithm 1 shows the generating procedure of directed networks without loops. We first make sure every node is connected with at least one other node in the network. Then, we randomly generate the other $M - N$ directed edges. For the simulation, we set network size $N = 1000$ because in reality, the distribution network may have a scale about 1000 nodes at a country level [19]. Essentially, we have found that it is not the network size rather than the ratio of the number of edges and nodes (i.e., $M/N$), which significantly affect the network robustness. In this work, we will do simulations for networks with $M/N = 1.5$ and 1.86 respectively, where $M/N = 1.86$ is the link-species scaling value of food webs [7].

### 6.1.2 Measurements for Structural Robustness

In this section, with the help of dominator trees [22], we define two structural measurements, i.e., $ES$ and $AS$, of network robustness with respect to random error and intentional attack. In a directed network $G$, there are certain routes are obligatory when there are no alternative routes from one node to another. A dominator tree is the topological structure that groups all these obligatory routes. Any node on the tree is said to be dominated by its ancestors. A simple example of a simple flow network and its dominator tree is shown in Figure 2. In the left-side figure, node $e$ is dominated by node $d$ ($d = dom(e)$), then in the dominator tree of right-side figure, $d$ is an ancestor of $e$. If

**Input**: Network Size: $N$, Edge/Node Ratio: $M/N$
**Output**: Adjacent Matrix: $A$

Assign a random value $l_i \in [0, 1]$ for each node $i$;
**foreach** $i$ **do**
  Randomly select a node $j \neq i$ ;
  **if** $l_j < l_i$ **then**
    $e_{ji} = 1$ ;
  **else**
    $e_{ij} = 1$ ;
  **end**
**end**
**while** *Number of edges* $< M$ **do**
  Randomly select a pair of nodes $j$ and $k$;
  **if** $e_{jk} \neq 1$ && $e_{kj} \neq 1$ **then**
    **if** $l_j < l_k$ **then**
      $e_{jk} = 1$ ;
    **else**
      $e_{kj} = 1$ ;
    **end**
  **end**
**end**

**Algorithm 1**: The algorithm to generate a directed network without loops.

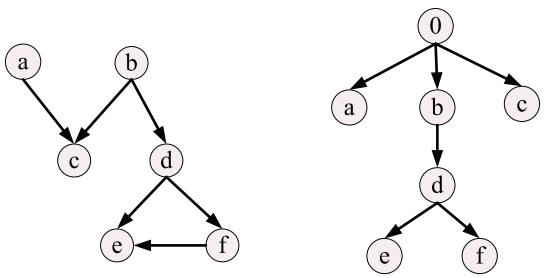



**Figure 2.** An example of a directed network and its dominator tree. In the left-side figure, node $e$ is dominated by node $d$ ($d = dom(e)$), then in the dominator tree of right-side figure, $d$ is an ancestor of $e$. If $d = dom(e)$, and every other dominator of $e$ is also dominator of $d$ (e.g., $b$ in left-side figure), then $d$ is the immediate dominator (i.e., $d$ is $e$'s parent in dominator tree). For $a$, $b$ and $c$ are not dominated by any other nodes in the network, we assume they are dominated by root 0 in the dominator tree.

$d = dom(e)$, and every other dominator of $e$ is also dominator of $d$ (e.g., $b$ in left-side figure), then $d$ is the immediate dominator (i.e., $d$ is $e$'s parent in dominator tree). For $a$, $b$ and $c$ are not dominated by any other nodes in the network, we assume they are dominated by root $0$ in the dominator tree.

The algorithm for generating the dominator tree of a directed network is straightforward. For each node $i \in V$, if $i$ is not the root of the directed network, we carry out the following steps:

- Starting from the roots of the directed network (nodes $a$ and $b$ in Figure 2), determine the set $S$ of nodes that reachable from the roots by routes which avoid $i$.
- The set $D_i = V - \{i\} - S$ is the node set which $i$ dominates.

Assume that $G$ has $N$ nodes and $M$ edges. The execution of above two steps requires $O(M)$ computational time. Hence, the computational complexity of the algorithm is $O(NM)$.

The authors in [22] have adopted dominator trees to analyze the error sensitivity ($ES$) and attack sensitivity ($AS$) of food webs. In is paper, $ES$ corresponds to random deletion of a node, while $AS$ corresponds to intentionally delete the most important node of $G$. The error sensitivity of a node of the network can be calculated as follows:

$$ES(i) = \frac{|dom(i)| - 1}{(n-1)} \tag{9}$$

where $|dom(i)|$ represents the number of nodes that dominate node $i$, and $n$ is the total number of nodes in the dominator tree. Hence, the error sensitivity and attack sensitivity of the network can then be calculated as follows:

$$ES = \sum_{i \neq 0} \frac{|dom(i)| - 1}{(n-1)^2} \tag{10}$$

$$AS = \max_{i \neq 0} \{ \frac{|\bigcup \{dom(j) = i\}|}{(n-1)} \} \tag{11}$$

where $|\bigcup \{dom(j) = i\}|$ represents the number of nodes that are dominated by $i$.

It can be calculated that star-like dominator tree (with root 0) has robust with smallest error sensitivity ($ES = 1/(n-1)$) and attach sensitivity $AS = 0$, in other words, star-like dominator trees are the most robust for random disturbance of the network and for deliberate attack. By definition, the structural robustness under random error is positively correlated with $1/ES$. Hence, we adopt $1/ES$ to represent the structural robustness of a network.

#### 6.1.3 Results and Observations

Figures 3 and 4 show the simulation results for $N = 1000$ and $M/N = 1.5$ and 1.86 respectively. Each node in the figures corresponds to a randomly generated directed network, and the entropy values and $1/ES$ are also calculated. The results show that for directed networks with the same number of nodes and edges but different connectivity, there is a positive correlation between structural robustness and

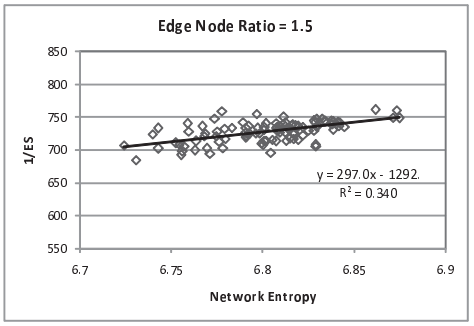



**Figure 3.** The entropy and robustness relationship of 100 randomly generated networks with $M/N = 1.5$. Each node in the figure corresponds to a randomly generated directed network.

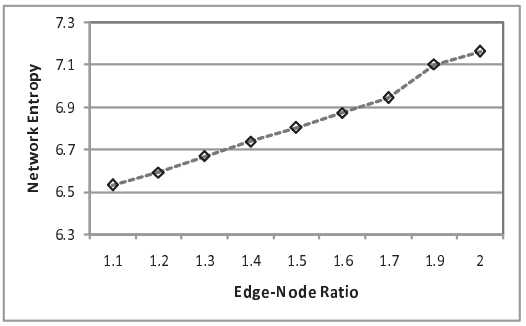



**Figure 5.** The average network entropy for different edge-node ratio with $N = 1000$. Each node in the figure is the average value of network entropy over 100 randomly generated networks.

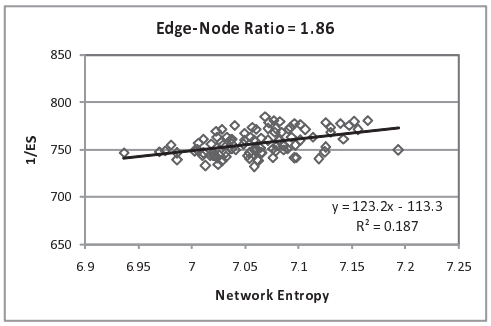



**Figure 4.** The entropy and robustness relationship of 100 randomly generated networks with $M/N = 1.86$. Each node in the figure corresponds to a randomly generated directed network.

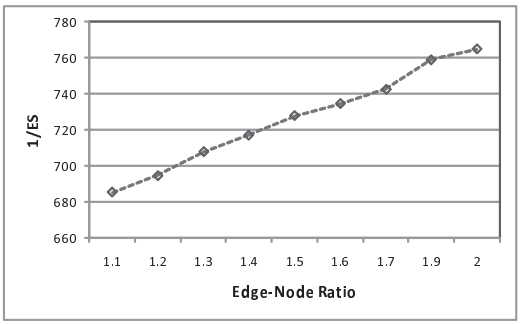



**Figure 6.** The average value of $1/ES$ for different edge-node ratio with $N = 1000$. Each node in the figure is the average value of $1/ES$ over 100 randomly generated networks.

network entropy. We have done simulations for different value of edge-node ratios, the positive correlation can also be observed.

From Figures 3 and 4, we can further find that as the value of edge-node ratio increase from 1.5 to 1.86, i.e., the number of edges increases from 1500 to 1860, the average value of $1/ES$ (i.e., structural robustness) increase from 727 to 756. It is reasonable because increasing the number of edges of a network will enhance its ability to maintain its connectivity in the face of random error. To validate the intuition, we do simulations for different values of edge-node ratio. The results are shown in Figure 6, where each node in the figure is the average value of $1/ES$ over 100 randomly generated networks. It can be observed that there is a positive correlation between $1/ES$ and the value of edge-node ratio. Similarly, as the value of edge-node ratio increases, we also find from Figure 5 that the network entropy also increase with the value of edge-node ratio. This observation indirectly validates Hypothesis 6.1 with respect to networks with the same number of nodes but different number of edges and connectivity.

We have also calculated the values of $AS$ and tried to find its correlation to network entropy. However, the results show that the correlation is not significant. (For page limitation, we do not include the results in this paper.) The reason is that by definition of $AS$, it only consider partial information of a network, i.e., the most vulnerable node. Hence, it is more appropriate for $AS$ to measure of fragility of a network other than robustness. In terms of this paper, we focus mainly on the network robustness. We leave network fragility for future study.

### 6.2 Functional Robustness

To validate Hypothesis 6.2, in this section, we will generate 500 networks with 500 different edge weight matrix (i.e., distribution strategies) but the same connectivity, to test the relationship between functional robustness and net-

work entropy. We also introduce a measurement, which is based on the satisfaction of all resource consumers after redistributing resources in the face of supply perturbations, to measure the functional robustness of the distribution dynamics. We assume that each node in the distribution network will retain the same distribution strategy under perturbation, i.e., the proportion of the distributed resources among its outgoing edges keeps constant.

#### 6.2.1 Measurement for Functional Robustness

For a distribution network $G$, each node may be either a supplier (e.g., the nodes 1 and 2 in Figure 1), or a consumer (e.g., the node 5 in Figure 1), or an intermediator (e.g., the nodes 3 and 4 in Figure 1). Both consumers and intermediators have their own energy demand. The robustness measurement focuses on the satisfaction of all the consumers and intermediators of the network after redistributing resources in the face of supply perturbation. Denote $D = \{n_1, \cdots, n_k\}$ the set of resource consumers and intermediators on the network. Each $n_i$ is associated with demand $d_i$. After redistributing, the satisfaction of $n_i$ can be calculated by $c_i/d_i$, where $c_i$ denote the amount of resources that are redistributed to $n_i$ based on the distribution strategy. In this case, the satisfaction of the whole network under perturbation can be calculated by the average satisfaction of all consumers and intermediators:

$$Sat = \frac{1}{k} \sum_{n_i \in D} \frac{c_i}{d_i} \tag{12}$$

If $n_i$ does not affected by the perturbation, we have $c_i/d_i = 1$; otherwise, if $n_i$ cannot be satisfied due to the perturbation, we have $c_i/d_i < 1$. Hence, the satisfaction value $S$ is positively correlated with the functional robustness of the network.

We simulate the supply perturbation by edge failure of the network. Denote $Sat_{e_{ij}} = \frac{1}{k} \sum_{n_i \in D} \frac{c_i}{d_i}$ the satisfaction of all consumers and intermediators under the edge failure of $e_{ij}$. Note that different edge failure will result in different value of $c_i$, which is determined by the distribution strategies. In this case, the robustness of the network can be measured by

$$R = \frac{1}{M} \sum_{l_{ij} \in G} S_{l_{ij}} \tag{13}$$

#### 6.2.2 Simulation Procedure

There are four major steps for the simulation in this section:

- Generate 500 networks with the same connectivity but different edge weight matrix;
- Calculate entropy values for all networks based on the definition in Section 3;

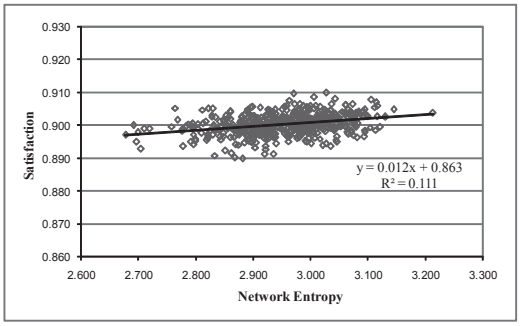



**Figure 7.** The entropy and robustness relationship of 500 randomly generated networks with the same connectivity but different edge wight matrix.

- For each network, fist evaluate the satisfaction of all consumers and intermediators under only one edge failure at a time. Then, average all satisfaction values for all $M$ links;
- Observe the relationship between network entropy and functional robustness.

### 6.3 Results and Observations

Figure 7 shows the simulation results of functional robustness and network entropy. It can be observed from Figure 7 that the robustness of the network is positively related to the network entropy defined in Section 3 with the following relationship:

$$R = 0.0124H + 0.8637 \tag{14}$$

However, the relationship between functional robustness and network entropy in Figure 7 seems not to be quite significant. The main reason is that the perturbation defined in Section 6.2.1 is too weak. It can be observed that all the satisfaction values are very high, i.e., within the region $[0.89, 0.91]$. To have a more accurate validation, we need (i) to do simulations for large perturbations, and (ii) to repeat the simulation procedure for networks with different connectivity. For page limitation, we leave these two problems for our future studies.

## 7 Conclusion and Future Work

In this paper, we have presented a general entropic measurement to analyze the robustness of resource distribution systems, which can be modeled as directed networks without loops. There are two kinds of network robustness, i.e., structural and functional robustness. We measure the structural robustness by error sensitivity ($ES$) and attack sensitivity ($AS$) with the help of dominator trees, which could

reflect the ability of a network to maintain its resource distribution under random error and intentional attack. We use resource consumers' satisfaction to measure functional robustness under resource supply perturbations. To study the relationship between network robustness and entropy, we firstly present a theoretical analysis based on Demetriusa et al.'s work [9][10] which shows that the network robustness and entropy are positively correlated. Based on the theoretical results, we have proposed two hypotheses, which state that the network entropy defined in this paper has a strong positive relationship with both structural and functional robustness for distribution networks. Finally, simulations are carried out to validate the two hypothesis.

With respect to resource distribution systems, there are still two challenges that need to be addressed in the future:

- For complex distribution systems (e.g., smart grid), how to find the critical components that are essential for the robustness of the system? The authors in [17] have adopted network entropy to characterize the essential nodes of protein interaction networks, is it possible to identify critical components of resource distribution systems using the entropy defined in this paper?

- How to build a robust resource distribution system? The authors in [24] have attempted to improve scale-free networks' robustness to random failures by entropy optimization. The network entropy proposed in this paper can also play as an important indicator, i.e., we need only to optimize the network entropy to pursue a robust network. However, since the entropy is related to both structural and functional robustness, it becomes a challenging problem to balance network structure and distribution dynamics during optimization.

These will be pursued in our future studies.

## References


[1] R. Albert, H. Jeong, and A. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[2] K. Anand and G. Bianconi. Entropy measures for complex networks: Toward an information theory of complex topologies. *Physical Review E*, 80:045102, 2009.

[3] L. Arnold, L. Demetrius, and V. M. Gundlach. Evolutionary formalism for products of positive random matrices. *Annals of Applied Probability*, 4:859–901, 1994.

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavezf, and D. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006.

[5] A. Z. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. L.Wiener. Graph structure in the web. *Computer Networks*, 33(1-6):309–320, 2000.

[6] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85:5468C5471, 2000.

[7] J. Cohen and C. Newman. A stochastic theory of community food webs: I. models and aggregated data. *Proceedings of the Royal Society B: Biological Sciences*, 224:421–448, 1985.

[8] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Breakdown of the internet under intentional attack. *Physical Review Letters*, 86:3682C3685, 2001.

[9] L. Demetrius, V. M. Gundlach, and G. Ochs. Complexity and demographic stability in population models. *Theoretical Population Biology*, 65(3):211–225, 2004.

[10] L. Demetrius and T. Manke. Robustness and network evolutionłan entropic principle. *Physica A: Statistical Mechanics and its Applications*, 346(3-4):682–696, 2005.

[11] Y. Gong and Z. Zhang. Network robustness due to multiple positive feedback loops: A systematic study of a th cell differentiation model. *Signal Transduction Insights*, 2:1–12, 2010.

[12] R. Guimerà, A. Arenas, A. Díaz-Guilera, and F. Giralt. Dynamical properties of model communication networks. *Physical Review E*, 66(026704), 2002.

[13] F. Jordán and I. Scheuring. Network ecology: topological constraints on ecosystem dynamics. *Physics of Life Reviews*, 1(3):139–172, 2004.

[14] A. I. Khinchin. *Mathenatical Foundations of Information Theory*. Dover, New York, 1957.

[15] H. Kitano. Towards a theory of biological robustness. *Molecular Systems Biology*, 3(137), 2007.

[16] Y.-K. Kwon and K.-H. Cho. Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. *Bioinformatics*, 24(7):987–994, 2008.

[17] T. Manke, L. Demetrius, and M. Vingron. An entropic characterization of protein interaction networks and cellular robustnes. *Journal of the Royal Society Interface*, 3:843–850, 2006.

[18] Y. Moreno, J.B.Gómez, and A. F. Pacheco. Instability of scale-free networks under node-breaking avalanches. *Europhysics Letters*, 58(4), 2002.

[19] 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.

[20] M. L. Sachtjen, B. A. Carreras, and V. E. Lynch. Disturbances in a power transmission system. *Physical Review E*, 61:4877–4882, 2000.

[21] R. V. Solé and S. Valverde. Information theory of complex networks: on evolution and architectural constraints. *Lecture Notes in Physics*, 650:189–207, 2004.

[22] A. Stefano and B. Antonio. Who dominates whom in the ecosystem? energy flow bottlenecks and cascading extinctions. *Journal of Theoretical Biology*, 230(3):351–358, 2004.

[23] A. Vázquez1 and Y. Moreno. Resilience to damage of graphs with degree correlations. *Physical Review E*, 67(015101), 2003.

[24] B. Wang, H. Tang, C. Guo, and Z. Xiu. Entropy optimization of scale-free networks robustness to random failures. *Physica A*, 363(2):591–596, 2006.

# A Computational Study on the Impact of Human Traveling Behaviors on Infectious Spreading Dynamics

Shang XIA
Department of Computer Science, Hong Kong Baptist University

## Abstract

*Infectious disease spreading dynamics among host population include two aspects: (1) disease infection through individuals' infectious contacts; (2) disease transmission through individuals' movements. Human plays an intermediary role for disease infection and transmission and thus human behaviors will directly or indirectly influence disease spreading dynamics. Yet the impacts of human behaviors still remain to be systematically understood. In our study, we develop a bottom-up multi-entity based individual traveling model that adopted with social attraction traveling mechanism to represent human collective traveling patterns and a direct contact based SIR disease infection model. By simulating the infection dynamics of 2009 H1N1 epidemics with different human mobility and contact patterns, we propose that the patterns of human behaviors play a determinant role in infectious disease spreading process as well as the disease pathological infection mechanism.*

## 1 Introduction

In recent years, the intensive epidemic outbreaks of infectious diseases, such as SARS in 2003 [34] and H1N1 in 2009 [35], have imposed great threatening on the safety of human public health. During the process of these disease infection spreadings, human beings are always playing roles of infection host and transmission intermediary, thus the dynamics of human traveling activities, such as the population aggregation or dispensation, in the underlining human social structures, such as the distribution of workplaces or schools, will create dynamics trends in disease infection spreadings.

A question of fundamental importance to epidemiology is understanding the disease infection spreading dynamics through human social traveling activities. That is, what kind of impacts human traveling behaviors will impose on disease infection spreadings and how they will shape the landscape of infection spreading dynamics. The study of the relationship between human traveling patterns and disease spreading dynamics thus have realistic meanings for making public health policies which aim at preventing the outbreak of epidemics by reducing the rate of disease spreadings through the interventions on individuals traveling behaviors.

To understand disease spreading dynamics in terms of human traveling activities, an extensive body of work has been explored, in which various simulation models have been proposed, such as homogeneous mixed compartmental models and human social contact network models. In the simplest model of homogeneous mixed compartmental model, SIR model [12][13], the host population are divided into several infection compartments, such as susceptible (S), infected/infectious (I) or recovered (R), and population age groups to represent individual's infection state and population age structure during disease spreadings [28][21][22]. The individuals within age groups are assumed to be homogenous mixed and the traveling activities between age groups are described by cross group contact frequency, both of which be represented by a contact matrix with respect to population age structure. The disease infection dynamics are governed by a set of differential equations, in which the rate of newly infection in susceptible population is proportional to $-\lambda S$, where $\lambda$ is the infectious contact rate, an estimation of infectious contacts from each age group, and infected individuals recover at rate $\gamma$ per unit time, such that the infection dynamics is characterized by $\lambda S - \gamma I$. The conventional SIR model is quite successful in reproducing and explaining the disease infection behaviors in the real world with large population size. However, limitations are also obvious, such as the basic assumption of individuals' homogeneous mixing behaviors, which could not reflect the heterogeneities of human contact pattern, and the contact frequency matrix, which is statistically estimation to characterize the individuals' traveling and contact frequency. Both of them have obscured the underlining impact of human traveling behaviors on disease spreading dynamics.

To address these shortcomings in homo-mixing compartmental models and describe human traveling and contact behaviors more precisely, individual-based human contact network models are widely discussed. These network based models simulate disease propagations in terms of peer-to-peer contact relationships. For human contact network, nodes present host individuals and edges connected two individual nodes mean a contact relationship between them. Therefore human traveling and contact activities can be characterized by the topology of a contact network, which provide a framework for the analysis of the relationship between disease spreading dynamics and the patterns of human traveling behaviors. Many prominent progress in the field of disease spreading dynamics on contact network have been achieved during the past several years. The work of Newman on epidemic spreading in random networks [29][27][23], for example, revealed that the probability of a major epidemic depends on the average degree (connectivity) of the network. Pastor-Satorras and Vespignani showed that epidemics are always possible in populations whose interpersonal contacts are power law-distributed [30]. Realistic and highly-structured contact networks formed from real-world statistics for population composition were constructed to model SARS transmission in Vancouver, Canada [24] and to capture the movement of individuals between locations in a city [14].

Compared with the homogeneous mixed compartmental models, the contact network based infection models have successfully simulate disease spreadings through individual to individual contact relationships, which is a representation of the individuals' heterogenous traveling and contact behaviors in terms of the topology of contact network. However, there are still some imperfections in these network based epidemic studies. For one thing, these network models are still the top-down and statistical analysis of human traveling behaviors, which is characterized by the topology features of the proposed contact network, such as small world, power law degree distribution. For another, these contact networks on which infectious disease spreads are static representation of human traveling patterns and fail to present the dynamics of human traveling activities. The question therefore arises what kind of features the dynamics of infectious disease spreadings will have when adopted with a dynamical human social network that emerged from individuals' traveling behaviors.

We explore the answer of this question by simulating realistic and detailed underlying human social networks that are built from relevant individuals traveling behaviors, which allows the structure of network to evolve according to traveling behavioral rules collected from the statistical observations of real human traveling activities. Based on this proposed dynamical human social network, we can simulate the disease infection spreading dynamics and then directly observe how the dynamics of human traveling network might be influenced by the individual traveling behaviors, and how, in turn, the individuals' traveling behaviors affect the infectious disease spreading dynamics.

In this paper, we first build up a dynamical social network emerged from a bottom-up mechanism of individuals' traveling activities which are characterized by the human statistical traveling patterns. For this proposed social network, nodes represent for the individuals' visiting locations; edges are defined as the frequency of individuals' collective visits of nodes. The edges connected the two nodes will be added/removed depending on the history of individuals' visits and thus the structure of this social network's connectivity is dynamically evolved according to the behavioral patterns of individual's travelings. We track the dynamics of human social network and the dynamics of infectious disease spreadings under this framework. Simulations have been generated for the H1N1 infection dynamics and we present a comparison of our simulation findings to the results of infection dynamics on typical human social networks in Refs[][]. We demonstrate that .... Furthermore, our study provide a unique insight into how and why the topology of social network affects the disease infection spreadings through individual's traveling behaviors.

The remainder of this paper is organized as follows: Section 2 provide a introduction of metapopulation system. Section 3 states the human behavioral patterns in infectious disease spreading process. Section 4 presents simulation and results analysis. Section 5 concludes the whole paper and highlights the major contribution of this paper.

## 2 Related Works

A framework for our study of the impact of human traveling behaviors on infectious disease spreading dynamics includes the following issues, (1) human traveling patters, (2) simulation model for individual traveling activities, (3) disease spreading dynamics on a social network. In this section, we will survey some previous works on the related topics, based on which we will expand our discussions.

In the real world, human traveling patterns can be interpreted by the statistical observations of several realistic data sets, such as transportation infrastructures, urban comminuting data and census information or even circulation records of bank notes. In air transportation network, the coupling is provided by the number of passengers

traveling on a given route connecting two airports, thus estimating the transmission of passengers between two corresponding cities. Barrat *et al.* [1] studied the world wide air transportation network based on the Interactional Air Transportation Association database. Guimera *et al.* in [16] proposed that the worldwide air transportation network is a scale-free small-world network with a multi-community structure. Colizza *et al.* in [10] depicted the air transportation network in terms of passenger capacity and respective urban population data and found that the obtained network was highly heterogenous both in connectivity pattern and the traffic capacities, which also exhibited heavy tails and very large statistical fluctuations.

Besides air transportation network, De Montis *et al.* in [26] and Chowell *et al.* in [8] represented the human social traveling network through the analysis of interurban commuting traffic flows. Brockmann *et al.* in [5] [6] reported a quantitative assessment of human traveling statistics by analyzing the circulation of bank notes in the United States and pointed out that the distribution of traveling distances decays as a power law and the probability of remaining in a small, spatially confined region for a time T is dominated by algebraically long tails. Then the general statistical features of human traveling patterns are summarized by Lee[20] in the following four aspects,

- Truncated power-law distribution for traveling distance and staying-times. [5][15][31]
- Heterogeneously bounded mobility areas. [15]
- Truncated power-law inter-contact times. [7][19]
- Fractal waypoints.

Based on these above mentioned direct or indirect statistical analysis of human traveling behaviors, it is clear that some universal patterns do exist in human collectively traveling activities. Eubank and Barrett in [14][1] presented an agent based simulation tool called EpiSims, which explored a large scale, dynamic bipartite contact graphs generated from the Transportation Analysis and Simulation System (TRANSIMS) to model the physical contact patterns that result from movements of individuals between specific locations. In their study, they found that the generated contact network showed the features of small-world-like graph and the locations graph was scale-free. Hackney and Axhausen in [17][18] proposed a multi-agent transportation simulation tool named as MatSim to model a dynamic social network based on individuals' traveling behaviors. In their study, the interactions and exchanges between agents have been analyzed and the relative influence of socializing and traveling behaviors have been discussed. Another human mobility network simulation model is proposed by Lee in [20] called SLAW (Self-similar Least Action Walk). In his model, the simulation of synthetic walk traces could represent the significant statistical patterns of human mobility, such as truncated power-law distributions of flights, pause-times and inter-contact times, fractal way-points. There are also many other models to simulate the human mobility dynamics and all of these models present a various kinds of simulation methods which can generate human social traveling networks based on individuals' traveling behaviors and present the heterogeneities of human mobility patterns. With these traveling simulation frameworks, we can validate the assumptions of human traveling behaviors and estimate the effect of human traveling patterns on the generation of social networks, which would provide a solid and reasonable foundation for analyzing the disease infection spreading dynamics that coupled with human traveling activities.

Eubank in [14] simulated the disease outbreaks in realistic urban social network with an agent based simulation tool - EpiSims - and analyzed the relative merits of several mitigation strategies for smallpox spread. Followingly, Barrett in [2] proposed an agent-based scalable and parallel algorithm, called EpiSimdemics, to simulate the spreading infectious contagion with a large scale and realistic social contact network. Colizza [11] provided an analysis of the dynamics of infectious disease spreading in a metapopulation system adopted with the heterogeneous spatial structure of subpopulations and the statistical properties of human mobility patterns. In his study, the basic reaction-diffusion equations were used to describe the disease invasion dynamics among the connected subpopulations and the properties of global threshold for epidemic outbreak was discussed. Moreover, an evolving social contact network based on populations' demographical dynamics is proposed by Christensen in [9], in which the coupling relationship between the demographical dynamics of host population and the dynamics of disease infection spreading is simulated and discussed. All of these mentioned progress on epidemic dynamics based on human traveling networks provide a solid foundation for our study of disease spreading dynamics on a dynamically evolved social network, which will be detailed described in the following section.

## 3 Problem Statement

Generally speaking, the systematically study of infectious disease spreadings accompanied with human traveling activities includes two aspects: one is how to model human autonomous movements in individual level while collectively they can reflect the statistical patterns of human travelings that have been observed in the real world; the other one is what is the relationship between disease in-

fection spreading dynamics and human traveling dynamics, that is to say, what kind of impacts will human traveling behavior rules impose on the disease infection spreading process. With these concerns, the problem discussed in the following sections is equivalent to a modeling of human traveling behaviors and an analysis of network infection dynamics.

To address these issues, we will propose a bottom-up multi-entity based modeling framework to characterize human's autonomous travelings and the resulting disease infection spreadings. First, we introduce a social connection network to represent individuals' social contact relationships. Then we propose rectangular grid as the field of individuals' movement, named as movement grid. Third, each individual can autonomously move on the proposed movement grid based on their mobility rules, along with which the infectious disease can spread among the whole population.

***Definition1:*** *Social Connection Network. Graph* $G =< V, L >$ *is network where* $V = \{v_1, v_2, ..., v_N\}$ *is the set of nodes (individuals), and* $L = \{< v_i, v_j > |1 \leq i, j \leq N\}$ *is the set of links (connection). N represents the total number of individuals in the network, and* $< v_i, v_j >$ *means a connection relationship between* $v_i$ *and* $v_j$*.*

Social connection network $G =< V, L >$ represents the social contact relationships between individuals. The topology and community structure of this network represent the different kinds of social organizing relationships.

***Definition2:*** *Movement Grid. A* $m \times n$ *rectangle grid is used as the individuals' movement field. Individuals can located on one point of the grid or move to others and the statistical data of individuals traveling behaviors can be calculated on this grid.*

***Definition 3:*** *Traveling Behavioral Rules. A set of behavioral rules will be used to define individuals traveling activities. Based on these rules, individuals can locally and independently determinate their movements in the following steps rather than a central coordination mechanism.*

We will propose a bottom-up multi-entity based human traveling model to characterize individuals' traveling behavioral rules, which collectively reflect the statistical patterns of human traveling activities, such as the distribution of human traveling distance that can be described by a truncated power law [5][15][31]. Individuals in social connection network will autonomous move based on social attraction mechanism proposed by [3][4] in his Home Cell Mobility Model (HCMM). The main idea behind this traveling mechanism is individuals would incline to move towards others who have social connections with them.

***Definition 4:*** *Infectious Disease Spreading Dynamics. For infectious disease spreading, there are three infection states to be used, susceptible (S), infectious (I) and recovered (R). In our modeling, the infection will be transmitted with a stochastic infection probability* $r_{inf}$*, if the infectious (I) and susceptible (S) individuals have a nearby location relationship on the movement grid. Then the infected individuals will be recovered (R) after recovery periods* $T_{rov}$ *and will be permanently immunized from infection again. Thus, we can observe the disease spreading dynamics along with individuals' traveling activities.*

The previous work of studying the disease infection dynamics on human contact network with a topology structure to characterize human statistical traveling patterns have some limitations and flaws:

1. Human contact network is not the direct disease transmission network. Disease infections happen between individuals who appeared in the same locations during the same time interval, rather than their social connection relationship, that is to say, infection transmission is possible between two individuals who do not have edge connections in the social contact network.

2. The social contact networks are mostly static networks used as a description of statistical features of human traveling patterns. However individuals' traveling and infection are two dynamical coupled processes. Thus the top-down methods based on statistical descriptions fail to describe the impact of human mobility mechanism on disease spreading dynamics.

Based on the above mentioned issues, the specific research questions and contributions in our study are as follows:

1. We use a bottom-up and multi-entities based human traveling models to model the statistical human traveling pattern – the truncated power-law distribution of human traveling distance.

2. We simulate infectious disease spreading dynamics based individuals' autonomous traveling activities rather rather individuals' social contact relationships.

3. We discuss the impact of human traveling behaviors on infection spreading dynamics, which will provide us a solid foundation for intervening epidemic outbreak by controlling social mobility activities.

## 4 Human Social Network and Infection Spreading Dynamics

In this section, we will present the detailed formulations of disease infection spreading model based on individuals' traveling behaviors. We first simulate human traveling behaviors adopted with social attraction mechanism by introducing a human mobility model named HCMM [3][4], a bottom-up multi-entities based simulation model, in which individual entities will be linked by a social connection network and mobility activities will be determined by their connection relationships. Also, we will propose that the directly nearby located individual pairs, as a result of individuals' mobility activities, will transmit the infectious disease. Thus the infection spreading dynamics together with individuals's traveling activities can be observed.

### 4.1 Individual Social Connection Network

To simulate human traveling behaviors, we first introduce a social connection network to represent the social connection relationships in human society. In this study, social connection relationships have broad meanings, such as mutual communications, friend relationships or some kinds of physical contacts, which will be used to characterize the social relationship of keeping in touch with each other. Graph $G =< V, L >$ is a social connection network where $V = \{v_1, v_2, ..., v_N\}$ is the set of nodes to represent for individuals, and $L = \{< v_i, v_j > |1 \leq i, j \leq N\}$ is a set of links meaning a social connection between two individuals $v_i$ and $v_j$. In this model, N represents the total number of individuals in the network. These social connectivity relationships provide foundation for individuals to traveling.

The proposed social connection network will be generated by an edge rewiring mechanism proposed by Boldrini in [3] and inspired by the Caveman model [32] to achieve a small-world like topology and it works as follows. First the set of individuals will be grouped into several communities, $N_C$ is the number of communities. For the reasons of clear observations in the simulation results, we assume that the number of individuals in each is equal referred as $n_c$. Then a link between two nodes means a social connection relationship, represented by a weighted undirected linkage $w_{ij}$. The value of the weight $w_{ij}$ implies the strength of social connections, which will be computed as the social attraction for individuals to traveling towards. For the edge connections, first the nodes in the same community will be full connected. Then each within community links will be rewired with the nodes in outside communities with a rewiring probability $p_{rewire}$.

Based on nodes connectivity mechanism of within community fully connecting and cross community probabilistic rewiring, we can get a social connection network with a near small world topology, as shown in Figure 1, which provides a foundation for individuals' traveling activities adopted with the social attraction mechanism.

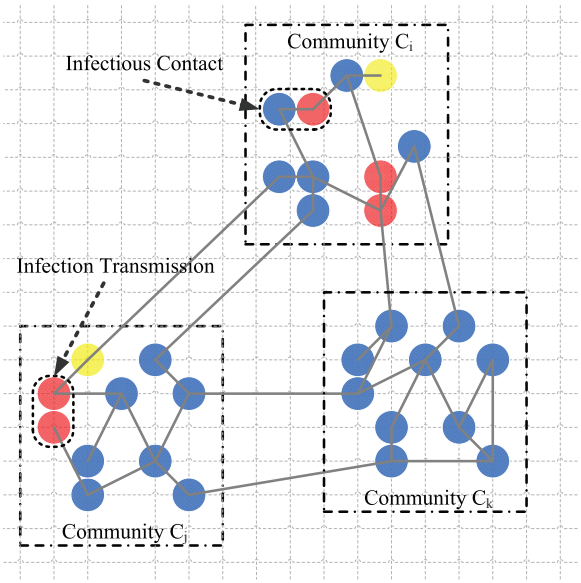



**Figure 1. HCMM modeling.**

### 4.2 Individual Traveling Behavioral Rule

In our study, we use Home Cell Mobility Model (HCMM) proposed by Boldrini in [3] as our basic framework to simulate human individuals' traveling activities adopted with the mobility rule of social attraction [4]. The basic idea of social attraction mechanism is that individuals are inclined to move towards the place where they have many socially connected counterparts. In our study, the socially connected nodes (individuals) will locate on a $m_G \times n_G$ movement grid, and nodes that belongs to a same community will be confined within a region of $m_C \times n_C$ in the movement grid and we assume that there are no overlaps between the location regions of two communities. Based on the movement grid, individuals can move within their original communities or towards other communities based on result of traveling behaviors adopted with social attraction mechanism as shown in Figure 2.

Adopted with the mobility rule of social attraction mechanism individuals in movement grid will select their traveling movements that determined by their social connectivity with other nodes in the same or different communities, that is, individuals will choose their traveling destination based on the comparative social attractions of internal and external communities. The traveling rule of social attraction is designed as follows. If individual (node) $k$ currently locate in its original community, it can travel inside original com-

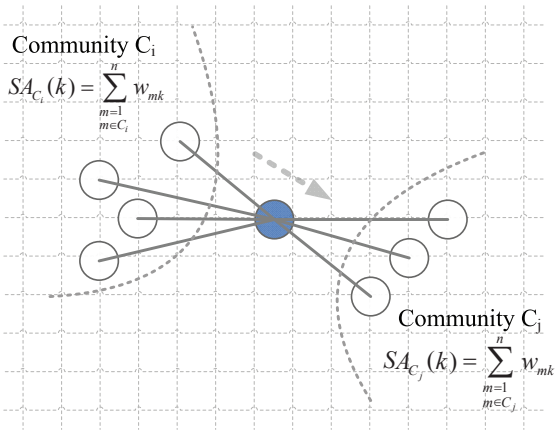



**Figure 2. HCMM Mobility Modeling.**

munity or towards external communities. The probability for individual $k$ to remain inside or move towards external community $i$ is $P_{C_i}(k)$, which is proportional to the social attraction of community $i$, $SA_{C_i}(k)$, exerted by nodes that have connectivity relationships with node $k$ in community $i$. For community $i$, the social attraction $SA_{C_i}(k)$ and the probability of movement towards community $i$, $P_{C_i}(k)$. are computed by,

$$SA_{C_i}(k) = \sum_{\substack{m=1 \\ m \in C_i}}^{n_C} w_{mk} \tag{1}$$

$$P_{C_i}(k) = SA_{C_i}(k) \Big/ \sum_{j=1}^{N_C} SA_{C_j}(k) \tag{2}$$

In these equations, $n_C$ is the number of nodes in each community, $N_C$ is the number of communities. Based on $SA_{C_i}(k)$ and $P_{C_i}(k)$, node $k$ could stochastically choose the traveling destination in the next step. If node $k$ currently locate on the external communities, it will remain inside its current community with probability $p_e$, or return back to its original community with probability $1 - p_e$.

Once the traveling destination for next step movement is stochastically selected, node $k$ will move to a randomly chosen free point in the grid region of the destination community. The individual traveling distance is then calculated by the length from the original position to the final located position in movement grid. Individuals travelings behaviors in the movement grid can be identified two categories: short traveling (within original community) and long range traveling (cross different communities). Whether the traveling type is short range or long range is determined by the social connection with the nodes in internal or external communities. Thus we can simulate human traveling pattern of traveling distance with the mobility rule of social attraction.

### 4.3 Infectious Disease Spreading

We use SIR model to describe the disease infection spreadings on our proposed human connection network along with individuals' traveling activities. Individuals disease infection have three stages, they are susceptible (S), infectious (I) and recovered (R). The disease will be transmitted if and only if the susceptible and the infectious individuals have a nearby traveling locations in the movement grid, which we call it as a infectious contact, as shown in Figure1. The probability for an infectious contact to be a successful infection is remembered as $r_{inf}$. The infected individuals will be recovered in a certain time units of recovery period, $T_{rov}$. The recovered individuals are assumed to be immunized to second infections and also can not infect others.

In our study, disease infection spreading is the result of individuals' nearby infectious contact because of their traveling activities rather than individuals' social connection relationships. Most of recent infectious diseases in the real world are transmitted based direct physical contact between individuals. However this kind of infectious contact might not be the result of the social connection relationship, that is to say, the two socially connected individuals might not have the infectious contact, nevertheless, the unknown individuals might exposed in the same locations, which is the result of mobility activities rather than the social contact relationships. The social connection relationships are the direct determinant of traveling behaviors, such as social attraction mechanism, and thus it can be viewed as indirect factor for disease spreadings. By modeling disease infection transmission direct correlated with individuals traveling behaviors through infectious contact of nearby locations and indirect coupled with the individuals' social connection relationships, we can more precisely characterize the disease infection spreading dynamics in the real world.

## 5 Simulation and Results

The objective of this study is to understand the impact of human traveling behaviors on infectious disease spreading dynamics. To achieve this goal, we have introduced a bottom-up multi-entity based individuals traveling framework with the social attraction mobility mechanism in section 4. In this section, some experiments will be used to investigate the relative impact of human traveling behaviors on the disease spreading dynamics.

For our experimental simulations, we have generated four synthetic social connection networks of different topologies to represent the social connection relationships between individuals, by adopting different rewiring rates and community-based structures which are proposed by Boldrini in [3]. Based on these social connection networks,

**Table 1. Parameter Sets in Social Connection Network and Individual Mobility Behavior**

| | | |
|---|---|---|
| Entity Number | $N_E$ | The total number of individual entities. |
| Community Number | $n_c$ | The total number of communities |
| Community Size | $n_s$ | The number of individuals in a community |
| Rewire Probability | $r_e$ | Cross communities edge rewiring probability |
| Average Path Length | $L_{avg}$ | Measures of generated network topology |
| Clustering Coefficient | $C$ | Measures of generated network topology |
| Average Degree | $D_{avg}$ | Measures of generated network topology |

we simulate individuals' collective autonomous traveling activities under the mobility rule of social attraction proposed by by Boldrini in [4]. We observe the simulation results in terms of the statistical features of individuals' traveling distance and find out that the proposed human individuals traveling patterns are accorded with the real world human traveling activities, that is truncated power-law distribution for traveling distance [5][15][31]. Thus we can use the proposed individuals' traveling model as a foundation to discuss the impact of human traveling patterns on disease spreading dynamics.

### 5.1 Social Connection Network

Social connection network provides a network structure to present the connection relationships among individuals which can be uesed to simulate the human social relationships in the real world. This network will be generated based on predefined community structure and linkage rewiring rules introduced in section 4. With these social connections, individuals in our model can autonomously move within or cross communities by following the social attraction mobility mechanism. In this section, we will generate four social connection networks with different rewiring rates and different community structures, simulate the individuals' autonomous mobile behaviors on these social networks and then observe their collective patterns.

The total amount of individual entities $N_E$ in our simulation experiments is 500, the number individuals in a signal community, the community size $n_s$, is chosen as 50 and 25, that is to say the number of community $n_c$ of the two scenario are 10 and 20 respectively. The linkage rewiring rates $r_e$ are selected as 0.05, 0.1, 0.2, which represent for the intensity of the connection between different communities. With these settings, through the within community full connection and cross community rewiring connection algorithm, we can generate four social connection networks to present the social relationships among these individual entities. The topology features of these generated social connection networks are shown in Table 2 and the degree distribution is shown in Figure 3.

The parameter set of the generation of social connection network is shown in Table 1 and they will play important roles in individuals' mobility activities and the disease infection spreading. The following are detailed description of these parameters.

**Table 2. Individual Social Connection Network**

| | $NW_1$ | $NW_2$ | $NW_3$ | $NW_4$ |
|---|---|---|---|---|
| $N_E$ | 500 | 500 | 500 | 500 |
| $n_c$ | 10 | 10 | 10 | 20 |
| $n_s$ | 50 | 50 | 50 | 25 |
| $r_e$ | 0.05 | 0.1 | 0.2 | 0.1 |
| $L_{avg}$ | 2.21 | 1.99 | 1.89 | 2.50 |
| $C$ | 0.78 | 0.61 | 0.40 | 0.59 |
| $D_{avg}$ | 51.43 | 54.11 | 58.86 | 26.68 |

- **Community Number** $n_c$. It is the total number of communities in a movement grid field and it will influence the traveling distance of individuals' cross community mobilities. These individual communities are randomly scattered on a fixed region movement, thus the more communities involved the less average distance among these communities have, which means the traveling distance of individuals might decreased.

- **Community Size** $n_s$**.** This is a parameter to describe the total number of individuals in a signal community, which is equal to the density of individuals in a community. It will have influences on both mobility dynamics and infection dynamics. As described in the model design of section 4, all of the individuals in a community will be confined within a certain region on movement grid, that is to say, the larger the community size $n_s$, the denser the individuals will gather, which will lead to a potentially shorter distance for within community travelings and a higher probability of a direct neighborhood relationship between two individuals.

- **Rewire Probability** $r_e$**.** During the generation of social connection network, individual nodes within a community will be fully connected and these inside community links will be randomly rewired with

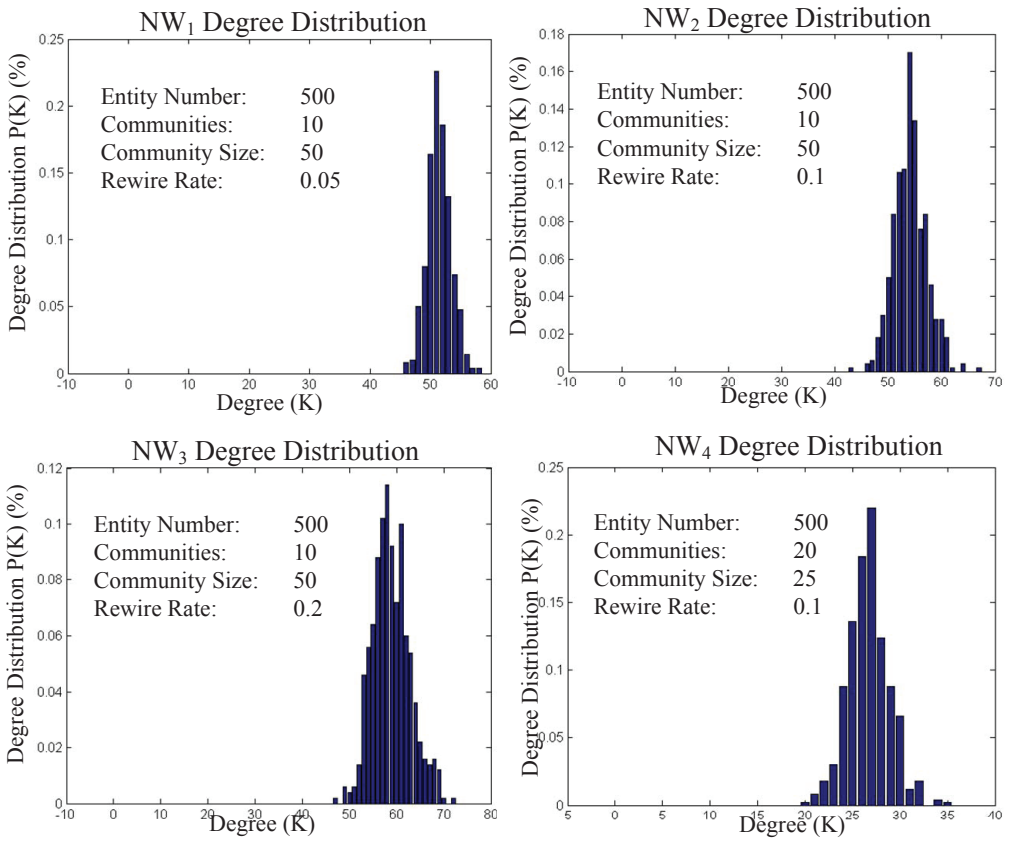



**Figure 3. Individual Social Network Degree Distribution.**

the nodes in outside communities with probability $r_e$. Thus the rewire probability $r_e$ determines the percentage of external connections for each community. These external connections will influence the value of social attractions of each communities for the destination selection of individual traveling activities.

For these parameters, how they will influence individuals traveling behaviors and infection dynamics will be analyzed in the following sections.

### 5.2 Individual Traveling Pattern

To simulate individuals' traveling activities, we use a $1000 \times 1000$ rectangle grid for individuals' movements. Individuals locate on the points and can move to others, during which the traveling distance can be calculated in terms of movement length on this grid. For each individual community, it will cover an sub-grid area of $10 \times 10$ and will be randomly scattered on the movement grid during the initiation stage, where we assume that there are no overlaps among these communities' covered regions.

In each steps, individuals will stochastically select its movement destination with the mobility rule of social attraction based on their connection relationships in social connection network, which is describe in section 4. After the destination community is chosen, individual will move to that community and randomly select a free point in the grid region of that community. After individual movement, the traveling distance will be logged. Thus the collective pattern of individuals' mobility behaviors can be observe after the simulations. Based on the four individual social network structure generated from the above section, we can get the statistical results of individuals' traveling behaviors, as shown in Figure 4.

Based on our model design, we know that for each individuals the traveling destination community can be its own original located community or other external communities. For these internal movements, we call them short range traveling, which is marked as traveling distance less than 10. For these external movements, they will be viewed as the middle range and long range travelings, which is characterized its traveling distance that middle range traveling is no more than 100 and long range traveling is larger than

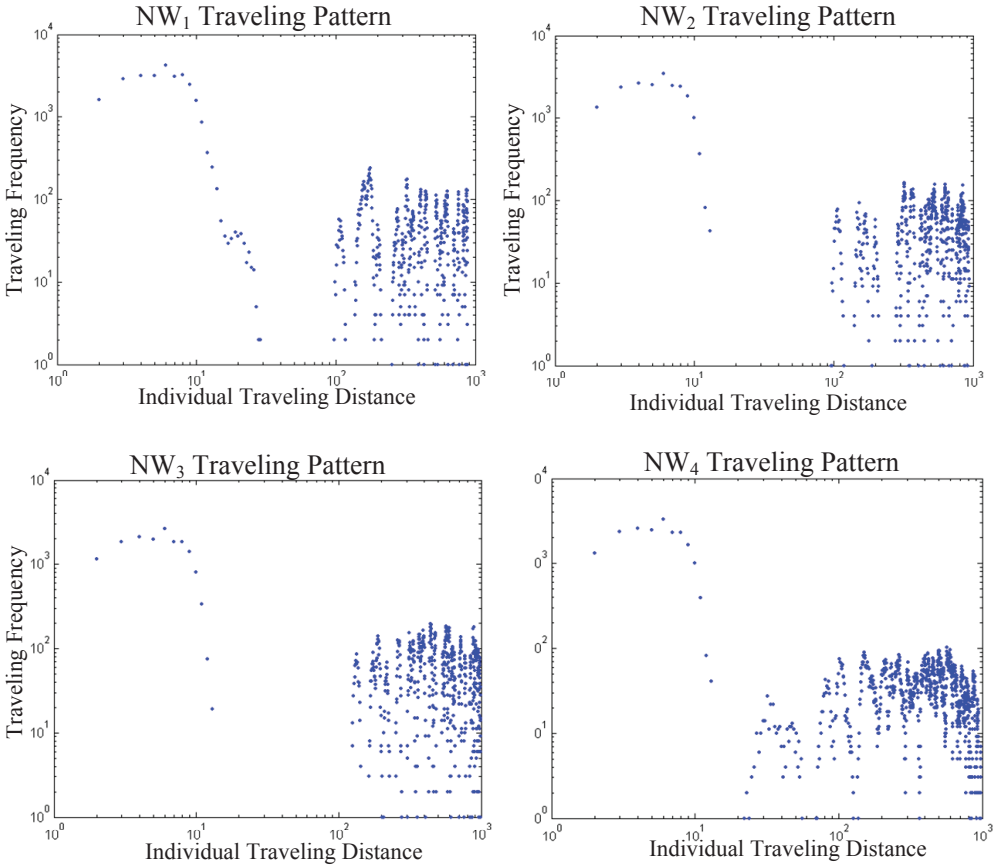



**Figure 4. Collective Individual Traveling Pattern.**

100. As is shown in simulation results in Figure 4, we can observe the following collective patterns of individuals' traveling behaviors, (1) short range traveling, near linear traveling distance distribution; (2) middle range traveling, near turned pow-law traveling distance distribution; (3) long range traveling, long tail effect in the distribution of traveling distance. These collective traveling patterns that emerged from our bottom-up individual movement behaviors are accord with the real world human mobility patterns, such as the scaling law of human mobility proposed by Brockmann in [5] with the statistical data of international banknote flows and the truncated Levvy flight pattern observed by Gonzalez in [15] based on the real mobility data sets of mobile phone users. Thus the individual traveling simulation model proposed in this paper provide us a reliable tool to study the impact of human mobility behavior on infectious disease spreading, which will be analyzed in the following section.

### 5.3 Disease Spreading Dynamics

In our study, we use the traditional SIR model to characterize individuals' infection dynamics. We assume that infections should only be transmitted when the susceptible individuals and infectious individuals have a direct neighborhood relationship on the movement grid, which will be named as infectious contact relationship in the following section. When a susceptible individual is exposed to an infectious contact it will be infected with a probability of individuals' infection rate $r_{inf}$ and the infected individuals will be recovered in $T_{rov}$ time units. Thus the disease infection spreading process can be describe as the susceptible individuals are infected through infectious contacts and then transmit the infection to other individuals in internal or external communities along with their traveling behaviors.

We will parameterize the disease infection model with pathogenical value of H1N1 swine flu virus in 2009 [25]. The individual infection rate $r_{inf} = 0.2$ and infection recovery period $T_{rov} = 7$ days. The simulation results of disease infection spreading dynamics with different individual traveling settings are shown in Figure 5 and Figure 6.

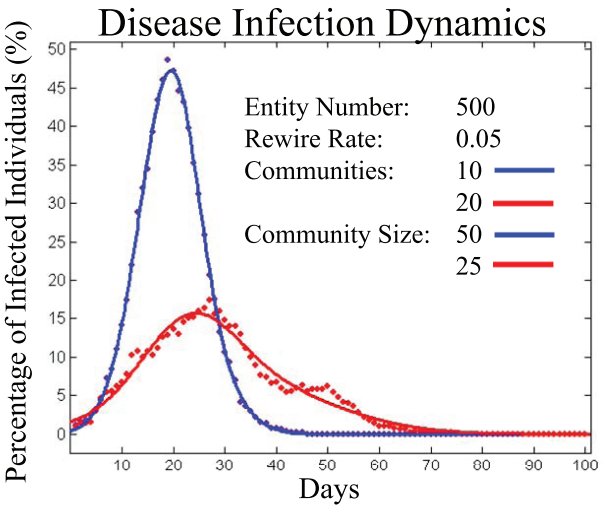



**Figure 5. Disease Infection Dynamics with different Community Size.**

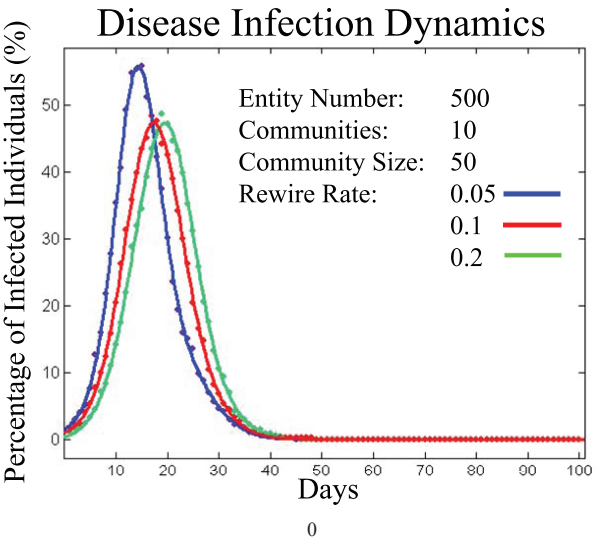



**Figure 6. Disease Infection Dynamics with different Social Network Rewire Rate.**

Figure 5 shows the disease spreading dynamics in network $NW_2$ and $NW_4$, which have different individuals community size, $n_s(2) = 50$ and $n_s(4) = 25$ that is equal to $n_c(2) = 10$ and $n_c(4) = 20$ in terms of community amounts in the movement grid. In our model, the larger community size means the average density of community population will be higher, which will lead to a higher probability for the susceptible individuals to engage in infectious contacts. Thus the disease infection spreading within community will be accelerated. Our simulation results show that the disease infection dynamics in individual social network $NW_2$ which have a twice time higher individual density to $NW_4$ will possess a higher percentage of infected population in the peak of infection outbreak, which means the severity of disease infection in higher density population community is intensified through individuals short range travelings.

Figure 6 provides the disease spreading dynamics in social network $NW_1$, $NW_2$ and $NW_3$ with different rewiring rate, $r_e(1) = 0.05$, $r_e(2) = 0.1$ and $r_e(3) = 0.2$. As we have discussed in section 5.1 rewiring rate $r_e$ represent for the connection relationships within and between communities. A larger rewiring rate means a thicker connection links among communities and a relative thinner connections within community. Our simulation results show that the infection dynamics in social network $NW_1$ with a relative smallest rewiring rate, $r_e(1) = 0.05$, have the most severe infection spreading dynamics that the highest value of infection cases in the peak of outbreak and fastest infection increasing speed. This means that the disease infection spreading are mainly deteriorated by the short range but high frequency travelings within a community.

## 6 Conclusions

In this paper, we have presented a bottom-up framework to study infectious disease spreading dynamics in terms of individuals' traveling behaviors. In order to investigate the relative impact of individuals' statistical traveling patterns on disease spreading dynamics, we introduce a social network structure and simulate individuals' self-mobile activities on a movement grid with the mobility mechanism of social attraction. Our simulation results show that the collective features of our proposed individuals traveling simulation is accord with the real world human mobility patterns. Specially, our analysis reveals the critical roles of individuals' traveling behavioral patterns played in infectious disease spreading process, such as the short range and long range traveling frequency.

In summary, our work has offered a means for epidemiology study. it enables us to evaluate the disease spreading process in the eyes of micro level individuals' prime traveling activities, which open a door for us to understand the relationship between human social dynamics and epidemics dynamics. This has significant meanings for preventing the mass outbreak of disease infections in human society.

## References

[1] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted net-

works. *Proceedings of the National Academy of Science*, 101(11):3747–3752, 2004.

[2] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe. Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks. *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, 2008.

[3] C. Boldrini, M. Conti, and A. Passarella. Users mobility models for opportunistic networks: the role of physical locations. *IEEE WRECOM*, 2007.

[4] C. Boldrini, M. Conti, and A. Passarella. The sociable traveller: Human travelling patterns in social-based mobility. *Proceedings of the 7th ACM international symposium on Mobility management and wireless access*, pages 34–41, 2009.

[5] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439:462–465, January 2006.

[6] D. Brockmann and F. Theis. Money circulation, trackable items, and the emergence of universal human mobility patterns. *IEEE Pervasive Computing*, 7(4):28–35, 2008.

[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.

[8] G. Chowell, J. M. Hyman, S. Eubank, and C. Castillo-Chavez. Scaling laws for the movement of people between locations in a large city. *Physical Review*, 68(6), 2003.

[9] C. Christensen, I. Albert, B. Grenfell, and R. Albert. Disease dynamics in a dynamic social network. *Physica A.*, 389(13):2663–2674, 2010.

[10] V. Colizza, A. Barrat, M. Barthelemy, and A. Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Science*, 103(7):2015–2020, 2006.

[11] V. Colizza and A. Vespignani. Epidemic modeling in metapopulation systems with heterogeneous coupling pattern: theory and simulations. *Journal of Theoretical Biology*, 251(3):450–467, 2008.

[12] D. J. Daley and J. M. Gani. *Epidemic Modelling: An Introduction*. Cambridge University Press, 2000.

[13] O. Diekmann and J. A. P. Heesterbeek. *Mathematical epidemiology of infectious diseases: model building, analysis and interpretation*. Wiley, 2000.

[14] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180–184, May 2004.

[15] M. C. Gonzlez, C. A. Hidalgo, and A.-L. Barabsi. Understanding individual human mobility patterns. *Nature*, 453:779–782, June 2008.

[16] R. Guimera, S. Mossa, A. Turtschi, and L. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Science*, 102(22):7794–7799, 2005.

[17] J. Hackney and K. W. Axhausen. An agent model of social network and travel behavior interdependence. *11th International Conference on Travel Behaviour Research*, 2006.

[18] J. Hackney and K. W. Axhausen. A model for coupling multi-agent social interactions and traffic simulation. *Eidgenossische Technische Hochschule, Institut fur Verkehrsplanung und Transportsysteme*, 2009.

[19] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnovi. Power law and exponential decay of inter contact times between mobile devices. *International Conference on Mobile Computing and Networking*, 2007.

[20] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Slaw: A mobility model for human walks. *IEEE INFOCOM 2009*, 2009.

[21] J. Medlock and A. P. Galvani. Optimizing influenza vaccine distribution. *Science*, 325(5948):1705 – 1708, September 2009.

[22] J. Medlock, L. A. Meyers, and A. Galvani. Optimizing allocation for a delayed influenza vaccination campaign. *PLoS Curr Influenza*, 2010.

[23] L. A. Meyers, M. Newman, and B. Pourbohloul. Predicting epidemics on directed contact networks. *Journal of Theoretical Biology*, 240(3):400–418, 2006.

[24] L. A. Meyers, B. Pourbohloulc, M. Newman, D. M. Skowronskic, and R. C. Brunhamc. Network theory and sars: predicting outbreak diversity. *Journal of Theoretical Biology*, 232(1):71–81, 2005.

[25] 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.

[26] A. D. Montis, M. Barthelemy, A. Chessa, and A. Vespignani.

[27] C. Moore and M. E. J. Newman. Epidemics and percolation in small-world networks. *Physical Review*, 61(5):5678C5682, 2000.

[28] 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.

[29] M. E. J. Newman. The spread of epidemic disease on networks. *Physical Review*, 66(1), 2002.

[30] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review*, 86(14), 2001.

[31] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong. On the levy-walk nature of human mobility. *IEEE INFOCOM 2008*, pages 924–932, 2008.

[32] D. J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.

[33] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[34] W.H.O. Consensus document on the epidemiology of severe acute respiratory syndrome (sars). Technical report, Department of Communicable Disease Surveillance and Response, 2003.

[35] W.H.O. Human infection with pandemic (h1n1) 2009 virus: updated interim who guidance on global surveillance. Technical report, Global Alert and Response (GAR), 2009.

# Semantic Indexing for Music Search with Adaptive Recommendation

DENG Jie

## Abstract

*Due to the rapidly increasing availability of digital music, advanced semantic music search and music recommendation are becoming important issues. Thus how to index music for semantic search and how to make recommendations in intrinsic of musical art are challenging problems currently. In this context, this paper used a semantic indexing method that is capable of flexibly retrieval music in semantic level. When the semantic and dynamic index has been established, updating, increasing and decreasing index scores are necessary to build a reinforcement index to gain accurately search results. Moreover, the paper also proposed useful and flexible music recommendation approaches, for example song-to-song matching and multi-song matching methods. In order to optimize the order of search result, genetic algorithm also used to re-ranking those near hidden music in the top. The proposed method will also support users with diverse needs when searching for music. The observations indicate that the present approach is able to get better performance.*

***Keywords****: semantic index, genetic algorithm, ranking, recommendation, music search*

## 1. Introduction

As the digital music becomes more and more huge and ubiquitous, music search becomes the vital important tool when people use music services in the web. The first generation search engines make great contribution to finding textual information on the web. Yahoo!Music represent the first generation search engine attempts to support audio search, which adopts text-based approach and the service is limited [6, 7]. Thus, non-textual multimedia documents such as music audio bring new challenges [3] to search engines – how to incorporate search by the musical content, which will achieve better music search result. Therefore, music search has significant commercial and research promise in nowadays. Many groups have already made a great effort on this area, for example, Last.fm is a popular music search engine also with a music recommender system. Pandora is an automated music recommendation service and custodian of the Music Genome Project which captures the essence of music at the fundamental level. Thus intelligent music search and recommendations will have becoming more and more popular.

According to the research, there are three key issues in audio-based music search: how to index the content of music objects, how to present the user with intuitive methods of querying music objects, and which music objects to present to the user and in which order. This paper mainly addresses the first and the third key issues and proposes a novel approach for semantic index of music data objects and improved method for music data ranking and recommendation when music browsing.

This paper mainly focuses semantic index of music and music recommendation. Thus, the paper is organized as follows. In section 2, previous literature reviews on music search and recommendation will be described. Section 3 gives an overview of music search. A detailed description on proposed methodology will be introduced in Section 4, which consists of music representation, semantic indexing, index updating, incremental and decreased Index, music matching and browsing. Experiment observations will be given in Section 5. Finally, some conclusions and future work are discussed in Section 6.

## 2. Related Work

Extraction of music feature vectors is the basis of music search system. Nicola Orio [5] has already given some music characteristics and features in "Music Retrieval: A Tutorial and Review". According to the Music Genome Project depict, a given song contains approximately 400 attributes. Each attribute corresponds to a characteristic of the music. Most of the today's approaches content-based music search systems are based on melody, timbre and rhythm as the main features and often only content descriptors. Therefore, depending on these main features, there are three category approaches: index terms, sequence matching, and geometric methods which deal with polyphonic music scores. S. Downie and M. Nelson has presented that melodies were indexed through the use of N-grams. M. Melucci and N. Orio presented an alternative approach, where indexing was carried out by highlighting musically relevant approaches. Most importantly, an architectural paradigm for collaborative semantic indexing of multimedia data objects has been present in [1]. Leung has also researched multimedia data mining and searching through dynamic index evolution in [2]. C. R. Buchanan presented some approaches to

semantic-based audio recognition and retrieval in [9]. Douglas Turnbull [4] gives semantic annotation and retrieval of music.

Some of the famous commercial music services (e.g. Pandora.com [15], Last.fm [16]) own music search and recommendation functions, which also rank the results based on relevance, quantified by music similarity. The founder of Pandora.com has presented the most used approach which calculates the distance between the source song and each of the database song. Each distance is regarded as a function of the differences between the musical features of the source song and database song. In addition, some hybrid similarity measures were also presented in recent work (e.g., [8, 13]), which combine music features with social tags. There are also some combined approaches which rank the search results by both their importance and relevance to the query. GenJam [14] proposed an interactive computer system improvising over a set of jazz songs using genetic algorithms. Douglas Turnbull, Luke Barrington, etc in [10] presented a computer audition system that can both annotate novel audio tracks with semantically meaningful words and use a semantic query to retrieve relevant tracks form database of unlabeled audio content.

## 3. Overview of Music Search

Intelligent music search contains basic search and recommendation. The following figure 1 shows the whole process of music search and recommendation.

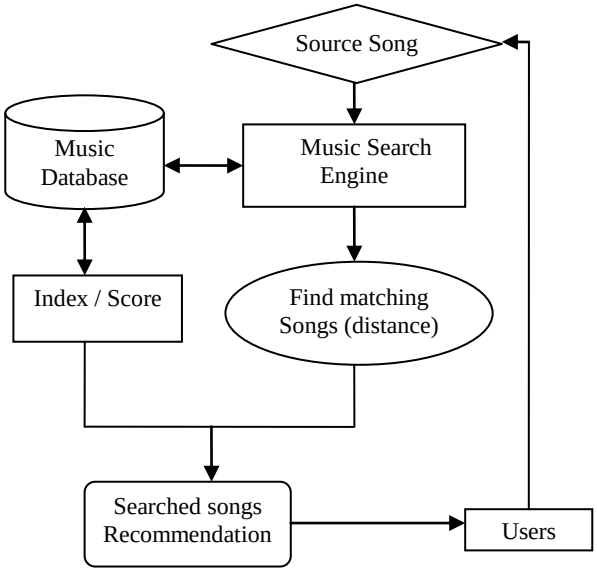



Figure 1, Overview of Music search

First of all, a music database has been built, which consists of two components: actually music songs and semantic index contents. The songs part contains the basic attributes of the music, and the indexing part contains four tables (music songs, songs with attributes, index term and index table). In addition, the recommendation will make prediction about what kind of music you are going to like next based on the search conditions and users feedback. More details will be described in the following sections.

## 4. Proposed Methodology

With the limitations of the current technologies, it's very difficult to extract the semantic content of multimedia data directly, especially for the music which is an art form. Thus indexing of these music data may become more costly and time-consuming. Therefore, this paper shall employ a novel music index approach to better support music search and retrieval.

### 4.1. Music Representation

The first task of music search is music representation. Without regard to the format of digital audio music, just think about the attributes and characteristics of music songs. Therefore, a given music song M is represented by an n-dimensional vector, which contains many different attributes (approximately 300 - 400). $M = <$ music_id, $e_1$, $e_2$, $e_3$, …, $e_n >$. Each attribute stand for a vector element, which is a characteristic of music songs. Suppose an n-dimensional music database vector which corresponds to the musical characteristics of a source song is determined. According to the Music Genome Project, a lot of music attributes are sorted by categories, which used for classifying music songs, for example Hip-Hop/R&B, Rock/Pop, Jazz/Blues, Country/Folk, etc.

By reference, Rock and Pop songs have 150 attributes; Rap songs has 350 attributes; And Jazz songs have approximately 400 attributes. Thus, these sufficient numbers of attributes have enough used to represent a given music. Each attribute is assigned a magnitude number which is between one and five, in have-integer increments every time. Thus, these attributes have been digitized. Therefore, the simple distance between any two songs in an n-dimensional space is able to be calculated by the Euclidean distance. As the different attribute may have different weight, thus the music song can be added a weighting vector $W = < w_1, w_2, w_3, ..., w_n >$. The sample formulation is in the following.

Music Representation:

Music Song $S = < s_1, s_2, s_3, \ldots, s_n >$ ;
Song Attribute Weight $W = < w_1, w_2, w_3, \ldots, w_n >$;
Where $1 <= s_n <= 5$, and $0 <= w_n <= 1$.

### 4.2. Indexing

Indexing is the process of collecting, parsing, and storing data to facilitate fast and accurate information retrieval. The index describes partial or whole content of the documents in one way or another. Index term may be related keywords, phrase which are meaningful, and it can be a well-defined hierarchy structures. Thus, without regard to the metadata of the music, for example song name, artist, album, the paper focus on the semantic content of the music data, for example style, mood, etc., which is more challenging than indexing metadata. The following part will describe the semantic content of music indexing approach in detail.

#### 4.2.1. Semantic Indexing Approach

The proposed music indexing approach is different from traditional text indexing approaches. Let's consider a collection of music songs { $S_j$ }, where their semantic features and characteristics cannot be directly and automatically extracted, it has been determined manually, as what Music Genome Project has done. Then every music song links with an index set { $I_j$ }, and this set contains a number of elements $e_{j1}$, $e_{j2}$, $e_{j3}$, …, $e_{jm}$. And each element is made up of a triple which has three components: song_id, index term, and index score. The index score stands for the signification of the triple of the index set to that music song. If the index score is higher, the index term to that music song is more important. According to the music representation and index representation, the following relations show their whole representation and relationship.

Music M = < music_id, music_name, description >
Song S = < song_id, $attribute_1$, $attribute_2$, ..., $attribute_n$ >
Index Term I = < index_id, index_term_name >
Index Table T = < index_id, music_id, score >

Thus the index table has a many-to-many relationship between the music song and the index term. In order to effectively and efficiently index all the music songs in the database, hierarchy the built index is required. Let's layer the index according to the different intervals of the score, thus there are N levels $L_1$, $L_2$, …, $L_n$ with a set of parameters $P_1$, $P_2$, …, $P_n$. Therefore, for the given score value x, if $P_i \leq x \leq P_{i+1}$, the given index term with score value x will be assigned to level $L_i$. According to this rule, all the index term will be placed in the suitable layer.

#### 4.2.2. Index Score Updating

Though the music index has already built in the above, the operations (add, delete, update) of the music index are also required. Because the index score is directly affected by the user search behavior and feedback, modified reinforcement learning algorithm is able to update music index score. Reinforcement learning is to maximize some notion of long-term reward. Here is to get accurate music search result. According to the SQL query, the query score for a music song can easily get from the index table.

Suppose when user input search terms Q($T_1$, $T_2$, $T_3$), music search engine will display the suggested N music songs ($m_1$, $m_2$, …, $m_n$) result in descending order by relevance (corresponding score $s_1$, $s_2$, …, $s_n$). Then the users will choose the result and give some feedback. When the user chooses the $m_i$ in the result list, the $s_i$ will be strengthened by $\alpha_1$. In addition, if the user provides the promising comments, the $s_i$ will be strengthened by $\alpha_2$. Conversely, if the user does not choose any song from the search result list, the related index score on $T_1$, $T_2$, and $T_3$ for the related songs will be decreased by $\beta_1$, with the different probability $(1 - \varepsilon)$ . Furthermore, if the negative feedback will be given to the engine, the penalty will be performed on the related songs in the database by different level $\gamma$. The following if statements show the above structure and calculations.

Algorithm 1: Updating Index Score

```
1.  If ( choose the mi in the result list) {
2.      Mi.score = Mi.score + α1
3.      If ( positive feedback ) {
4.          Mi.score = Mi.score + α1 + (1 – γ) * α2
5.      }
6.      Else {
7.          Mi.score = Mi.score + α1 - (1 – γ)
8.      }
9.  Else
10.     Mi.score = Mi.score – (1 – ε) * β1
```

#### 4.2.3. Incremental and Decreased Index

Apart from updating the music index, in order to maintain the index become more comprehensive and complete, adding and deleting some index terms are necessary. As we all know, if we give more search terms (index term) in the query, the more accurate search results will be gain. Let's consider this situation: when a user input a symphony "Eine kleine Nachtmusik", which is a very famous song composed by Mozart. However, the index table only maintains the several index terms, which don't contain the composer of the song. Thus when the users search this song by inputting Mozart, they may be not able to find this song in the return list. Therefore, adding the new index term "Mozart" to the index table is necessary.

Because the index is hierarchy built, properly assigning the score to the new index and placing it to the suitable hierarchy is very important, which affects the future search results. Suppose a new index term is added, the score of the related music song to the new index term is initially related to the highest score ($S_{max}$) of that music song in the index table. Let's give a level factor $\mu$ ($0 \leq \mu \leq 1$) to the added index term. Thus, the initially index score to the specific song can be gained by $(1 - \mu) * S_{max}$. Thus, the new index has been properly added, with the future continuously updating this index score, the search result by this index term will become more accurate.

In theory, the more indexes, the better search results. However, consider the time cost, keeping a proper number of indexes to a specific music is promising. Sometimes some existed index term may have a very low relevance with some music. Thus, in order to decrease the complexity of the computation and speed up search process, deleting those index terms and some records in the index table are required, which will keep the database complete and flexible. Suppose an existed record in index table is deleted, the delete criteria is related to the highest score ($S_{max}$) of that music song in the index table. Let's give a threshold factor $\Theta$ ($0 < \Theta \leq 0.05$), if the index score of the specific music song is smaller than $\Theta * S_{max}$, those records satisfying the above condition will be delete. Therefore, according to adding and deleting index terms or records, the music database and search engine will maintain an efficient and effective performance.

Algorithm 2: Increasing and Deleting Index Term

```
1.  If (add new index term)
2.      mj.score = (1 - μ) * Smax
3.  If (delete index term or index table record)
4.      While (mj.score < Θ * Smax)
5.          Delete (Ij)
```

### 4.3. Matching

Though the above indexing of music is able to successfully retrieve music, it focuses on the semantic music content. In order to consider more music attributes or characteristics, the paper has used a hybrid matching approach to find the similar songs. According to the music representation, considering given a song $S = < s_1, s_2, s_3, \ldots, s_n>$ and a song $T = <t_1, t_2, t_3, \ldots, t_n>$ , and each attributes of these vectors have been assigned a number between zero and five, in have-integer increments for every value. Thus, the simple distance between these two songs in n-dimensional space can be calculated by the following formulation:

$$\text{Distance}(S, T) = \text{sqrt}[\ \sum(s_i - t_i)\sim 2\ ], \text{ for } i = 1 \text{ to } n \quad (1)$$

When consider the weight $W = < w_1, w_2, w_3, \ldots, w_n >$ of these attributes of music songs, where $1 < = s_n <= 5$, and $0 <= w_n <= 1$, the revised distance can be calculated as follows:

$$\text{Distance}(S, T) = \text{sqrt}[\ w_i * \sum(s_i - t_i)\sim 2\ ], \text{ for } i = 1 \text{ to } n \quad (2)$$

In order to provide the user a flexibly control to the matching behavior, customizing some search conditions of the music attributes is required, which may be used to re-weight the song of the above matching approach and refine some searches for matching songs to include or exclude some selected attributes. Suppose when a user choose to modify the weighting vector, for example, increasing some weights of the attributes which are specific to the selected conditions, to complete particular matching result. Thus, the new resulting songs will be resembled closely to the source song in the new selected conditions.

### 4.4. Recommendation

Recommendation attempts to recommend information items (music, etc.) that are likely to be interest to the user. So they give us what they think we want, based on what we and other people like us have wanted in the past experience. Most current existent recommendation engines work backward instead, using information that comes not from the art but from their customers or audience, and they work on the principle that the behavior of a lot of people can be used to make educated guesses about the behavior of a single individual. Here is the idea: if most people who liked "The Second Waltz" also like "Medley Strauss And Co", then if we know that a particular individual liked "The Second Waltz", we can make an educated guess that that individual will also like "Medley Strauss And Co". This technique called collaborative filtering. Take music recommendation for example, the key point to grasp about collaborative filtering is that it knows absolutely nothing about the music, which has no preconceptions, and it works entirely on the basis of the audience's reaction. However, this mechanism may result some problem: when most people claim to have enjoyed "The Second Waltz" and also liked "Eine Kleine Nachtmusik K.525", the recommendation engine would be forced to infer that those two songs share some common quality that the users liked. Collaborative filtering works only as well as the data it has available, and humans produce noisy, low-quality data. Therefore, collaborative filtering has to improve to get accurate predictions. So the recommendation engine should use the information not only from the audience but also from the Intrinsic of art.

According to above song to song matching approach, the computation of the distance of the song in the same

category can be calculated. Let's give a threshold σ, all the distance smaller than σ will be recommended to the users. By improving above matching approach, multi-song matching approach can be used to strength recommendation. It builds functionality that will return the best matches to a group of source songs. This functionality provides a list of songs which are similar to the collection of an artist or album. Thus it will generate recommendations for the users, purely on their taste, without any source songs. Let's group the songs into a single virtual song, and the virtual center is defined to be a song vector whose attributes are the arithmetic average of the songs in the collection. Then associating this center a deviation vector represents the distribution of the songs within the collection. The following figure 2 shows the process of multi-song matching. So an individual attribute which has narrow distribution of values around the average value will have a strong affinity for the center value. Therefore, the small deviation will be assigned higher weight.

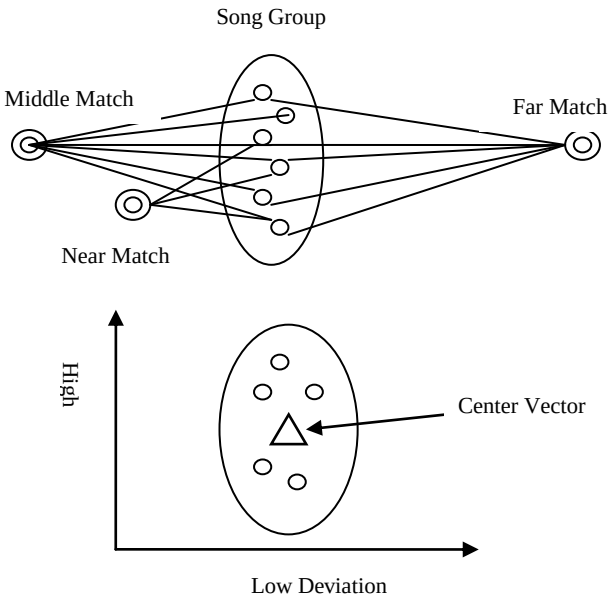



Figure 2, Multi-song Matching

Suppose the center vector of the selected song group $C = < c_1, c_2, \ldots, c_n >$, the standard deviation vector of the song group $D = < d_1, d_2, \ldots, d_n >$, thus the distance between the target vector to the center vector is in the following formulation:

$$\text{Distance}(S, T) = \text{sqrt}\left[ \sum( 1 / d_i ) \sim 2 \ * (c_i - t_i) \sim 2 \right] \qquad (3)$$

Where i from 1 to n; thus the smallest distance are the best matches, thus those songs with smallest distance will be recommended to the users.

### 4.5. Browsing

In order to make users conveniently find what they want in the search result list, which music songs to present to the user and in which order is related to browsing. Ranking music data by relevance and importance has been proposed by Maria M.Ruxanda, etc. in [12]. According to the previous music indexing, the simple approach is to rank the result by the index score in descending order. Thus the high scored music songs in the index table are always ranked in the top of the result list. Because the users choosing the top music list has a very large probability, the scores of these music songs have a great chance to be increased. Thus those new added songs which have great relevance to the search term as the above issue will be ranked lowly, which have affected the users' behavior. In addition, initially the search engine may be not work perfectly, thus the top search result may not what the users really want.

As the above issues, optimizing the search result will solve these problems. Genetic algorithm generates solutions to optimization problems using techniques inspired by natural evolution. Thus, by genetic algorithm, those related songs which ranked lowly will have a chance to place in a top result list, so the users will find those near hidden , actually important songs. Consider the above situation, let's give a probability to the each song Mi with a corresponding index score Si in the result list. Suppose there are N songs in the result list, according to the probability theory, initially the score of each returned song can be calculated by the following formulation:

$$P_i = S_i / \sum S_j \text{ , where i, j from 1 to N} \qquad (4)$$

Thus the songs with higher scores would have a higher probability. So those songs with higher probability would a higher rank in the search result list. After initializing, then it would search for those index terms, and then select acceptable songs from the search result list and mark down some index terms from it. This procedure would utilize a fitness function. Fitness evaluation is based on external feedback from the users. So do this until the results are approximately what you are really looking for, then stop do it if you are searching over and over for many times but you are not getting good results.

## 5. Observations

In order to best evaluate the suggested indexing and recommending approaches on music retrieving, some observations have to make. Initially, as the index term is limited, the search result not always works perfectly. With gradually adding new index term to the music database, the search result will be more accurate. Then though the

users continue to increase new indices, the accurate of search results tend towards an equilibrium level. When users set different search conditions, the more search terms input, the more accurate result gain. If the dimensional of music representation is very large, the songs will be effectively distinguished, while the computation may be very complex. On the contrary, it is not able to accurately recommend songs the user likes.

As improved ranking is a genetic algorithm it also suffers from the shortcomings that genetic algorithm has. First of all adaptive search depends on your first initial guess, which may become so unrelated that your search results are really bad. Secondly, it depends on your judgment of accepting a result. If you expect results too soon and accept no result you may end up unhappy. On the other hand, if you expect all results you may end up doing too many searches with all unrelated results.

## 6. Conclusions and Future Work

This paper presents a semantic indexing method that is capable of flexibly retrieval music in semantic level. In order to retrieve music accurately, first of all, music representation in a given song is represented by an n-dimensional vector, which contains many different attributes. Then four tables (music, songs, index term, index tables) have been established to semantic index. After that some algorithms and formulations of index score updating, increasing, and decreasing have been detailed explained. In addition, useful and flexible music recommendation approaches, for example song-to-song matching, and multi-song matching methods, has been detailed introduced to make better recommendations to the users. Finally, improved genetic algorithm is used to optimize the ranking of the search results list when users browse. Thus, the above proposed methods can be useful when incorporated into commercial music search engines for improving music service.

In future work, other new schemes that adopted in semantic indexing and retrieval areas will be investigated. In addition, new intelligent music recommendation which based on the intrinsic of musical art will also be studied, whether content-based filtering or collaborative filtering, with the popularity of the social media [11]. As well, the suggested music search engines will be constructed, and some experiments, evaluations and user feedbacks will be conducted, to evaluate user's satisfaction with respect to the proposed music search and recommendation methods.

## References


[1] C. H. C. Leung, J. Liu, W. S. Chan, and A. Milani. An Architectural Paradigm for Collaborative Semantic Indexing of Multimedia Data Objects. Proceedings of the 10th international conference on Visual Information Systems: Web-Based Visual Information Search and Management, Vol. 5188, pp.216 – 226, 2008

[2] Leung, C. and Liu, J. Multimedia data mining and searching through dynamic index evolution. In 9th proceedings of advances in Visual Information systems, Shanghai, China, pp.298-309, 2007.

[3] Gros, P., Delakis, M., and Gravier, G. Multimedia Indexing: The Multimedia Challenge. In ERCIM News No. 62, 2005.

[4] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic Annotation and Retrieval of Music and Sound Effects. Audio, Speech, and Language Processing, IEEE Transactions on Issue Date: Feb. 2008, Vol.16, pp. 467 – 476.

[5] Nicola Orio. Music Retrieval: A Tutorial and Review. Foundations and Trends in Information Retrieval, Volume1, Issue 1, Pages 1-96, 2006.

[6] Alexandros Nanopoulos, Dimitrios Rafailidis, Maria M. Ruxanda, Yannis Manolopoulos: Music search engines: Specifications and challenges. Inf. Process. Manage. 45(3): pp.392-396, 2009.

[7] Donald Byrd, Tim Crawford: Problems of music information retrieval in the real world. Inf. Process. Manage. 38(2): pp.249-272, 2002.

[8] R. Stenzel and T. Kamps. Improving content-based similarity measures by training a collaborative model, in Proc. ISMIR, 2005, pp. 2264-271.

[9] C. R. Buchanan Semantic-based audio recognition and retrieval, pp. 2005.

[10] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet "Towards musical query-by-semantic description using the CAL500 data set", Proc. SIGIR"07, pp. 439 2007

[11] Music Information Retrieval Using Social Tags and Audio, Levy, M. Sandler, M. Multimedia, IEEE Transactions on page(s): 383 - 395, Volume: 11 Issue: 3, April 2009

[12] Maria M.Ruxanda, Alexandros Nanopoulos, Christian S. Jensen, Yannis Manolopoulos. Ranking music data by relevance and importance. Multimedia and Expo, 2008 IEEE international conference, pp.549- 552, 2008.

[13] C. R. Buchanan J. Kandola, J. Shawe-Taylor, and N. Cristianini. Learning Semantic Similarity, in Proc. NIPS, 2002, pp.657-664.

[14] Biles, J.A. GenJam: a genetic algorithm for generation of jazz sols. In Proceedings of the International Computer Music Conference. Aarhus, Denmark, 1994.

[15] Pandora.cm. www.pandora.com

[16] Last.fm . www.last.fm

# A ROBUST LIP TRACKING ALGORITHM USING LOCALIZED ACTIVE CONTOURS AND DEFORMABLE MODELS

*Xin Liu and Yiu-ming Cheung*

Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China
{xliu,ymc}@comp.hkbu.edu.hk

**ABSTRACT**

Lip tracking is crucial to the success of a lip reading recognition system. This paper presents a robust lip tracking algorithm using localized active contours and deformable models. The proposed approach utilizes a combined semi-ellipse as the initial evolving curve, through which a separation of the original lip image into lip and non-lip regions can be found. Moreover, the dynamic parameter selection of local regions and a 16-point deformable model are obtained to achieve the finial tracking results. The proposed approach is adaptive to the movement of the lips from frame to frame, and also robust against the appearance of the teeth. Experiments show the promising results of our algorithm.

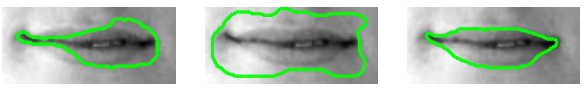
***Index Terms***— Lip tracking, localized active contour, deformable model

## 1. INTRODUCTION

In recent years, lip tracking has received wide attention in the community because of its potential applications in areas such as lipreading, audio-visual speech recognition and facial expression analysis. Nevertheless, it is a non-trivial task to track the lip movements due to the large variations caused by different speakers, noise, low contrast between lip and skin, teeth effect and so forth.

In the last decade, a few techniques have been proposed to realize the lip tracking with the focus on segmentation of lip regions or extraction of lip contours. Essentially, point-based approach and region-based approach are two main approaches for tracking lips from frame sequences. In the point-based method [1], a set of characteristic points are detected through the low level spatial cues such as color and edges, as well as a priori knowledge of the lip structure. However, such a method is somewhat sensitive and ambiguous to the initial position of feature points along the lip edges. In general, the region-based method can make the tracking results more robust and realistic. Typical examples include deformable template (DT) [2] and active contour model (ACM) [3]. The DT algorithm employs a cost function to partition a color lip image into lip and non-lip regions via a parametric model. Generally, the tracking performance of this method may deteriorate when there exists a poor contrast between lip and surrounding skin regions. The conventional ACM algorithm allows an initial closed curve to deform via minimizing a global energy, such that an object contour is obtained. However, this approach is sensitive to the choice of parameters and uneven illuminations. Recently, some researchers attempt to combine the merits of the above approaches, papers [4][5] have shown the desired tracking results in their application domain. Nevertheless, their performance will be affected by the teeth appearances. Further, these methods usually involve iterative complexity or stochastic optimization, which is quite time-consuming.

Moreover, almost all the region-based approaches utilize the global statistical characteristics. When images have heterogeneous statistics or complex components, it is found that the localized active contour models (LACM) [6] can generally achieve a better segmentation result, e.g., as shown in Fig.1(a) and (c). Nevertheless, this model is dependent on the appropriate selection of correlative parameters, otherwise, its performance will deteriorate as shown in Fig.1(b).



(a) (b) (c)

**Fig. 1**. Lip contour extraction with uneven illuminations. (a) Conventional ACM based extracting result, (b) LACM based extracting result with improper parameters, (c) LACM based extracting result with proper parameters.

In this paper, we propose a robust lip tracking algorithm using LACM and deformable model. We find a combined semi-ellipse as the initial evolving curve fitted in LACM to extract the lip contour of the first frame. Subsequently, we utilize the dynamic selections of local radius and a 16-point deformable model to achieve the final tracking results. Experimental results show the efficacy of our algorithm.

## 2. OVERVIEW OF LACM

This section overviews the framework in LACM [6], which assumes that the foreground and background regions are locally different. This framework utilizes the evolving curve to split the local regions into the local-interior and local-exterior , through which a group of local energies are constructed. The advantage of this framework is that the complex appearances of objects can be successfully segmented with localized energies when the corresponding global energies fail.

Let $I$ denote a pre-specified image defined on the domain $\Omega$, parameters $u$ and $v$ are expressed as independent spatial variables to represent a single point, individually. $C$ denotes a closed curve represented as the zero level set of a signed distance function $\phi$, i.e., $C = \{u \,|\phi(u) = 0\}$ [6]. 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 in the following is used to specify the area adjacent to the curve.

$$\delta\phi(u) = \begin{cases} 1, & \phi(u) = 0 \\ 0, & |\phi(u)| < \varepsilon \\ \frac{1}{2\varepsilon}\left\{1 + \cos\left(\frac{\pi\phi(u)}{\varepsilon}\right)\right\}, & otherwise. \end{cases} \quad (2)$$

Along the curve $C$, the characteristic function $\mathcal{B}(u, v)$ marked the local regions in terms of a radius parameter $r$ can be described as follows:

$$\mathcal{B}(u, v) = \begin{cases} 1, & \|u - v\| < r \\ 0, & otherwise. \end{cases} \quad (3)$$

Let $\mu_{in}(u)$ and $\mu_{out}(u)$ represent the intensity mean in local interior and exterior regions localized by $\mathcal{B}(u, v)$ at a point $u$, respectively.

$$\mu_{in}(u) = \frac{\int_{\Omega_v} \mathcal{B}(u, v) \cdot \mathcal{H}\phi(v) \cdot I(v)dv}{\int_{\Omega_v} \mathcal{B}(u, v) \cdot \mathcal{H}\phi(v)dv}, \quad (4)$$

$$\mu_{out}(u) = \frac{\int_{\Omega_v} \mathcal{B}(u, v) \cdot (1 - \mathcal{H}\phi(v)) \cdot I(v)dv}{\int_{\Omega_v} \mathcal{B}(u, v) \cdot (1 - \mathcal{H}\phi(v))dv}. \quad (5)$$

Subsequently, a localized region-based energy formed from the global energy [7] is obtained:

$$F = -(\mu_{in}(u) - \mu_{out}(u))^2. \quad (6)$$

By ignoring the image complexity that may arise outside the local region, only the contributions from the points within the radius $r$ of the contour are considered. Consequently, for the purpose of keeping the curve smooth, a regularization term is added in the local energies. In addition, the arclength of the curve is penalized and weighted by a parameter $\lambda$, and the final energy $\mathrm{E}(\phi)$ is given as follows:

$$\mathrm{E}(\phi) = \int_{\Omega_u} \delta\phi(u) \int_{\Omega_v} \mathcal{B}(u, v) \cdot F(I(v), \phi(v))dvdu + \lambda \int_{\Omega_u} \delta\phi(u) \|\nabla(u)\|du. \quad (7)$$

By taking the first variation of this energy with respect to $\phi$, the following evolution equation is obtained:

$$\frac{\partial\phi}{\partial t}(u) = \delta\phi(u) \int_{\Omega_v} \mathcal{B}(u, v) \cdot \nabla_{\phi(v)} F(I(v), \phi(v))dv + \lambda\delta\phi(u)div\left(\frac{\nabla\phi(u)}{|\nabla\phi(u)|}\right) \|\nabla\phi(u)\|. \quad (8)$$

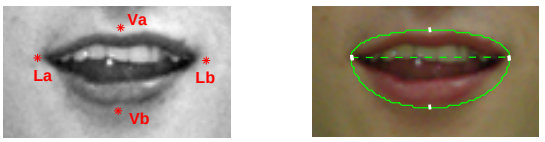
It is noteworthy that almost all the region-based segmentation energies can be put into this framework.

## 3. THE PROPOSED ALGORITHM

Our proposed lip tracking algorithm mainly includes the two steps: (a) lip contour extraction for the first frame, (2) lip tracking in the following frames.

### 3.1. Lip Contour Extraction

In LACM, the initial evolving curve $C$ and the radius $r$ of the local region are two crucial parameters. As the uneven illuminations and the teeth appearance always exist during the speech, improper parameters such as far away evolving curve, large radius may cause inaccurate result as shown in Fig.1(b).




(a) Lip corner dots (b) Combined semi-ellipse

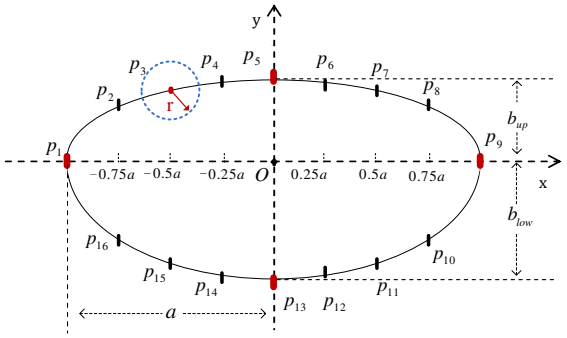
**Fig. 2**. A combined semi-ellipse around the lip.

Empirical studies have found that a lip shape can be approximatively surrounded by a combination of two semi-ellipses, which can be used as the initial evolving curve fitted in LACM. According to the previous work [4] [8], the primary lip corner dots have been successfully detected. We denote the left corner, right corner, up corner and down corner as $La, Lb, Va$ and $Vb$, respectively. Let $(x_c, y_c)$ be the origin center of the combined semi-ellipse, through which the mathematical equations can be described in the following:

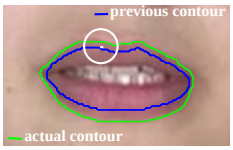
$$x_c = \frac{1}{2}(La_x + Lb_x), y_c = \frac{1}{2}(La_y + Lb_y),$$

$$\theta = \arctan\left(\frac{Lb_y - La_y}{Lb_x - La_x}\right),$$
$$a = \frac{1}{2}((Lb_x - La_x)^2 + (Lb_y - La_y)^2)^{\frac{1}{2}},$$
$$b_{up} = ((Va_x - x_c)^2 + (Va_y - y_c)^2)^{\frac{1}{2}},$$
$$b_{low} = ((Vb_x - x_c)^2 + (Vb_y - y_c)^2)^{\frac{1}{2}},$$
$$X = (x - x_c)\cdot\cos\theta + (y - y_c)\cdot\sin\theta,$$
$$Y = (y - y_c)\cdot\cos\theta - (x - x_c)\cdot\sin\theta,$$
$$\frac{X_{up}^2}{a^2} + \frac{Y_{up}^2}{b_{up}^2} = 1, \frac{X_{low}^2}{a^2} + \frac{Y_{low}^2}{b_{low}^2} = 1, \quad (9)$$

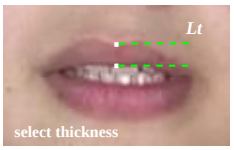
where $a$ is the semi-major axes, $b_{up}$ and $b_{low}$ are the up and low semi-minor axes, respectively. $\theta$ is the inclined angle, which is positively defined in the counter-clockwise direction. Consequently, we let the combined semi-ellipse be the initial evolving curve fitted in LACM. Meanwhile, as a rule of thumb, $r = \frac{b_{up}}{2}$ is appropriate to extract the lip contour in this step.




(a) A 16-point lip model




(b) Opening mouth




(c) Upper lip thickness

**Fig. 3**. Proper parameters in our algorithm.

In addition, the extracted lip contours usually exist unsmoothness, as well as the slightly lack of the geometric lip shapes. Therefore, we employ a 16-point geometric deformable model [5] with cubic spline interpolation to model the lip shape as shown in Fig.3(a) , which is always physically meaningful in tracking applications.

### 3.2. Lip Tracking

As speaking usually includes many frames in a second, which can be regarded as a group of continuous sequences. Additionally, a lip shape of one frame changes a little compared with the adjacent one. After accurately extracted the lip contour of the first lip frame, we can use it as the initial evolving curve fitted in LACM to the next frame .

It is noteworthy that, the lip movements, especially in the process of opening a mouth, the lip contour of previous one may inside the current one. As is shown in Fig.3(b), to avoid the effects of teeth appearance, the dynamic parameter selection is proposed in LACM. We can easily compute the middle thickness $Lt$ of the upper lip by the variations of pixel value along the line segment $OP_5$ as shown in Fig.3(c). Therefore, we employ the dynamic parameter $r_i$, which can be fitted in LACM as the local radius for tracking the lip movements, i.e.,

$$r_i = \frac{Lt_{i-1}}{2}, i \geq 2, \quad (10)$$

where $Lt_{i-1}$ expresses the middle thickness of the upper lip in the previous one. We let N denote the total number of points on the evolving curve. The pseudocode for lip tracking algorithm is given as follows:

**Input**: $I_i \in \Omega, C_{i-1}$( *The previous lip contour* ).
**Output**: $C_i$ ( *The tracking contour* ).
Lip image preprocessing;
**while** *minimize the local energy is not met* **do**
  **for** $(j = 1; j \leq N; j++)$ **do**
    Let $C_{i-1}$ be the initial evolving curve;
    Assign proper $r$ to the $\mathcal{B}(u, v)$;
    $\delta\phi(u_j) = \delta\phi(u_j) + \frac{\partial\phi}{\partial t}(u_j)$;
  **end**
  **if** *all* $E(\phi(u_j)) < \varepsilon, j \in [1, \cdots, N]$ **then**
    $C_i = \{u \,|\phi(u) = 0\}$;
    Obtain the tracking result;
  **else**
    Go back to the beginning;
  **end**
**end**

**Algorithm 1**: The tracking algorithm

## 4. EXPERIMENTAL RESULT

The algorithm has been implemented on an Intel® Core™2 Quad Q9450 2.66 GHz machine and applied with Matlab 7.0 image processing toolkit. We project the RGB lip images into the gray-level space, each lip image is performed with a $3 \times 3$ mean filter and a contrast stretching adjustment. In our experiments, we set the parameter $\lambda$ is equal to $0.3$.

### 4.1. Experiment 1

We have applied our lip contour extraction approach to the 200 frontal face images from the CVL face database and 300 face images from our laboratory database.

Examples of lip contour extraction are shown in Fig.4. It can be seen that the accurate lip contours can be extracted using the proposed algorithm. In our experiments, more than 95% of the test database have a satisfactory result. We have

also examined the unsatisfactory ones and found that they are all have the very poor contrast between the lip and skin region, or have obvious beard effects around the lips.

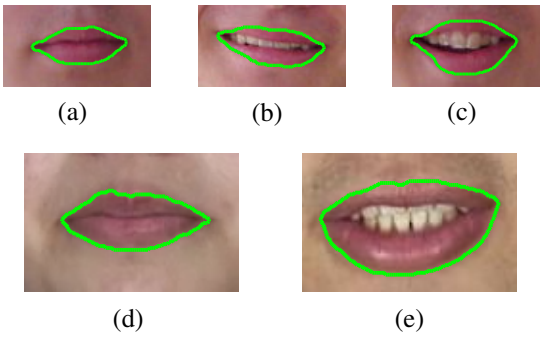



**Fig. 4**. LACM based extracting results. (a)(b)(c) from the CVL database and (d)(e) from our laboratory database.

### 4.2. Experiment 2

We have performed the proposed lip tracking algorithm on a large number of face sequences captured from 10 speakers, who were speaking English and Chinese digits (0-9) in a uniform illuminance environment. Each second records 20 colorized face frames and the located lip image of size $116 \times 72$ from the face sequences.

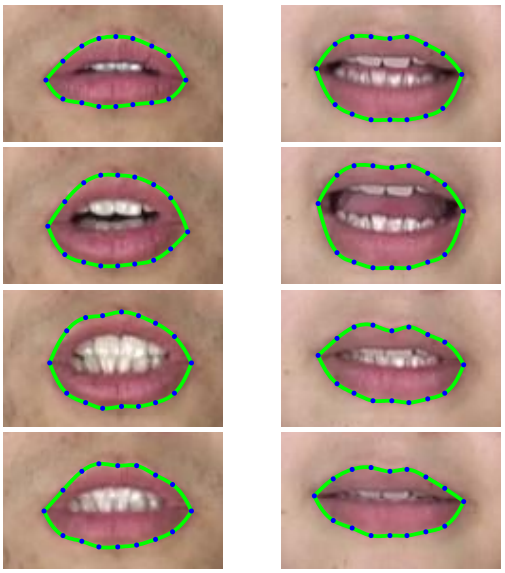


**Fig. 5**. Tracking result using the proposed method.

**Table 1**. Computing time of the proposed algorithm.

| Algorithm Step | Extracting | Tracking |
|---|---|---|
| Iteration [average] | 26 | 5 |
| Computing time[average] | 0.374s | 0.073s |

Tracking results are shown in Fig.5 and Table 1. It is found that the proposed algorithm has a promising tracking result, which is robust against the teeth appearance. Meanwhile, the use of a 16-point deformable model to describe a lip shape is physically meaningful. In addition, the computing time of tracking one lip frame is less than the extracting process. When there exists a large lip sequences, it is effective to utilize the previous lip contour as the initial evolving curve of the proceeding one, which can reduce a large amount of computing time. From the results, our approach is feasible and effective.

## 5. CONCLUSION

In this paper, we have proposed an algorithm to track the lip movements using localized active contours and deformable models. This algorithm is adaptive to the lip movements, as well as robust against the appearance of teeth effect, it is very suitable for applications that require a high level of accuracy such as lip reading or speech recognition.

## 6. REFERENCES

[1] M.U. Ramos Sánchez, J. Matas, and J. Kittler, "Statistical chromaticity-based lip tracking with b-splines," in *Proc. of ICASSP*, 1997, vol. 4, pp. 2973–2976.

[2] A.W.C Liew, S.H. Leung, and W.H Lau, "Lip contour extraction from color images using a deformable model," *Pattern Recognition*, vol. 35, no. 12, pp. 2949–2962, 2002.

[3] G.I. Chiou and J.N Hwang, "Lipreading from color video," *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1192–1195, 1997.

[4] N. Eveno, A. Caplier, and P.Y. Coulon, "Accurate and quasi-automatic lip tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 706–715, 2004.

[5] S.L. Wang, W.H. Lau, and S.H. Leung, "Automatic lip contour extraction from color images," *Pattern Recognition*, vol. 37, no. 12, pp. 2375–2387, 2004.

[6] S. Lankton and A. Tannenbaum, "Localizing region-based active contours," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2029–2039, 2008.

[7] 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*, vol. 13, pp. 195–216, 2002.

[8] M. Li and Y.M. Cheung, "Automatic lip localization under face illumination with shadow consideration," *Signal Process*, vol. 89, no. 12, pp. 2425–2434, 2009.

# Learning the Relationship between High and Low Resolution Images in Kernel Space for Face Super Resolution*

Wilman W.W. Zou and Pong C. Yuen
*Department of Computer Science, Hong Kong Baptist University*
{*wwzou, pcyuen*}*@comp.hkbu.edu.hk*

## Abstract

*This paper proposes a new nonlinear face super resolution algorithm to address an important issue in face recognition from surveillance video namely, recognition of low resolution face image with nonlinear variations. The proposed method learns the nonlinear relationship between low resolution face image and high resolution face image in (nonlinear) kernel feature space. Moreover, the discriminative term can be easily included in the proposed framework. Experimental results on CMU-PIE and FRGC v2.0 databases show that proposed method outperforms existing methods as well as the recognition based on high resolution images.*

## 1. Introduction

Face recognition from surveillance camera has wide range of applications, ranging from single standalone camera domestic application to multiple network camera law enforcement [11]. In all these applications, camera is normally installed in a way that viewing area is maximized and the face region is very small. Moreover, the person to be recognized is normally not cooperative. Therefore, to recognize a face from surveillance camera, we need to handle low resolution face image with variations, such as pose, illumination and occlusion.

Super-resolution (SR) for face image (face hallucination [1]) algorithms have been proposed to enhance the resolution of the face image for recognition [3] [4]. Theoretically, applying SR technique on the low-resolution (LR) face image, the reconstructed high-resolution (HR) image can be used for face recognition. This approach works well only if the input face image is frontal and captured under good illumination.

In order to normalize the face image variations to frontal view before applying the SR method, a two-step approach has been proposed. Li and Lin [7] made use of a view-based model to normalize the pose variations before the SR procedure. Jia and Gong [5] synthesized the HR images with different poses, lighting conditions and expressions based on tensor space. 3D-model are also adopted for handling the face variations. Mortazavian *et al.* [9] employed the 3DMM model to synthesis different pose images. Yu *et al.* [10] modeled pose and illumination using bilinear function, and employed pose-illumination-based SR method to reconstruct images. However when the face image resolution is smaller than 20x20 pixels, the results of these normalization algorithms may not be satisfactory.

To overcome the problems in face recognition from video, this paper proposes a kernel based face super resolution algorithm. With the highly complex and nonlinear face image variations, the relationship between LR and HR images will be nonlinear. It is well-known that the nonlinear kernel mapping could transform complex distributed data into high dimensional feature space where the data becomes linear separable. In this way, by conducting the relationship learning in kernel feature space, their nonlinear relationship can be learnt. Details are discussed in next section.

## 2 Proposed Super Resolution with Face variations

Figure 1 shows the block diagram of the proposed method. The proposed method consists of two phases, namely training and recognition. In training phase, given a set of LR and HR image pairs, both images are mapped to the kernel feature space by nonlinear mapping $\Phi_L$ and $\Phi_H$. The nonlinear relationship between HR and LR images can be better modeled after mapping. Then, we would like to learn their relationship ($R^{**}$) for super resolution from recognition perspec-

*This work has been accepted by ICPR 2010

tive. A kernel subspace based regression relationship learning is designed to estimate the relationship operator $R^{**}$. Also, a discriminative cost function is designed and easily integrated with the learning processing, so that the relationship learning can better handle the face variations. In recognition phase, given the query image, its HR image representation in kernel space can be generated using $R^{**}$. Then, any kernel-based face recognition methods, such as Kernel PCA [6] and Kernel Direct Discriminative Analysis (KDDA) [8] can be employed for recognition. In the following, we will first report our proposed framework. The key issues in the proposed method namely, learning the relationship $R^{**}$ and designing the discriminative term, will be followed.

### 2.1 Relationship Learning Framework for Super Resolution

Let $I_h$ and $I_l$ be the high-resolution (HR) image and low-resolution (LR) image respectively. We assume that $I_l$ and $I_h$ have been aligned, so

$$I_l = DI_h + n \tag{1}$$

where $D$ is the downsampling operator, and $n$ is noise. To reconstruct the HR image from the corresponding LR one, we have proposed a novel framework to learn this relationship [13]. Suppose the relationship between HR and LR image is $\mathcal{R}$. That means for each LR image, we have

$$\tilde{I}_h = \mathcal{R}(I_l) + n \tag{2}$$

Therefore after determining the relationship $\mathcal{R}$, the HR image can be easily reconstructed by Eq.(2). To evaluate $\mathcal{R}$, the error between the reconstructed HR image and the original HR image should be minimized as follows,

$$\mathcal{R} = \arg\min_{\mathcal{R}'} \sum_{i=1}^{N} \|\mathcal{R}'(I_l^i) - I_h^i\|^2 \tag{3}$$

where $(I_l^i, I_h^i)$ is the $i$-th training image pair, and $N$ is number of image pairs in training data set. This framework provides higher flexibility and can easily integrate a discriminative term to enhance the discriminability of the reconstructed image (to be discussed in Section 2.3).

### 2.2 Discovering the Nonlinear Relationship

Because of the nonlinear face variations, the relationship between LR images and HR images may be nonlinear and complicated. There does not exist closeform solution for finding $\mathcal{R}$, so that estimating $\mathcal{R}$ by Eq.(3) is not feasible. In this paper, we employ the nonlinear mapping $\Phi$ to map the original image to the high dimension feature space, as shown in Figure 1. In this feature space, the relationship between HR and LR features can be better modeled by linear approximation. That means in this feature space, the relationship operator $\mathcal{R}$ can be approximated by a matrix $R$. Following Eq.(3), $R$ can be estimated by

$$R = \arg\min_{R'} \sum_{i=1}^{N} \|R'\Phi_L(I_l^i) - \Phi_H(I_h^i)\|^2 \tag{4}$$

Let $\mathbf{E}_H = \{e_H^1, e_H^2, \cdots\}$ and $\mathbf{E}_L = \{e_L^1, e_L^2, \cdots\}$ are the orthonormal bases of HR image feature space and LR image feature space. So we have:

$$\begin{aligned}
&\min \sum_{i=1}^{N} \|R\Phi_L(I_l^i) - \Phi_H(I_h^i)\|^2 \\
=&\min \sum_{i=1}^{N} \|R \sum_{j} < \Phi_L(I_l^i), e_L^j > e_L^j \\
&- \sum_{k} < \Phi_H(I_h^i), e_H^k > e_H^k\|^2 \\
=&\min \sum_{i=1}^{N} \|R\mathbf{E}_L f_L^i - \mathbf{E}_H f_H^i\|^2 \\
=&\min \sum_{i=1}^{N} \|\mathbf{E}_H^{-1} R\mathbf{E}_L f_L^i - f_H^i\|^2
\end{aligned} \tag{5}$$

where $f_L^j$ is the weight (coefficient) of image $I_L^j$ in feature space. Let $R^* = \mathbf{E}_H^{-1} R\mathbf{E}_L$, to determine $R$ is equivalent to determine $R^*$. And we have

$$R^* = \arg\min_{R'} \sum_{i=1}^{N} \|R' f_L^i - f_H^i\|^2 \tag{6}$$

However, it is very computationally expensive to calculate the nonlinear mapping $\Phi$ explicitly due to the high dimensionality of the feature space, so it is not feasible to calculating the relationship $R^*$ by Eq.(6) directly. Kernel trick is used and kernel subspace is used to represent the features in the kernel feature space. Let $\mathbf{S}_H = \{S_H^1, S_H^2, ..., S_H^{k_1}\}$ and $\mathbf{S}_L = \{S_L^1, S_L^2, ..., S_L^{k_2}\}$ represent the the kernel subspaces for HR image and LR image space respectively. Replace $\mathbf{E}_L$ and $\mathbf{E}_H$ with $\mathbf{S}_L$ and $\mathbf{S}_H$, we have

$$R = \arg\min_{R'} \sum_{i=1}^{N} \|S_H' R' S_L \hat{f_L^i} - \hat{f_H^i}\|^2 \tag{7}$$

where $\hat{f_L^i}$ and $\hat{f_H^i}$ are the LR and HR kernel subspace coefficients for representing the image features, respectively. Under this subspace representation form, we

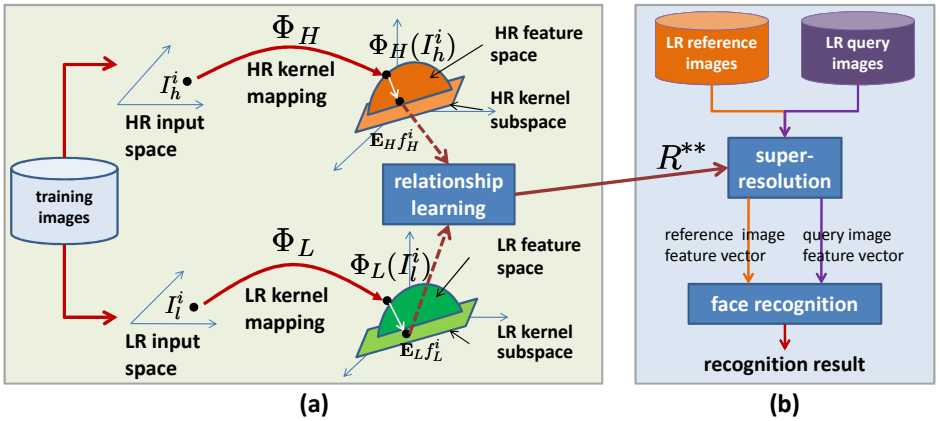



**Figure 1. Block diagram of proposed method: (a) the relationship training phase, (b) face recognition phase.**

learn the relationship between coefficients instead of features, so Eq.(6) becomes

$$R^* = \arg\min_{R'} \sum_{i=1}^{N} \|R' \hat{f}_L^i - \hat{f}_H^i\|^2 \tag{8}$$

Given the LR query image, the HR image kernel subspace features can be reconstructed by

$$\hat{f}_H = R^* \hat{f}_L \tag{9}$$

After reconstructing the HR image kernel features, we can directly make use of those features for recognition using existing kernel based face recognition engine. Also, if the HR image is required, pre-image learning [12] can be adopted.

### 2.3 Discriminative Relationship Learning

Discriminability of the reconstructed HR image is important for recognition purpose. To achieve that, a discriminability cost function is designed so that the reconstructed HR images (features) are suitable for recognition.

A good way to enhance the discriminability is to make use of the label information of the training data. 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. Inspired by the success of Maximum Margin Criterion, a discriminability cost function is developed as follows,

$$\begin{aligned} d(R) = &\sum_{k=1}^{K} \sum_{\hat{f}_L^i \in \Omega_k} \|R\hat{f}_L^i - \bar{\hat{f}}_H^k\|^2 \\ &+ \sum_{k=1}^{K} n_k \|R\bar{\hat{f}}_L^k - \bar{\hat{f}}_H\|^2 \end{aligned} \tag{10}$$

where $\bar{\hat{f}}_L^k$ ($\bar{\hat{f}}_H^k$)is the mean of the features of LR (HR) images from class $\Omega_k$, while $\bar{\hat{f}}_H$ is the mean of the features of all HR training images; $n_k$ is the number of training images in class $\Omega_k$, while $K$ is the number of classes.

So the discriminative relationship learning can be conducted by minimizing the following equation.

$$R^{**} = \arg\min_{R'} \sum_{i=1}^{N} \|R' \hat{f}_L^i - \hat{f}_H^i\|^2 + \alpha d(R') \tag{11}$$

where $\alpha$ is weight to balance the reconstruction error and the discriminability cost function.

## 3. Experimental Results

KPCA [6] and KDDA [8] face recognition algorithms, as well as CMUPIE and FRGC V2.0 face databases are used for experiments. The resolution for the HR images and LR images are 56x64 and 14x16 respectively. To obtain the super-resolved images / features, the proposed super resolution algorithm is applied to LR image. Hallucination Face [1] (HF) and kernel-based face hallucination (KF) method [2] are also employed. The recognition algorithms are also applied to

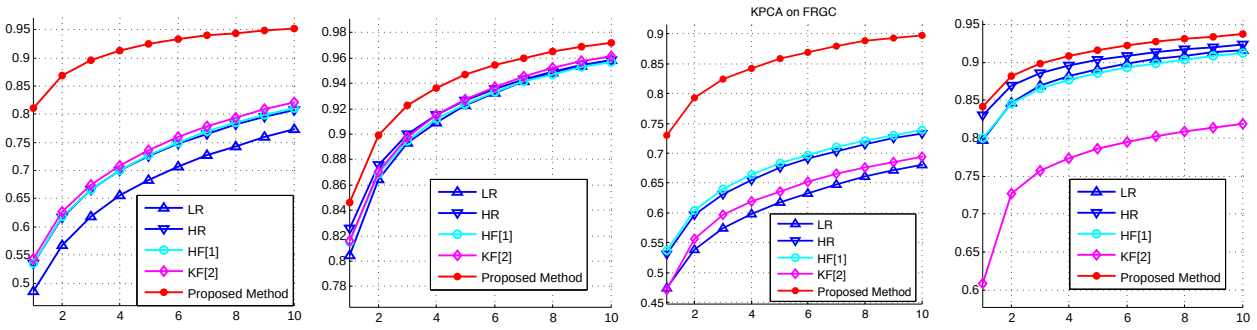



**Figure 2. The CMC Curves of different face recognition methods on CMUPIE and FRGC database: (a) KPCA on CMUPIE (b) KDDA on CMUPIE (c) KPCA on FRGC (d) KDDA on FRGC**

both HR and LR images and the results are used for comparison.

To estimate the effectiveness of the proposed method on dealing with nonlinear face variations, the face images used in our experiments contains face variations due to different poses, lighting conditions and expressions. In CMUPIE database, 21 lighting conditions and 5 poses are covered for each person (total 68 persons); in FRGC database, the training dataset is used, containing 12776 images (222 persons) with different pose, illumination and expression or a combination. Each database is divided into two non-overlapped sets. 10 images per person are randomly selected as training data which are used for learning the relationship $R^{**}$ in Eq.(11) and the reference template while the rest of images are used as testing data.

The recognition results in terms of CMC curves are reported in Figure 2. It can be seen that, in general, the recognition based on HR images outperforms that of LR images. Recognition accuracy based on HR image generated by Hallucination Face (HF) method is close to that of HR images, which implies that HF method is good. Kernel-base face hallucination (KF) method, also performs better than LR in most cases. In all cases, recognition using the the HR image features generated by our proposed method gives the best results, which implies that the proposed method could handle both low resolution and non-linear variations well.

## 4 Conclusion

The problem of face hallucination with nonlinear face variations is discussed in this paper. To solve this problem, we have proposed a new face hallucination framework, which discovers the nonlinear relationship between the LR images and HR images with nonlinear face variations. The experimental results show that the proposed method significantly enhances the recognition performance.

**Acknowledgement** This project is partially supported by Science Faculty Research grant of Hong Kong Baptist University and NSFC-GuangdDong research grant U0835005.

## References

[1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *PAMI*, 24(9):1167–1183, 2002.

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

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

[4] P. H. Hennings-Yeomans, S. Baker, and B. Kumar. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. In *IEEE Intl. Conf. on CVPR*, pages 1–8, 2008.

[5] K. Jia and S. Gong. Generalized face super-resolution. *IEEE Trans. on Imag. Proc.*, 17(6):873–886, 2008.

[6] K. Kim, K. Jung, and H. Kim. Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 9(2):40–42, 2002.

[7] Y. Li and X. Lin. Face hallucination with pose variation. In *6th IEEE Intl. Conf. on AFGR, 2004. Proceedings*, pages 723–728, 2004.

[8] J. Lu, K. Plataniotis, and A. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. on Neural Networks*, 14(1):117–126, 2003.

[9] P. Mortazavian, J. Kittler, and W. Christmas. 3D-assisted Facial Texture Super-Resolution. In *BMVC*, 2009.

[10] J. Yu, B. Bhanu, Y. Xu, and A. Roy-Chowdhury. Super-resolved facial texture under changing pose and illumination. In *Intl. Conf. on Image Processing*, 2007.

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

[12] W.-S. Zheng, J. Lai, and P. C. Yuen. Penalized Pre-image Learning in Kernel Principal Component Analysis. *IEEE Trans. on Neural Networks, in Press*.

[13] W. W. Zou and P. C. Yuen. Very Low Resolution Face Recognition Problem. *Submitted to ECCV*, 2010.

# Cooperative and Penalized Competitive Learning for Clustering Analysis

Hong Jia
Department of Computer Science, Hong Kong Baptist University
hjia@comp.hkbu.edu.hk

## Abstract

*Competitive learning approaches with penalization or cooperation mechanism have been applied to unsupervised data clustering due to their attractive ability of automatic cluster number selection. In this paper, we further investigate the properties of different competitive strategies and propose a novel learning algorithm called Cooperative and Penalized Competitive Learning (CPCL), which implements the cooperation and penalization mechanisms simultaneously in a single competitive learning process. The integration of these two different kinds of competition mechanisms enables the CPCL to have good convergence speed, precision and robustness. Experiments on synthetic and real data sets are performed to investigate the proposed algorithm. The promising results demonstrate its superiority.*

## 1 Introduction

As a very efficient approach to clustering analysis, competitive learning has been widely applied to a variety of research areas such as data mining [1], image progress [2], Bioinformatics [3] and so forth. In the literature, a typical competitive learning algorithm is k-means [4] that learns $k$ pre-assigned seed points (also called units or centroids interchangeably) on the basis of minimizing the mean-square-error (MSE) function. Although k-means has a variety of applications in different areas, it suffers from a selection problem of cluster number as pointed out in [5, 9]. That is, k-means needs to pre-assign the number of clusters exactly; otherwise, it will almost always give out an incorrect clustering result.

To solve this selection problem, there have been two main kinds of techniques in the literature. The first one is to utilize a criterion, such as the Akaike's information criterion (AIC) [6, 19], the minimum description length (MDL) [21], the Bayesian inference criterion (BIC) [7] and the minimum message length (MML) [8, 20], to select an appropriate number of clusters by optimizing a nonlinear function over all candidates of cluster number. However, due to the repeating process for different values of cluster number k, these methods incur a large computational cost [22]. The other kind of technique is to introduce some competitive learning mechanisms into an algorithm so that it can perform automatic cluster number selection during the learning process. For example, the Rival Penalized Competitive Learning (RPCL) [9] can automatically select the cluster number by gradually driving extra seed points far away from the input data set. In this learning approach, for each input, not only the winner is updated to adapt to the input, but also the rival nearest to the winner (i.e., the second winner) is penalized by a much smaller fixed rate (also called delearning rate hereinafter). Nevertheless, the empirical studies have also found that the RPCL may completely break down without an appropriate delearning rate. Under the circumstances, paper [10] has proposed an improved version, namely Rival Penalization Controlled Competitive Learning (RPCCL), which determines the rival-penalized strength through an adaptive way based on the distance between the winner and the rival relative to the current input. Subsequently, the delearning rate in this algorithm can be fixed at the same value as the learning rate. However, both of RPCL and RPCCL always penalized the extra seed points even if they are much far away from the input data set. Consequently, the seed points as a whole will not tend to convergence. By contrast, another variant of RPCL called Stochastic RPCL (S-RPCL) [11], developed from the Rival Penalized Expectation-Maximization (RPEM) algorithm, can lead to a convergent learning process by penalizing the nearest rival stochastically based on its posterior probability. Nevertheless, when the data clusters are overlapped, the convergence speed of S-RPCL, as well as the RPCL, may become slow and the final locations of seed points may have a bias from the cluster centers.

Alternatively, Competitive and Cooperative Learning (CCL) [12] implements a cooperative learning process, in which the winner will dynamically select several nearest competitors to form a cooperative team to adapt to the input together. The CCL can make all the seed points converge to the corresponding cluster centers and the number of those seed points stayed at different positions is exactly the clus-

ter number. Nevertheless, further experiments indicate that the performance of CCL is somewhat sensitive to the initial positions of seed points. To overcome this difficulty, Li et al. [13] have proposed an improved variant; namely Cooperation Controlled Competitive Learning (CCCL) method, in which the learning rate of each seed point within the same cooperative team is adjusted adaptively based on the distance between the cooperator and the current input. The CCCL has inherited the merits of CCL and is insensitive to the initialization of the seed points. Nevertheless, the CCCL may still not work well if the initial seed points are all gathered in one cluster.

In this paper, we will present a new competitive learning algorithm, namely Cooperative and Penalized Competitive Learning (CPCL), which performs the two different kinds of learning mechanisms simultaneously: cooperation and penalization, during the single competitive learning process. That is, given an input, the winner generated from the competition of all seed points will not only dynamically select several nearest competitors to form a cooperative team to adapt to the input together, but also penalize some other seed points which compete intensively with it. The cooperation mechanism here enables the closest seed points to update together and gradually converge to the corresponding cluster centers while the penalization mechanism supplies the other seed points with the opportunity to wander in the clustering space and search for more appropriate cluster centers. Consequently, this algorithm features the fast convergence speed and the robust performance against the initialization of the seed points. The experiments have demonstrated the outstanding performance of the CPCL on both synthetic and real data. Furthermore, it is also found that the CPCL is robust against the overlap of the data cluster to a certain level, and gives a quite good estimate of the cluster centers.

The rest of this paper is organized as follows. Section II describes the proposed Cooperative and Penalized Competitive Learning approach and gives out the corresponding algorithm. Then, Section III shows the experimental results on some synthetic and real data sets. Finally, we draw a conclusion in Section IV.

## 2 Cooperative and Penalized Competitive Learning (CPCL) Approach

### 2.1 Cooperation and Penalization Mechanisms in CPCL

Here we will first describe the cooperation and penalization mechanisms of CPCL learning approach. Suppose $N$ inputs, $X_1, X_2, ..., X_N$, come from $k^*$ unknown clusters, and $k$ ($k \geq k^*$) seed points $m_1, m_2, ..., m_k$ are randomly initialized in the input space. Subsequently, given an input

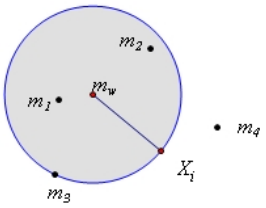



**Figure 1. The territory of the winner $m_w$, indicated by a shadow circle, in the competition with the other seed points as given an input $X_i$.**

$X_i$ each time, as described in [14], the winner among k seed points is determined by

$$I(j|X_i) = \begin{cases} 1 \ \ if \ \ j = \arg\min_{1 \leq r \leq k} \gamma_r ||X_i - m_r||^2, \\ 0 \ \ otherwise; \end{cases} \tag{1}$$

with the relative winning frequency $\gamma_r$ of $m_r$ defined as

$$\gamma_r = \frac{n_r}{\sum_{j=1}^{k} n_j}, \tag{2}$$

where $n_j$ indicates the winning times of $m_j$ in the past. That means the winning chances of frequent winning seed points are gradually reduced by an implicit penalty. After selecting out the winner $m_w$, as shown in Fig. 1, the circle centered at $m_w$ with the radius $||m_w - X_i||$ is regarded as the territory of $m_w$. Any other seed points which have intruded into this territory will be dominated by $m_w$. That is, any other seed points fallen into or on the circle, as $m_1$, $m_2$ and $m_3$ in Fig. 1, will either cooperate with the winner or be penalized by it.

The winner $m_w$ in this learning approach always chooses the seed points nearest to it as its cooperators and the number of cooperators needed by the winner is gradually increased as the learning process repeats. For example, if current status is the first epoch of the whole learning algorithm, the winner will not choose any cooperators but penalize all the seed points which have intruded into its territory. Then, in the second learning epoch, the winner will select one seed point which is nearest to it in its territory to form a cooperating team and penalize the other intruders. Consequently, for the *t*-th learning epoch, the number of cooperators chosen by the winner can be denoted as $C_t$, where $C_t = \min\{t-1, k-1\}$. This kind of cooperating scheme ensures that the seed points have enough opportunities to drift in the whole input space and converge smoothly.

After choosing cooperators, each member in the cooper-

ating team, denoted as $m_o$, will be updated by

$$m_o^{new} = m_o + \eta \frac{\|m_w - X_i\|}{\max(\|m_w - X_i\|, \|m_o - X_i\|)}(X_i - m_o), \quad (3)$$

where $\eta$ is a specified positive learning rate. It means that all the cooperative units tend to move toward the point $X_i$ and the learning strength of different seed points is adjusted adaptively based on the distance between the cooperator and the current input. Since we have a factor $\gamma$ involving in (1) when selecting the winner seed, the nearest seed point to $X_i$ is not always the winner. Therefore, the "max" function in (3) is necessary. We can find that a cooperator will have a full learning rate $\eta$ as $\|m_o - X_i\| \leq \|m_w - X_i\|$. Otherwise, the learning strength is gradually attenuated when the distance between the cooperator and the current input increases.

The other non-cooperating seed points in the winner's territory, denoted as $m_p$, will be penalized with a dynamical penalizing rate:

$$m_p^{new} = m_p - \eta \frac{\|m_w - X_i\|}{\|m_p - X_i\|}(X_i - m_p). \quad (4)$$

That is, all the penalized seed points will be moved away from the $X_i$ and the closer the seed point is to the input, the more penalization it will suffer from.

As a whole, at the earlier stage of CPCL learning approach, the penalization mechanism plays a leading role, which leads the initial seed points to drift in the input space to find a more appropriate cluster center. But during the next period, the cooperation is strengthened while the penalization is weakened, this makes all the seed points converge to the corresponding cluster centers gradually.

### 2.2 The CPCL Algorithm

Based on the previous description, the CPCL algorithm can be given as follows:

**Step1:** Pre-specify the number $k$ of clusters ($k \geq k^*$), and initialize the $k$ seed points $\{m_1, m_2, ..., m_k\}$. Set $t = 1$, $i = 1$ and $n_j = 1$ with $j = 1, 2, ..., k$, where $t$ and $i$ are used to record the number of epochs and input data, respectively.

**Step2:** Given an input $X_i$, calculate $I(j|X_i)$ by (1).

**Step3:** Determine the winner unit $m_w$. Let $S_w$ be the set of seed points fallen into the territory of $m_w$. That is, let $S_w = \emptyset$, and then we span $S_w$ by

$$S_w = S_w \cup \{m_j | \|m_w - m_j\| \leq \|m_w - X_i\|, j \neq w\}. \quad (5)$$

**Step4:** Sort the units in $S_w$ based on the distance between each unit to the winner $m_w$. We denote these units as: $m'_1, m'_2, ..., m'_s$, with

$$\|m'_1 - m_w\| \leq \|m'_2 - m_w\| \leq \cdots \leq \|m'_s - m_w\|, \quad (6)$$

where $s = |S_w|$.

**Step5:** Select a subset $S_c$ of $S_w$ to form a cooperating team of $m_w$, where $S_c = \{m'_1, m'_2, ..., m'_k\}$ with $c = |S_c| = min\{s, t-1\}$. Then update all members in $S_c$ by (3).

**Step6:** Let $S_p = S_w - S_c$, then, penalize all seed points in $S_p$ by (4).

**Step7:** Update the winner $m_w$ by

$$m_w^{new} = m_w + \eta \cdot (X_i - m_w). \quad (7)$$

**Step8:** Update $n_w$ by $n_w^{new} = n_w^{old} + 1$. Let $i = i + 1$, $t = 1 + \lfloor i/N \rfloor$.

The above step 2 to step 8 are iterated for each input until all the seed points converge.

## 3 Experimental Results

### 3.1 Experiment 1

To demonstrate the performance of the CPCL algorithm in comparison with the CCCL and S-RPCL, we generated 1,000 data points from a mixture of three 2-dimension Gaussian densities:

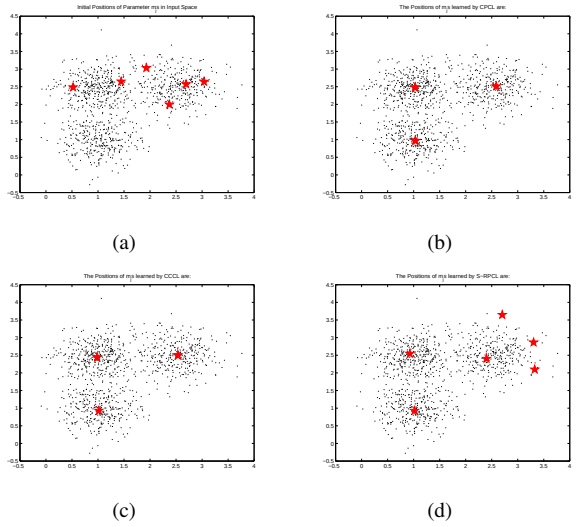
$$\begin{aligned} p(X|\Theta) = 0.3G\left[X|\begin{pmatrix}1.0\\1.0\end{pmatrix}, \begin{pmatrix}0.15, 0.0\\0.0, 0.15\end{pmatrix}\right] \\ + 0.4G\left[X|\begin{pmatrix}1.0\\2.5\end{pmatrix}, \begin{pmatrix}0.15, 0.0\\0.0, 0.15\end{pmatrix}\right] \\ + 0.3G\left[X|\begin{pmatrix}2.5\\2.5\end{pmatrix}, \begin{pmatrix}0.15, 0.0\\0.0, 0.15\end{pmatrix}\right] \end{aligned}. \quad (8)$$

It can be seen from Fig. 2(a) that the data points generated from this mixture densities are overlapped moderately and each cluster is ball-shaped. For each algorithm, the learning rate $\eta$ was set at $0.001$ and the parameter $\varphi$ in CCCL algorithm was set to 0.5 according to [13]. Six seed points were randomly initialized in the input space and Fig. 2(a) has given out their positions.

After 200 learning epochs, the positions of seed points obtained by the three algorithms are shown in Fig. 2(b) to Fig. 2(d), respectively. It can be seen that all the three algorithms have identified the true number of clusters successfully. However, the s-RPCL had not located the cluster centers accurately yet. A snapshot of the class centers obtained by S-RPCL is:

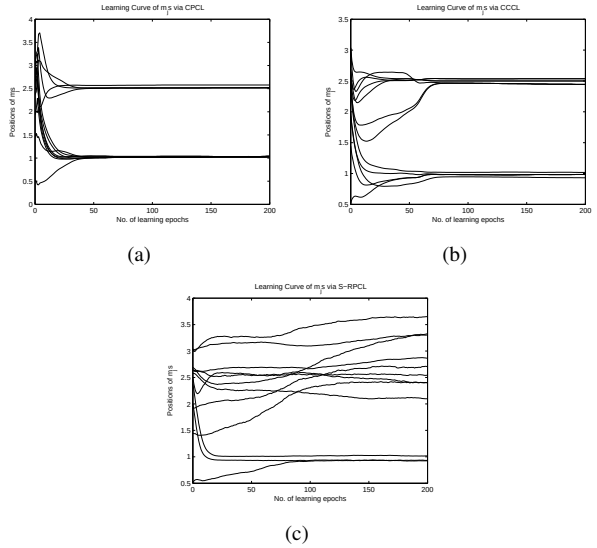
$$m_1 = \begin{pmatrix}1.0180\\0.9303\end{pmatrix}, m_2 = \begin{pmatrix}0.9264\\2.5452\end{pmatrix}, m_3 = \begin{pmatrix}2.3989\\2.4025\end{pmatrix}. \quad (9)$$

Moreover, Fig. 3 shows the learning curves of $m_j$s via each method and the convergence time of each algorithm is given out in Table 1.It can be seen that the CPCL converges faster than the other two algorithms. This scenario shows the good performance of CPCL in terms of convergence rate and precision.



(a) (b) (c) (d)

**Figure 2. The positions of six seed points marked by "★" in the input space in Experiment 1: (a) the initial random positions, (b) the positions learned via the CPCL, (c) the positions learned via the CCCL, and (d) the positions obtained by the S-RPCL.**



(a) (b) (c)

**Figure 3. The learning curves of six seed points in Experiment 1: (a) the learning curves obtained by CPCL, (b) the learning curves obtained by CCCL, and (c) the learning curves obtained via S-RPCL.**

### 3.2 Experiment 2

In this experiment, we further investigated the performance of CPCL on the mixture clusters that were seriously overlapped and some cluster was in elliptical shape. Similar to Experiment 1, 1,000 data points were generated from a mixture of three 2-dimension Gaussian densities:

$$p(X|\Theta) = 0.3G\left[X|\begin{pmatrix}1.0\\1.0\end{pmatrix},\begin{pmatrix}0.2,0.05\\0.05,0.3\end{pmatrix}\right] + 0.4G\left[X|\begin{pmatrix}1.0\\2.5\end{pmatrix},\begin{pmatrix}0.2,0.0\\0.0,0.2\end{pmatrix}\right] + 0.3G\left[X|\begin{pmatrix}2.5\\2.5\end{pmatrix},\begin{pmatrix}0.2,-0.1\\-0.1,0.2\end{pmatrix}\right]. \quad (10)$$

Furthermore, in order to verify that the CPCL algorithm is insensitive to the initial positions of seed points, we forcibly generated seven highly centralized seed points, which were located in one cluster region. Fig. 4(a) has given out the positions of input data and initial seed points. Similarly, the number of learning epochs was set to 200. As shown in Fig. 4(b), the seed points learned by CPCL had been converged accurately to the corresponding cluster centers. To further demonstrate the efficiency of CPCL, Fig. 4(c) shows the learning curves of $m_j$s, which converged during the first 100 epochs. These attractive results indicate that the penalization mechanism in CPCL can supplies the initial seed points with sufficient opportunity to wander in the clustering space even though they are centralized seriously. Moreover, it also can be seen from this experiment that, although the concept of winner's territory in CPCL is based on Euclidean distance only, this new algorithm can work well on not only ball-shaped data clusters but also elliptical ones.

**Table 1. Convergence Time of Each Algorithm in Experiment 1**

| Methods | CPCL | CCCL | S-RPCL |
|---|---|---|---|
| Convergence time | 4.3679s | 6.4311s | 9.9803s |

### 3.3 Experiment 3

The CPCL algorithm has shown its good performance under three clusters during the previous experiments. In this experiment, we will further investigate its learning capability when the true number of clusters is much larger. 1,000 data points were generated from a mixture of 10 2-dimension Gaussian density distributions which were moderately overlapped. And the proportions of the mixture components are heterogeneous.

We randomly initialized 20 seed points in the input space as shown in Fig. 5(a). After 400 learning epochs, the stable positions of seed points learned by CPCL are shown in Fig. 5(b). It is obvious that the true number of clusters has been identified and all the seed points have been converged

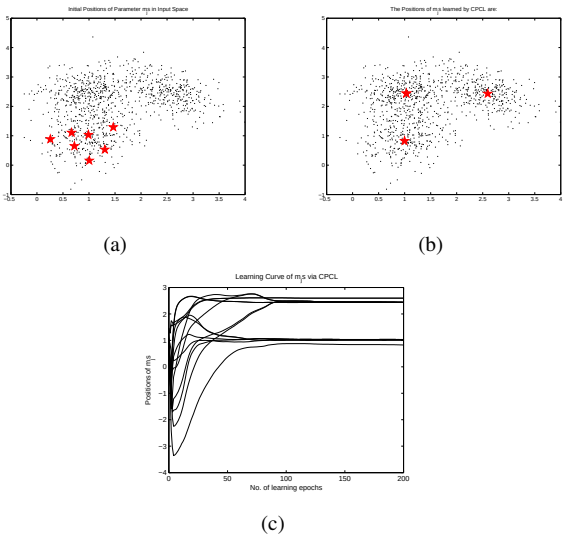



**Figure 4. The results of Experiment 2: (a) the initial positions of seven seed points marked by "★" in the input space, (b) the positions of seed points learned by CPCL, and (c)the learning curves of seven seed points.**

to the exact 10 cluster centers. So, we can see that CPCL algorithm has the robust performance even if the values of $k^*$ and $k$ both become large.

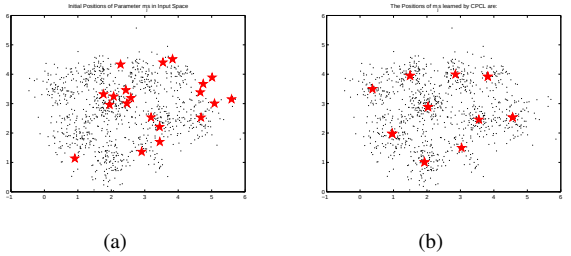



**Figure 5. In this figure, (a) shows the initial positions of 20 seed points marked by "★" in the input space, and (b) shows the positions of seed points learned by CPCL.**

### 3.4 Experiment 4

The previous experiments have showed the performance of CPCL under the synthetic data clustering. In this experiment we will investigate its efficiency on the real-world microarray data–the yeast cell cycle data published by Cho et al.[15]. The original data contained the expression profiles of 6220 genes over 17 time points taken at 10 minute intervals, covering nearly two cell cycles. The data set we used was comprised of 384 genes which had expression levels peaking at different time points corresponding to the five phases of the cell cycle. Hence, it is expected that each of the 384 genes can be assigned to one of the five clusters [15]. This subset of data is available at *http://www.cs.washington.edu/homes/kayee/model* and the standardized data was adopted in this experiment.

We assumed that the true number of clusters was unknown and arbitrarily initialized 10 seed points in the running of CPCL algorithm. For further analysis, we compared the proposed approach with other three methods: the EM algorithm based on BIC (called *Method I* hereinafter) [17], the supervised clustering method (Method II) [18] and the support vector machines algorithm (Method III) [16]. After the implement of clustering, each gene had four possible outcomes: false positive (FP), false negative (FN), true positive (TP) and true negative (TN). And the total error rate can be defined as FP+FN [18]. The clustering results of the four methods are given out in Table 2, where the results of method I-III are obtained from [18]. Furthermore, the total error rates of different algorithms are summarized in Table 3. We can see that, in terms of the small error rates, our method has a good performance in this experiment. In addition, the five groups of genes whose expression level peak at different phases of the cell cycle formed by CPCL algorithm is shown in Fig. 6. It can be observed that the genes with similar attributes have been clustered together, which indicated that the proposed method is indeed effective for the cluster analysis of gene expression data.

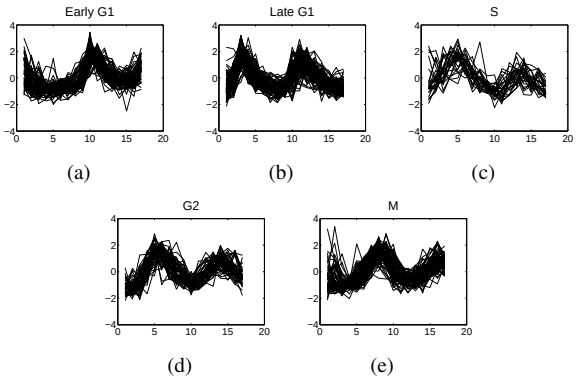



**Figure 6. The five groups of genes classified by the CPCL algorithm .**

## 4 Concluding Remarks

In this paper, we have presented a novel competitive learning algorithm named Cooperative and Penalized Competitive Learning (CPCL), which performs the competition

**Table 2. Comparison of the Clustering Results of the Four Methods on the Microarray Data**

| Cell division phase | Methods | FP | FN | TP | TN |
|---|---|---|---|---|---|
| Early G1 (67 genes) | CPCL | 24 | 18 | 49 | 293 |
| | Method I | 50 | 12 | 55 | 267 |
| | Method II | 21 | 21 | 46 | 296 |
| | Method III | 38 | 10 | 57 | 279 |
| Late G1 (135 genes) | CPCL | 38 | 22 | 113 | 211 |
| | Method I | 28 | 40 | 95 | 221 |
| | Method II | 24 | 35 | 100 | 225 |
| | Method III | 43 | 10 | 125 | 206 |
| S (75 genes) | CPCL | 11 | 58 | 17 | 298 |
| | Method I | 33 | 49 | 26 | 276 |
| | Method II | 37 | 36 | 39 | 272 |
| | Method III | 72 | 18 | 57 | 237 |
| G2 (52 genes) | CPCL | 24 | 22 | 30 | 308 |
| | Method I | 28 | 41 | 11 | 304 |
| | Method II | 18 | 29 | 23 | 314 |
| | Method III | 46 | 5 | 47 | 286 |
| M (55 genes) | CPCL | 26 | 3 | 52 | 303 |
| | Method I | 38 | 42 | 13 | 291 |
| | Method II | 19 | 8 | 47 | 310 |
| | Method III | 47 | 2 | 53 | 283 |

**Table 3. The Total Error Rates of the Four Methods on the Microarray Data**

| Methods | FP | FN | FP+FN |
|---|---|---|---|
| CPCL | 123 | 123 | 246 |
| Method I | 177 | 184 | 361 |
| Method II | 119 | 129 | 248 |
| Method III | 246 | 45 | 291 |

with the two different kinds of mechanisms simultaneously: cooperation and penalization. On the one hand, the cooperation mechanism enables the closest seed points to update together and gradually converge to the corresponding cluster centers, which gives the algorithm good convergence speed and high precision. On the other hand, the penalization mechanism provides the other seed points with the opportunity to wander in the clustering space, which enables it to perform the clustering problem with the robustness against the initialization of the seed points and the overlap of the data clusters. Experimental results on both synthetic data and the yeast cell cycle microarray data have shown the outstanding performance of the proposed approach.

## References


[1] U. Fayyad, G. Piatetsky-Shpiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[2] T. Uchiyama and M.A. Arib. Color Image Segmentation Using Competitive Learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1197-1206, Dec. 1994.

[3] Y. Lu, S.Y. Lu, F. Fotouhi, Y.P. Deng, and S. Brown. Incremental Genetic K-means Algorithm and Its Application in Gene Expression Data Analysis. *BMC Bioinformatics*, vol. 5, no. 172, Oct. 2004.

[4] J.B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. *Proc. 5-th Berkeley Symp. Mathematical Statistics and Probability*, vol. 1, pp. 281-297, Berkeley, Calif., USA, 1967.

[5] Y.M. Cheung. K*-means: A New Generalized K-Means Clustering Algorithm. *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2883-2893, 2003.

[6] H. Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. *Proc. Second International Symp. Information Theory*, pp. 267-281, 1973.

[7] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

[8] C.S. Wallace and D.L. Dowe. Minimum Message Length and Kolmogorov Complexity. *The Computer Journal*, vol. 42, no. 4, pp. 270-283, 1999.

[9] L. Xu, A. Krzyzak, and E. Oja. Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 636-648, Jul. 1993.

[10] Y.M. Cheung. Rival Penalization Controlled Competitive Learning for Data Clustering with Unknown Cluster Number. *Proc. 9th Int'l Conf. Neural Information Processing*, pp. 18-22, Singapore, Nov. 2002.

[11] Y.M. Cheung. Maximum Weighted Likelihood via Rival Penalized EM for Density Mixture Clustering with Automatic Model Selection. *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 6, pp. 750-761, Jun. 2005.

[12] Y.M. Cheung. A Competitive and Cooperative Learning Approach to Robust Data Clustering. *Proc. IASTED Int'l Conf. Neural Networks and Computational Intelligence*, pp. 131-136, Grindelwald, Switzerland, 2004.

[13] T. Li, W.J. Pei, S.P. Wang, and Y.M. Cheung. Cooperation Controlled Competitive Learning Approach for Data Clustering. *Proc. Int'l Conf. Computational Intelligence and Security*, pp. 24-29, 2008.

[14] S.C. Ahalt, A.K. Krishnamurty, P. Chen, and D.E. Melton. Competitive Learning Algorithms for Vector Quantization. *Neural Networks*, vol. 3, no. 3, pp. 277-291, 1990.

[15] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, and R.W. Davis. A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, vol. 2, pp. 65-73, 1998.

[16] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares, and D. Haussler. Knowledge-Based Analysis of Micro- Array Gene Expression Data by Using Support Vector Machines. *Proc. Natl Academy of Sciences of the USA*, vol. 97, pp. 262-267, 2000.

[17] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo. Model- Based Clustering and Data Transformations for Gene Expression Data. *Bioinformatics*, vol. 17, pp. 977-987, 2001.

[18] Y. Qu and S. Xu. Supervised Cluster Analysis for Microarray Data Based on Multivariate Gaussian Mixture. *Bioinformatics*, vol. 20, pp. 1905-1913, 2004.

[19] T. Bengtsson and J.E. Cavanaugh. An Improved Akaike Information Criterion for State-space Model Selection. *Computational Statistics & Data Analysis*, vol. 50, no. 10, pp. 2635-2654, Jun. 2006.

[20] N. Bouguila and D. Ziou. Unsupervised Selection of a Finite Dirichlet Mixture Model: An MML-based Approach. *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 8, pp. 993-1009, Aug. 2006.

[21] S. Marsland, C.J. Twining, and C.J. Taylor. A Minimum Description Length Objective Function for Groupwise Non-rigid Image Registration. *Image and Vision Computing*, vol. 26, no. 3, pp. 333-346, Mar. 2008.

[22] Z. Lu and H.H.S. Ip. Generalized Competitive Learning of Gaussian Mixture Models. *IEEE Trans. Systems, Man, and Cybernetics*, vol. 39, no. 4, pp. 901-909, Aug. 2009.

# A Linear-chain CRF-based Learning Approach For Web Opinion Mining

Luole Qi

August 19, 2010

## Abstract

The task of opinion mining from product reviews is to extract the product entities, opinions on the entities and determine whether the opinions are positive, negative or neutral. Reasonable performance on this task has been achieved by employing rule-based, statistical approaches or generative learning models such as hidden Markov model (HMMs). In this paper, we proposed a discriminative model using linear-chain Conditional random field (CRFs) for opinion mining and extraction. CRFs can naturally incorporate arbitrary, non-independent features of the input without making conditional independence assumptions among the features. This can be particularly important for opinion mining on product reviews. We evaluate our approach base on three criteria: recall, precision and F-score of extracted entities, opinions and their polarity. Compared to other methods, our approach is more effective for opinion mining tasks.

## 1 Introduction

The reviews are now well recognized increasingly useful in business, education, especially in e-commerce, since they contain valuable opinions and customers could assess a product by reading opinions of other customers, which will help them to decide whether to purchase the product or not. Nowadays, many e-commerce websites such as Amazon.com, Yahoo shopping, Epinions allow users to post their opinions freely. Thus, there are usually a large amount of product reviews available on the internet. The reviews number even could be thousands in some large websites for a hot product, which makes it difficult for a potential customer to go over all of them.

This is indeed a problem for the customers. In response, researchers have done some work on opinion mining which aims to extract the essential information of reviews. Previous works have been based on two major approaches: rule-based techniques [2] and statistic methods [10]. In [1], a new learning approach based on a sequence model named hidden Markov model (HMMs) was adopted and was proved more effective. However, HMMs have the limitation that it is difficult to model arbitrary, dependent features of the input sequence.

Conditional random field (CRFs)[11] are discriminative graphical models that can model these overlapping, non-independent features. A special case, linear-chain CRFs, can be thought of as the undirected graphical model version of HMMs. Motivated by the fact that CRFs have out-performed HMMs on language processing[12][13], we proposed a linear-chain CRF-based instead of HMM-based learning approach to mine and extract opinions from product reviews on the web in this paper. Our objective is to answer the following questions: (1) how to construct and restrict our linear-chain CRFs model by defining feature functions. (2) How to choose criteria to train our specific model from the manually labeled data. (3) How to automatically extract potential product entities and opinion entities from the reviews and identify opinion polarity with our trained model. In our work, we evaluate the model on recall, precision and F-score of extracted entities, opinions and their polarity, and the experimental results prove the proposed approach in web opinion mining and extraction from online product reviews is effective.

In summary, this paper has the following contributions: 1) we demonstrate linear-chain CRFs models performs

better than L-HMMs approach integrating linguistic features in opinion mining and extraction. 2) The feature functions of CRFs we defined in our work for the model construction are proved to be robust and effective by our experimental results.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 describes in detail our proposed CRFs Opinion Mining model. We describe in Section 4 experimental results and make a comparison among different methods. In the end, we give our conclusions and our future work in Section 5.

# 2 Related Work

Recently, many researchers have studied the problem. In Turney al et's work [2], they used pointwise mutual information (PMI) to calculate the average semantic orientation (SO) of extracted phrases for determing documents polarity. Pang al et [4] examined the effectiveness of applying machine learning techniques to the sentiment classification problem with movie review data. Hatzivassiloglou and Wiebe [6] studied the effects of dynamic adjectives, semantically oriented adjectives, and gradable adjectives on a simple subjectivity classifier, and proposed a novel trainable method that statistically combines two indicators of gradability. Wiebe and Riloff [7] proposed a system called OpinionFinder that performs subjectivity analysis, automatically identifying when opinions, sentiments, speculations, and other private states are present in text. Das and Chen [9] studied document level sentiment polarity classification on financial documents. All these works are related to sentiment classification, it uses the sentiment to represent reviewer's whole opinion and does not find what the reviewer liked and disliked. For example, an negative sentiment on an object does not mean that the reviewer dislike everything and also an positive sentiment does not mean that the reviewer like everything.

To solve this problem, some researchers try to mine and extract opinions on the feature level as well as the sentence level. Hu and Liu [10] proposed a feature-based opinion summarization system capturing high frequency feature words by using association rules under a statistical framework. It mines only the features of the product that customers have expressed their opinions on and a summary is generated by using high frequency feature words (the top ranked features) and ignoring infrequent features. Popescu and Etzioni [3] improved Hu and Liu's work by removing those frequent noun phrases that may not be features. It tries to identify part-of relationship and achieves a better precision but a small drop in recall. Christopher Scaffidi et al [8] presented a new search system called Red Opal which could examine prior customer reviews, identify product features, and score each product on each feature. Red Opal uses these scores to determine which products to show when a user specifies a desired product feature. However, all these work failed to identify infrequent entities effectively. Our approach can address this issue effectively.

Another work we want to mention here is a machine learning system called OpinionMiner which was designed by Weijin et al [1]. It was built under the framework of lexicalized HMMs integrating multiple important linguistic features into automatic learning. it's closely related to our work, but we employ CRFs instead of HMMs to avoid some limitations inherent in HMMs. For example, it can not represent neither distributed hidden state nor complex interaction among labels, it also can not use rich, overlapping feature sets.

# 3 The Proposed Approach

Fig. 1 gives the architecture overview for our approach, which performs the opinion mining in four main tasks: (1) Pre-work which include crawing raw review data and cleaning; (2) data processing; (3) train the liner-chain CRFs model; (4) Testing.

## 3.1 CRFs Models

Conditional random fields (CRFs) are conditional probability distributions that factorize according to an undirected model. It could be defined as follows: considering a graph $G = (V, E)$, let $Y = (Y_v)_{v \in V}$, and $(X, Y)$ is a CRF, where $X$ is the set of variables over the observation sequences to be labeled (e.g., a sequence of natural language words which form a sentence), and $Y$ is the set of random variables over the corresponding labeling sequences which obey the Markov property with respect to the graph(e.g., part-of-speech tags for the words sequences). It models $p(y|x)$ globally conditioned on the

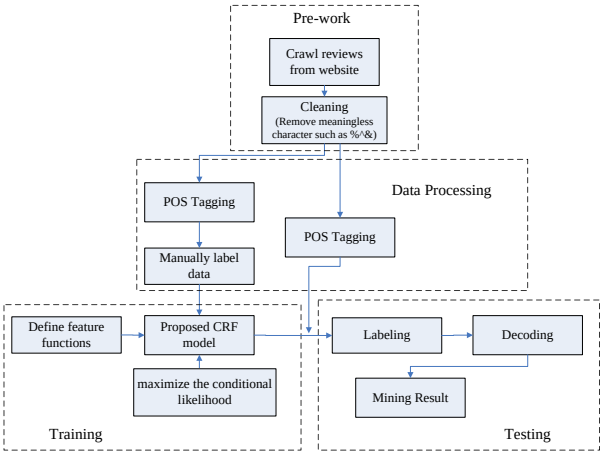



Figure 1: The architecture of proposed system

observation X:

$$p(y|x) = \frac{1}{Z(x)} \prod_{i \in N} \phi_i(y_i, x_i) \quad (1)$$

where $Z(x) = \sum_y \prod_{i \in N} \phi_i(y_i, x_i)$ is a normalization factor over all state sequences for the sequence $x$. We assume the potentials factorize according to a set of features $f_k$, as

$$\phi_i(y_i, x_i) = \exp(\sum_k \lambda_k f_k(y_i, x_i)) \quad (2)$$

Given such a model as defined in equation (1), the most probable labeling sequence for an input $x$ is

$$\widehat{Y} = \arg\max_y axp(y|x) \quad (3)$$

## 3.2 Problem Statement

Our goal was to effectively extract product entities from reviews and identify opinion's polarity. The product entities could be divided into four categories according to [1]: Components, Functions, Features and Opinions. Please notice that the features mentioned here indicate product feature but feature functions for constructing CRFs models. In our work, we use the same classifications of product entities in [1]. Table 1 shows the four categories of entities and their examples. [1]

Table 1: Four entities and their examples

| | |
|---|---|
| Components: | Physical objects of a product, such as cellphone's LCD |
| Functions: | Capabilities provided by a product, e.g., movie playback, zoom, automatic fillflash, auto focus |
| Features: | Properties of components or functions, e.g., color, speed, size, weight |
| Opinions: | Ideas and thoughts expressed by reviewers on product features, components, functions. |

Our work employ three types of tags to represent the words: entity tags, position tags and opinion tags. We use the categories names of a product entity as entity tags. For the word which is not an entity, we use character 'B' to represent it. Usually, an entity could be a single word or a phrase (words chuck). For a phrase entity, we assign a position to each word of this phrase. Any word of a phrase could only be three types of positions: the beginning of the phrase, the middle of the phrase and the end of phrase. Here we use character 'B', 'M' and 'E' as position tags to represent the position of a word in beginning, middle and end of a phrase respectively. Considering each opinion has different orientations and whether it is explicit or implicit, we use character 'P', 'N' to represent Positive, Negative and use "Exp", "Imp" to represent explicit opinion and Implicit opinion. These tags are opinion tags. Combining all these tags, we could tag out any word and its role in a sentence. For example, we label the sentence "The image is good and it is ease of use" from a camera review:

The(B) image(Featue-B) is(B) good(Opinion-B-P-Exp) and(B) it(B) is(B) ease(Feature-B) of(Feature-M) use(Feature-E).

In this sentence, 'image' and 'ease of use' are both features of this camera and 'ease of use' is a phrase, thus we add '-B', '-M' and '-E' to the end of categories tag 'Feature to specify the position of each word in the phrase. 'Good' is a positive explicit opinion expressed on the feature 'image', so its tag is 'Opinion-B-P-Exp'. All other

words which are not any category of entity are given the tag 'B'.

In this way, if we know each word's hybrid tag, we could extract the product entities and identify the opinions' orientations. Thus, the task of opining mining and extraction is transformed to a labeling task.

We now describe the model topology and the feature functions used to construct a CRFs opinion mining system. Our problem can be described as follows: given a sequence of words $W = w_1 w_2 w_3 ... w_n$ and corresponding parts of speech $S = s_1 s_2 s_3 ... s_n$, and the objective is to find an appropriate sequence of hybrid tags which maximize the conditional likelihood according to equation (3)

$$\widehat{T} = \arg\max_T p(T|W,S) = \arg\max_T \prod_{i=1}^{N} p(t_i|W,S,T^{(-i)}) \tag{4}$$

where $T^{(-i)} = \{t_1 t_2 ... t_{i-1} t_{i+1} ... t_N\}$. However, we could see that the hybrid tag in position $i$ depends on all the words $W = w_{1:N}$, part-of-speeches $S = s_{1:N}$ and tags except itself, which makes it very hard for computing in practice as this involves too many parameters. To simply our problem, we employ linear-chain CRFs as an approximation to restrict the relationships within tags. It's graphical structure shown in Figure 2. We could see that in a linear-chain CRF, all the nodes in the graph form a linear chain and each feature involves only two consecutive hidden states. Thus equation (4) could be rewritten as

$$\widehat{T} = \arg\max_T p(T|W,S) = \arg\max_T \prod_{i=1}^{N} p(t_i|W,S,t_{i-1}) \tag{5}$$

## 3.3 Feature Functions

From the model above, we can see there are still many parameters need to be considered. To make the model computable, we need to define the relationships among the observation states $W = w_{1:N}$, $S = s_{1:N}$ and hidden states $T = t_{1:N}$ to reduce the unnecessary calculations. Thus, as important components of CRFs, feature functions' definition is crucial to our problem. Let $w_{1:N}$, $s_{1:N}$ be the observations (e.g., words sequence and corresponding parts of speech respectively), $t_{1:N}$ the hidden labels (e.g., hybrid tags). In our case of linear-chain CRF, the general form of a feature function is $f_i(t_{n-1}, t_n, w_{1:N}, s_{1:N}, n)$, which looks at a pair of adjacent states $t_{n-1}$,$t_n$, the whole input sequence $w_{1:N}$ as well as $s_{1:N}$ and where we are in the sequence. For example, we can define a simple feature function which produces binary values: the returned value is 1 if the current word $w_n$ is "good", the corresponding part-of-speech $s_n$ is "JJ" which means single adjective word and if the current state $t_n$ is "Opinion":

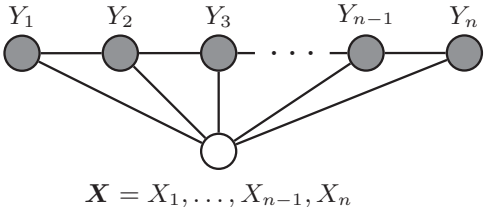



Figure 2: liner-CRFs graphic structure

$$f_i(t_{n-1}, t_n, w_{1:N}, s_{1:N}, n) = \begin{cases} 1 & if\ w_n = good, \quad s_n = JJ \\ and\ t_n = Opinion & \\ 0 & otherwise \end{cases} \tag{6}$$

Combining feature function with equation (1) and equation (2), we have:

$$p(t_{1:N}|w_{1:N}, s_{1:N}) = \frac{1}{Z} \exp\left(\sum_{n=1}^{N}\sum_{i=1}^{F} \lambda_i f_i(t_{n-1}, t_n, w_{1:N}, s_{1:N}, n)\right) \tag{7}$$

According to equation (7), the feature function $f_i$ depends on its corresponding weight $\lambda_i$. That is if $\lambda_i > 0$, and $f_i$ is active, it will increase the probability of the tag sequence $t_{1:N}$ and if $\lambda_i < 0$, and $f_i$ is inactive, it will decrease the probability of the tag sequence $t_{1:N}$.

What worth mentioned here is another example of feature function, please consider

$$f_i(t_{n-1}, t_n, w_{1:N}, s_{1:N}, n) = \begin{cases} 1 & if\ w_n = good, \quad s_{n+1} = NN \\ and\ t_n = Opinion & \\ 0 & otherwise \end{cases} \tag{8}$$

For the phrase "good image", the features in equation (6) and (8) are both active if the current word is "good". It boosts up the belief of $t_i = Opinion$ to both $\lambda$. This is an example of overlapping features which HMMs can not do: HMMs cannot consider the next word, nor can they use overlapping features.

Thus in addition to employing linear-chain CRFs to simplify the relations within hidden states $T$, we also define several different types of feature functions to specify state-transition structures among $W$, $S$ and $T$. As different state transition features can be defined to form different markov-order structures, our different state transitions features are based on different markov order for different classes of features. Here we will describe first order features in detail:

1. The assignment of current tag $t_i$ is supposed to only depend on the current word. The feature functions are represented as $f(t_i, w_i)$.

2. The assignment of current tag $t_i$ is supposed to only depend on the current part-of-speech. The feature functions are represented as $f(t_i, s_i)$.

3. The assignment of current tag $t_i$ is supposed to depend on both the current word and current part-of-speech. The feature functions are represented as $f(t_i, s_i, w_i)$.

All the three types of feature functions are first-order, of which the inputs are examined in the context of the current state only. There are no separate parameters for state transitions at all. We also define first-order+transitions features and second-order features which are examined in the context of the current and previous states. We did not define third-order or higher-order features because they create more data sparse problem and require more memory in training. Table 2 shows all the features we defined in our model.

Table 2: The feature functions types and their expressions

| Feature type | Expressions |
|---|---|
| First-oder | $f(t_i, w_i)$, $f(t_i, s_i)$, $f(t_i, s_i, w_i)$ |
| First-order+transitions: | $f(t_i, w_i)f(t_i, t_{i-1})$, $f(t_i, s_i)f(t_i, t_{i-1})$, $f(t_i, s_i, w_i)f(t_i, t_{i-1})$ |
| Second-order: | $f(t_i, t_{i-1}, w_i, )$, $f(t_i, t_{i-1}, s_i)$, $f(t_i, t_{i-1}, s_i, w_i)$ |

## 3.4 CRFs Training

After the graph and feature functions are defined, the model is fixed. Thus the purpose of training is find out all the values of $\lambda_{1:N}$. Usually One may set $\lambda_{1:N}$ by domain knowledge, however, in our case, we would learn $\lambda_{1:N}$ from data as there is little knowledge available for us. The fully labeled data sequence is $\{(w^{(1)}, s^{(1)}, t^{(1)}), ..., (w^{(m)}, s^{(m)}, t^{(m)})\}$, where $w^{(1)} = w^{(1)}_{1:N_1}$ the first words sequence, $s^{(1)} = s^{(1)}_{1:N_1}$ the first part-of-speech sequence, $t^{(1)} = t^{(1)}_{1:N_1}$ the first tags sequence respectively and so on. Since CRFs define the conditional probability $p(t|w, s)$, the appropriate objective for parameter learning is to maximize the conditional likelihood of the training data

$$\sum_{j=1}^{m} \log p(\mathbf{t}^{(j)}|\mathbf{w}^{(j)}, \mathbf{s}^{(j)}) \tag{9}$$

To avoid over-fitting, log-likelihood is usually penalized by some prior distribution over the parameters. A commonly used prior is a zero-mean Gaussian. If $\lambda \sim N(0, \sigma^2)$, the objective becomes

$$\sum_{j=1}^{m} \log p(t^{(j)}|w^{(j)}, s^{(j)}) - \sum_{i}^{F} \frac{\lambda_i^2}{2\sigma^2} \tag{10}$$

The objective is concave, so the $\lambda$ have a unique set of global optimal values. We learn parameters by computing the gradient of the objective function, and using the gradient in an optimization algorithm like L-BFGS. The

gradient of the objective function is computed as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \lambda_k} \sum_{j=1}^{m} \log p(t^{(j)}|w^{(j)}, s^{(j)}) - \sum_{i}^{F} \frac{\lambda_i^2}{2\sigma^2} \\
&= \frac{\partial}{\partial \lambda_k} \sum_{j=1}^{m} (\sum_{n} \sum_{i} \lambda_i f_i(t_{n-1}, t_n, w_{1:N}, s_{1:N}, n) - \log T^{(j)}) \\
&= \sum_{j=1}^{m} \sum_{n} f_k(t_{n-1}, t_n, w_{1:N}, s_{1:N}, n) \\
&- \sum_{j-1}^{m} \sum_{n} E_{t'_{n-1}, t'_n}[f_k(t'_{n-1}, t'_n, w_{1:N}, s_{1:N}, n)] - \frac{\lambda_k}{\sigma^2}
\end{aligned} \tag{11}$$

In equation (10), The first term is the empirical count of feature $i$ in the training data, the second term is the expected count of this feature under the current trained model and the third term is generated by the prior, we could ignore it. Hence, the derivative measures the difference between the empirical count and the expected count of a feature under the current model. Suppose that in the training data a feature $f_k$ appears A times, while under the current model, the expected count of $f_k$ is B. When $|A| = |B|$, the derivative is zero. Therefore, training can be thought of as finding $\lambda$s that match the two counts.

<Review>
<Title>Great Camera</Title>
<Reviewer>l_infante69</Reviewer>
<CreateTime>1133976475</CreateTime>
<HelpfulRecommendations>3</HelpfulRecommendations>
<TotalRecommendations>4</TotalRecommendations>
- <Ratings>
<Rating ratingType="Features">5</Rating>
<Rating ratingType="Overall">5</Rating>
<Rating ratingType="Quality">5</Rating>
<Rating ratingType="Support">5</Rating>
<Rating ratingType="Value">5</Rating>
</Ratings>
<OverallRating>5</OverallRating>
<Pro>Light weight, great battery power</Pro>
<Con>PC Picture Software and Users Guide</Con>
<Posting>This is a great camera. I shopped around and got a great price. This is my first digital camera. No problems with the pictures or the screen. The battery power is fantastic, the size is great, and the pictures and photo options are really nice. <br> <br>The user guide isn't very user friendly. If you are not electronic savvy, it may take some time to figure out this camera. <br> <br>The software to load the pictures on my PC is also not very user friendly. The only way I can crop and edit pictures is by loading into a different application (such as HP photo director).</Posting>
</Review>

Figure 3: Example of one full review

# 4 Evaluation

In this section, we present a detailed comparison of recall, precision and F-score of extracted entities, opinions and their polarity. Recall is $\frac{|C \cap P|}{|C|}$ and Precision is $\frac{|C \cap P|}{|P|}$, where $C$ and $P$ are the sets of correct and predicted hybrid tags, respectively. F score is the harmonic mean of precision and recall, $\frac{2RP}{R+P}$. Three methods would be evaluated: the baseline method, the L-HMM model in [1] and our proposed CRF model. We crawl product reviews from Yahoo shopping as datasets for the evaluation. The review format is semi-structured and it consists of several parts: Title, Reviewer, Ratings, Pro, Con, Posting, etc. Fig. 3 shows a whole review format.

In our work, we only use the Posting part since it is free text which could express costumer's opinion fully. After some cleaning work such as removing meaningless characters, we then use the LBJPOS tool by natural language processing group of University of Illinois, Urbana-Champaign [16] to produce the part-of-speech tag for each word in every review. We also use the corpus of Liu and Hu's as datasets for evaluation. As their data only tag the entities for each product Thus we need to retag their data. We randomly choose 476 reviews in total for three cameras and one cellphone and manually label all of them using the hybrid tags. All tagged data are then divided into 4 four sets to perform a 4-fold crossvalidation.The whole process is as the following: a single set is retained as the validation data for testing, and the remaining 3 sets are used as training data. The cross-validation process is then repeated 4 times, with each of the 4 subsamples used exactly once as the validation data. The 4 results then would be averaged to produce a single estimation.

## 4.1 Rule-base Method

Motivated by [2], we design a straightforward rule-based method as the baseline system for comparison. The first step is performing Part-of-Speech task, one example of POS tagging result is here:

(PRP I) (VBD used) (NNP Olympus) (IN before) (, ,)
(VBG comparing) (TO to) (NN canon) (, ,) (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) (. .)

The baseline system then will extract product entities, opinions and their orientations.

#### 4.1.1 Product Entities Extraction

This system will apply for some basic rules to extract product entities: 1. a single noun which follows an adjective word or consecutive adjective words will be seen as a product entity. (JJ + NN or JJ + JJ + NN) 2. Any single noun word connects an adjective word by a linking verb will be seen as a product entity. (NN + VBZ +JJ) 3. Any consecutive noun words which appear in the position described above will be seen as a product entity phrase.

#### 4.1.2 Opinion Extraction and orientations determination

Rules for determining opinion word and their orientations are described as follow: 4. All the adjective words appearing in rule 1 and 2 will be seen as opinion entities. 5. The orientations of opinion entities will be determined by a lextion which indicate the polarity of over 8000 adjective words. This is made by Pang et al's work [15].

### 4.2 L-HMMs Approach

The work in [1] integrated linguistic features such as part-of-speech and lexical patterns into HMMs. It also aims to find out an appropriate sequence of hybrid tags $T = t_1 t_2 t_3 ... t_n$ that maximize the conditional probability described in equation (4). They rewrite equation (4) as

$$\widehat{T} = \arg\max_{T} axp(W, S|T)p(T) = \arg\max_{T} axp(S|T) \\ p(W|T, S)p(T)$$

Three hypotheses are then made for simplifying the problem: (1) the assignment of current tag is supposed to depend not only on its previous tag but also previous J words. (2) The appearance of current word is assumed to depend not only on the current tag, current POS, but also the previous K words. (3) The appearance of current POS is supposed to depend both on the current tag and previous L words and J=K=L. Then the objective is

$$\arg\max_{T} ax \prod_{i=1}^{N} \left\{ \begin{array}{l} p(s_i|w_{i-1}, t_i)\times \\ p(w_i|w_{i-1}, s_i, t_i)\times \\ p(t_i|w_{i-1}, t_{i-1}) \end{array} \right\}$$

Maximum Likelihood Estimation (MLE) is used to estimate the parameters. Some techniques are also used in this approach such as information propagation using entitys synonyms, antonyms and related words, token transformations and etc.

### 4.3 Experimental Results and Discussion

Table 3 shows the evaluation results of the 4-fold cross-validation based on the recall, precision and F-score of extracted entities, opinions and their polarity. In the table, line 1 lists each method we want to evaluate: the baseline system, the L-HHMs model with POS tagging proposed by [1] and our approach. the following lines give the recall, precision and F-scores of the four kinds of product entities.

We observe that CRFs increase the performance on nearly all the four entities. CRFs performs better than L-HMMs, with the overall word precision improved from 83.9% to 90.0% and the F-score improved from 77.1% to 84.3%. There are two major reasons that lead to its results. First of all, HMMs assume that each feature is generated independently by some hidden process, that's to say, only tags can affect each other and ignore the underlying relationships among the words and POS tags. But, this is in general not the case in a labeling task, the words and POS tags also play an important role in predicting the hidden states. Secondly, HMMs does not model the overlapping features. We also achieved a high recall in the proposed approach. The average recall of our method for the four entities was improved from 72.0% to 79.8%. This is promising as the recall rate for labeling task is easily affected by errors in tagging. For example, if the tags "Opinion-B-P-Exp, Opinion-M-P-Exp, Opinion-M-P-Exp, Opinion-E-P-Exp" is labled as "Opinion-B-P-

Exp, Opinion-E-P-Exp, Opinion-M-P-Exp, Opinion-E-P-Exp", the labeling accuracy is 75%, but recall is 0.

CRFs also perform better than Baseline approach, yielding new state of the art performance on this task. After a close inspection of the terms generated by baseline system, we see that there are lots of errors in entities such as it takes "seller" as a product feature because it's a noun word. Our method can avoid this problem fully.

We also conduct a comparison using different datasets with our approach (Hu's 238 documents and yahoo shopping's 238 documents), Fig. 4 shows the precision, recall and F-scores of four entities. We can see in all three criteria, the corresponding values generated by different datasets are close. The biggest absolute value of their difference is the precision of functions, but it is only 3%. This shows our method achieves a stable performance with different datasets which proves the robustness of our approach.

Our work could also identify some infrequent entities since the overlapping feature functions can increase the impact of infrequent entities. This greatly avoid the affect of the low counts of such entities. For example, the entity "ISO" only appears once in our data, which will be omitted in other approaches. In our method, both function $f(t_i, s_i, w_i)$ and $f(t_i, w_i)$ would model this feature which makes $f_1$ and $f_2$ can be both active for the tagging.

In this paper, we also extracted the potential non-noun product entities, such as "adjust" and non-adjective opinions, such as "strongly recommended". These non-noun entities and non-adjective opinions were ignored by previously proposed approaches which were based on the assumption that product entities must be noun or noun phrases and the opinions must be adjective words. Our system can well identify these overlooked product entity information.

# 5 Conclusion

In this paper, we proposed a novel linear-chain CRFs-based learning approach for opinion mining and extraction. Contrasting to HMMs which assume that each feature is generated independently by some hidden process, CRFs can handle non-independent input features, which can be beneficial in complex domains. The experiment results showed the effectiveness of the proposed approach.

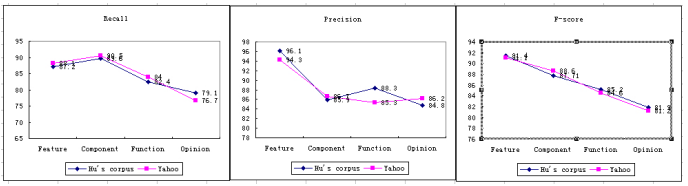



Figure 4: The Precision, Recall and F-scores of two datasets

Table 3: 4-fold crossvalidation results(R: recall, P:precision, F: F-score)

| Methods | | Baseline | L-HMM+POS | CRF+POS |
|---|---|---|---|---|
| Feature Entities(%) | R | - | 78.6 | 81.8 |
| | P | - | 82.2 | 93.5 |
| | F | - | 80.4 | 87.2 |
| Component Entities(%) | R | - | 96.5 | 91.8 |
| | P | - | 95.3 | 98.7 |
| | F | - | 96.0 | 95.1 |
| Function Entities(%) | R | - | 58.9 | 80.4 |
| | P | - | 81.1 | 83.7 |
| | F | - | 68.2 | 82.0 |
| Opinion Entities(%) | R | - | 53.7 | 65.3 |
| | P | - | 76.9 | 84.2 |
| | F | - | 63.2 | 73.5 |
| All Entities(%) | R | 27.2 | 72.0 | 79.8 |
| | P | 24.3 | 83.9 | 90.0 |
| | F | 25.7 | 77.1 | 84.3 |

In our future work, we would consider incorporating richer features to improve our model. Due to the complicacy of human natural language, some long-distance features of the input are beyond our hypotheses. Thus, if some rich features are included, we could further employ automatic feature induction techniques to find non-obvious conjunctions of features that may improve performance. Moreover, the mining result of our system can also be used for us to develop some feasible web applications such as recommender systems. We will also consider training our proposed model to assign a score for each entity based on the degree of the opinion word which describe it.

# References


[1] Jin, W., Ho, H., and Srihari, R. K.: OpinionMiner: a novel machine learning system for web opinion mining and extraction. In: 15th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, pp. 1195–1204. ACM, New York, NY, USA (2009)

[2] Turney, Peter D. : Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: 15th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, pp. 417–424. Association for Computational Linguistics, Morristown, NJ, USA (2002)

[3] Popescu, A. and Etzioni, O. : Extracting Product Features and Opinions from Reviews. In: Conference on Empirical Methods in Natural Language Processing, pp. 339-346. Association for Computational Linguistics, Morristown, NJ, USA (2005)

[4] Pang, B. and Lee, L. and Vaithyanathan, Shivakumar. : Thumbs up?: sentiment classification using machine learning techniques. In: the ACL-02 conference on Empirical methods in natural language processing, pp. 79–86. Association for Computational Linguistics, Morristown, NJ, USA (2002)

[5] Dave, K. and Lawrence, S. and Pennock, David M. : Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: the 12th international conference on World Wide Web, pp. 519–528. ACM, New York, NY, USA (2002)

[6] Hatzivassiloglou, Vasileios and Wiebe, Janyce M. : Effects of adjective orientation and gradability on sentence subjectivity. In: the 18th conference on Computational linguistics, pp. 299–305. Association for Computational Linguistics, Morristown, NJ, USA (2000)

[7] Wilson, Theresa and Hoffmann, Paul and Somasundaran, Swapna and Kessler, Jason and Wiebe, Janyce and Choi, Yejin and Cardie, Claire and Riloff, Ellen and Patwardhan, Siddharth. : OpinionFinder: a system for subjectivity analysis. In: HLT/EMNLP on Interactive Demonstrations, pp. 34–35. Association for Computational Linguistics, Morristown, NJ, USA (2005)

[8] Scaffidi, Christopher and Bierhoff, Kevin and Chang, Eric and Felker, Mikhael and Ng, Herman and Jin, Chun. : Red Opal: product-feature scoring from reviews. In: the 8th ACM conference on Electronic commerce, pp. 182–191. ACM, New York, NY, USA (2007)

[9] Das, Sanjiv and Mike Chen. : Yahoo! for Amazon: Extracting market sentiment from stock message boards. In: Asia Pacific Finance Association Annual Conf, (2001)

[10] Hu, M. and Liu, B. : Mining and Summarizing Customer Reviews. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168-177. ACM, New York, NY, USA (2004)

[11] John Lafferty, Andrew McCallum, and Fernando Pereira. In: International Conference on Machine Learning, pp. 282–289 ACM, New York, NY, USA (2001)

[12] Fuchun P. and Andrew McCallum. : Accurate information extraction from research papers using conditional random fields. In: Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA (2004)

[13] Fei S. and Fernando Pereira. : Shallow parsing with conditional random fields. In: the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pp. 134 - 141. Association for Computational Linguistics, Morristown, NJ, USA (2003)

[14] McCallum, A. : Efficiently inducing features of conditional random fields. In: the conference on Conference on Uncertainty in Artificial Intelligence, (2003)

[15] http://www.cs.cornell.edu/People/pabo/movie-review-data/review_polarity.tar.gz

[16] http://l2r.cs.uiuc.edu/~cogcomp/software.php

# Incremental Maintenance of Minimal Bisimulation of Cyclic Graphs

Jintian Deng[1], Byron Choi[1], Jianliang Xu[1], and Sourav S Bhowmick[2]

[1] Hong Kong Baptist University, China
{jtdeng,bchoi,xujl}@comp.hkbu.edu.hk
[2] Nanyang Technological University, Singapore
assourav@ntu.edu.sg

**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. We present a maintenance algorithm for a minimal bisimulation of a cyclic graph in the style of merging algorithm. Third, we propose a feature-based optimization to prune the computation on non-bisimilar SCCs. 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. Our experiment shows that our algorithm introduces a capability to handle cyclic graphs explicitly.

## 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., [3, 4, 6, 11, 16, 18, 23], have been derived from the notion of *bisimulation* equivalence. In addition to indexing, bisimulation has been adopted for selectivity estimation [13, 19, 20] and schemas for semi-structured data [2].

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 the left hand side of 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 persons; person in turn watches a few open_auctions. To model the bidding and watching relationships, open_auctions reference persons and vice versa. The references are encoded by IDREFs and represented by the dotted arrows in the figure. Two nodes in a data graph are bisimilar if they

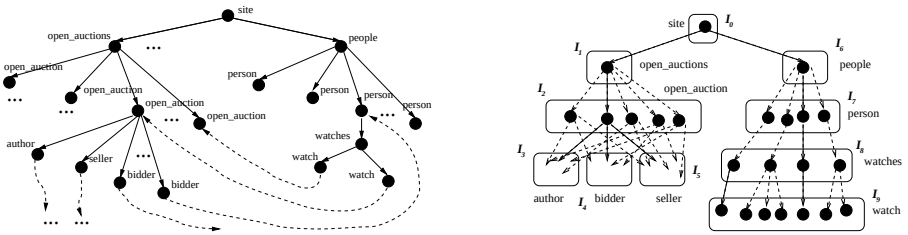



**Fig. 1.** Illustration – A sketch of `XMark` and its bisimulation

have the same set of incoming paths. A sketch of the bisimulation graph of `XMark` is shown in the right hand side of Figure 1. In the bisimulation graph, bisimilar data nodes are placed in a partition, denoted as $I_n$. Consider a query $q$ `/site/open_auction/seller` that selects all `seller`s of `open_auction`s. We can evaluate $q$ on the partitions and retrieve the data nodes in simply $I_3$. It is crucial to minimize the bisimulation graph for efficient index.

In practice, data graphs are often cyclic (e.g., [1]) and subjected to updates. Therefore, in contrast to other applications of bisimulation, its maintenance problem is much more important in database applications [12,21]. 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 are numerous bisimulation applications, there is relatively few work on its maintenance; (ii) Previous works [12,21] on maintenance of bisimulation of graphs mainly focus on directed *acyclic* graphs. These techniques do not explicitly handle cyclic structures.

In this paper, we take the first step to systematically and comprehensively investigate *incremental maintenance of minimal bisimulation in cyclic graphs*. The first contribution is a study of some properties of bisimulation of cyclic graphs, which influence the design of our algorithms (Section 4). There are two key challenges in the maintenance problem. Firstly, merging bisimulation algorithm as opposed to partition refinement is more natural for incrementally maintaining bisimulation. However, it is known [12] that merging algorithms fail to determine the minimum bisimulation of cyclic graphs. 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). Secondly, the nodes of `SCC`s must be considered *together*. In particular, a node of an `SCC` can be bisimilar to a node of another `SCC` *only if* the two `SCC`s are bisimilar.

Second, we present a maintenance algorithm for minimal bisimulation of cyclic graphs (Section 5). 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. We remark that our algorithm introduces an explicit handling of bisimulation between `SCC`s to the current state-of-the-art. As such, our algorithm *always* produces smaller (if not the same) bisimulation graphs when compared to previous works.

The third contribution is on our feature-based optimization for determining bisimulation between two `SCC`s (Section 6). On one hand, the computation of

bisimulation between two SCCs can be costly. On the other hand, there may not be many bisimilar SCCs, in practice. We aim at deriving structural features from SCCs such that two SCCs are bisimilar *only if* they have the same features. Specifically, we explore label- or edge-based, path-based, tree-based and circuit-based features, by studying their pruning power and maintenance efficiency.

The fourth contribution is our extensive experimental evaluation on our proposed technique (Section 7). It shows that (i) our algorithms scales well; (ii) the feature-based optimization prunes 50% computation on non-bisimilar SCCs on average; (iii) the path-based feature is the most effective; and (iv) our algorithm always produces a smaller bisimulation than previous works'.

## 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 [12, 21] for incremental maintenance of bisimulation of cyclic graphs. The algorithm proposed in [12] 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, [12] produces the minimum bisimulation of the updated graph. If the graph is cyclic, [12] returns a minimal bisimulation only. Thus, to support cyclic graphs, the minimum bisimulation is occasionally re-computed from scratch. [21] proposes a split-merge-split algorithm with a rank flag for SCCs, which is originally proposed in [5]. [21] also returns a minimal bisimulation in response to an update of a cyclic graph. However, there is neither experimental evaluation [21] nor implementation for us to perform comparisons. 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 [10] can be considered as a variant of Paige and Tarjan's algorithm [17] – 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 [10] produces a bisimulation that is always within 5% of the minimum bisimulation. It is shown, through a later experiment, that [12] may produce even smaller bisimulations, which we compare via experiments in Section 7.

Bisimulation (relation) [15] 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 [7], in the context of modeling checking. A conclusion of [7] 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.

## 3 Background

This section presents the background and the notations used.

**Definition 3.1.** *A* graph-structured database *(or* data graph*) 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 $\Sigma$ is a finite set of labels.*

For clarity, we may often denote a data graph as $G(V, E)$ when either of $r$, $\rho$ or $\Sigma$ are irrelevant to our discussions. Since our work focuses on cyclic graphs, we recall some relevant definitions below.

**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 `SCC`s of a graph can be determined by classical graph contraction algorithms, e.g., Gabow's algorithm, in $O(|V|+|E|)$, where each `SCC` is reduced to a supernode. The resulting graph is a *directed acyclic graph* `DAG`, which is often called the *reduced graph*. In subsequent discussions, we use `SCC`s to refer to non-trivial `SCC`s only. In the definition below, we hightlight two special kinds of nodes in `SCC`s namely, exit and entry nodes,

**Definition 3.2.** *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 $n \in V'$.*

**Bisimulation.** Next, we recall the relevant definitions of bisimulation.

**Definition 3.3.** *Given two graphs $G_1(V_1, E_1, r_1, \rho_1)$ and $G_2(V_2, E_2, r_2, \rho_2)$, an* upward bisimulation $\sim$ *is a binary relation between $V_1$ and $V_2$:*

$\forall\, v_1 \in V_1, v_2 \in V_2.\ v_1 \sim v_2 \rightarrow$
$\quad \forall\, (v_1', v_1) \in E_1.\ \exists\, (v_2', v_2) \in E_2.\ v_1' \sim v_2' \wedge \rho_1(v_1') = \rho_2(v_2') \wedge$
$\quad \forall\, (v_2'', v_2) \in E_2.\ \exists\, (v_1'', v_1) \in E_1.\ v_1'' \sim v_2'' \wedge \rho_1(v_1'') = \rho_2(v_2'').$

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

Definition 3.3 presents upward bisimulation in the sense that two nodes can be bisimilar only if their parents are bisimilar. The definition can be paraphrased in terms of paths, which is often convenient to simplify our discussions:

**Proposition 3.1:** *Two nodes are upward bisimilar if and only if the incoming path set 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 can be described as a set of partitions.

We should remark that there have been other notions of bisimulation, such as downward bisimulation and $k$-bisimulation, that have been applied in indexing/selectivity estimation but have not been the focus of this paper. Our techniques can be extended to support them with minor modifications.

In this work, we consider the notion of bisimulation minimality as defined in Definition 3.5. First, we recall the notion of *stability*.

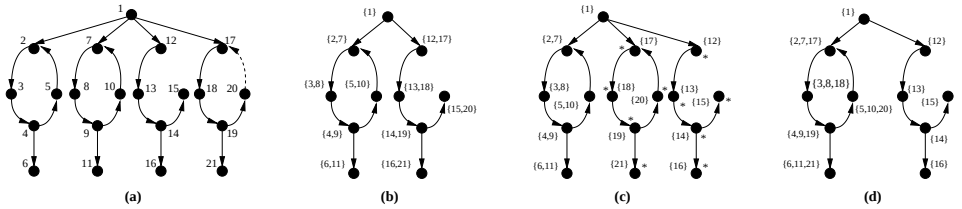



**Fig. 2.** (a) A cyclic data graph; (b) the minimal bisimulation graph; (c) the split bisimulation graph; and (d) an updated minimal bisimulation graph

**Definition 3.4.** *Given two partitions of nodes $X$ and $I$, $X$ is* stable *with respect to $I$ if either (i) $X$ is contained in the children of the nodes in the partition $I$ or (ii) $X$ and the children of the nodes in $I$ are disjoint.*

**Definition 3.5.** *Given a bisimulation $B$ of a graph $G$, $B$ is* minimal *if for any two 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 partition $K \in B$ unstable.*

**Definition 3.6.** *A bisimulation $B$ of a graph $G$ is the* minimum *bisimulation if $B$ contains the minimum number of partitions, among all bisimulations of $G$.*

Since the bisimulation $B$, which can be viewed as a graph, is used as indexes, the partitions are sometimes referred to as *index nodes*, or simply *Inodes*, whereas the nodes of the data graph are referred to as *data nodes*, or simply *nodes*.

**Bisimulation minimization.** Next, we illusrate the intuitions of merging algorithm for bisimulation minimization with a brief example shown in Fig. 2. Assume the nodes of the data graph shown in Fig. 2(a) have the same label. We show the node ids next to each node. We use {} to denote an inode. A merging algorithm initially places each node in a single partition. Assume that the algorithm merges pairs of paritions top-down, which attemps to merge Nodes 2 and 7. However, the algorithm has not yet determine Nodes 5 and 10. Hence, the algorithm does not return the minimum bisimulation shown in Fig. 2(b), unless it memorizes the `SCC`s containing Nodes 2 and 7 together.

## 4 Bisimulation of Cyclic Graphs

This section presents a minimization algorithm for bisimulation of cyclic graphs, shown in Figure 3, which is a component of the maintenance algorithm.

The algorithm can be divided into two parts. First, Lines 01-06, if $n_1$ and $n_2$ are not both in some `SCC`s, we compute bisimulation between $n_1$ and $n_2$ in the sytle of a merging 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 $G$. Then, we recursively invoke `bisimilar_cyclic`.

Second, if both $n_1$ and $n_2$ are in some `SCC`s, Lines 07-20 check if $S_1$ *and* $S_2$, as opposed to simply $n_1$ and $n_2$, can be bisimilar. We prune non-bisimilar `SCC`s by using the feature-based optimization presented in Section 6, in Line 08. For

**Procedure** `bisimilar_cyclic`
**Input:** Nodes $n_1$ and $n_2$ where $\rho(n_1) = \rho(n_2)$; $B$, the current bisimulation
**Output:** An updated bisimulation relation $B'$

```
01 if n1 and n2 are not both in some SCC
02     if ∀p1 ∈ n1.parent ∃p2 ∈ n2.parent s.t. p1 ∼ p2 then
03         add (n1, n2) to B
04         for all c1 in n1.next_nodes_top_order(G1)
05             for all c2 in n2.next_nodes_top_order(G2)
06                 B = bisimilar_cyclic(c1, c2, B)
07 else                    /* check bisimulation of the two SCCs */
08     assume n1 and n2 are in SCCs S1 and S2, respectively
       if feature_pruning(S1, S2) return B                  /* Sec. 6*/
09     clone S1 to S1';        create an artificial node n1' for n1
10     for all (n, n1) ∈ S1'.E
11         replace (n, n1) with (n, n1') ∈ S1'
12     clone S2 to S2';        create an artificial node n2' for n2
13     for all (n, n2) ∈ S2'.E
14         replace (n, n2) with (n, n2') ∈ S2'
15     clone B to B';     add (n1, n2) to B'                /* assume n1 ∼ n2 */
16     for all c1 in n1.next_nodes_top_order(S1')
17         for all c2 in n2.next_nodes_top_order(S2')
18             B' = bisimilar_cyclic(c1, c2, B')
19     if (n1', n2') in B' then B = B ∪ B'                  /* S1 ∼ S2 */
20 return B
```

**Fig. 3.** Bisimulation minimization of cyclic graphs

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 09-15. The main idea is illustrated with Fig. 4. 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 16-18). If we can construct a possible bisimulation between $n_1'$ and $n_2'$ (Line 19), then $S_1$ and $S_2$ are bisimilar.

The main idea of `bisimilar_cyclic` on handling SCCs is that `bisimilar_cyclic` explicitly breaks a cycle, whereas previous work *does not*. `bisimilar_cyclic` may be recursively called due to nested SCCs (Line 18). Without breaking cycles, the feature-based optimization (Line 07) may always derive features of the "topmost" SCC. As verified by experiments (Figures 7(b) and 7(c)), the features will be essential for pruning computation on non-bisimilar SCCs.

**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`. The inner loop can be performed in $O(log(|V|))$. The overall runtime is $O(|E|log(|V|))$.

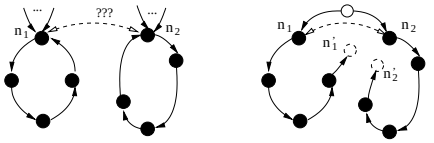



**Fig. 4.** Breaking one cycle in an SCC

## 5 Maintenance of Bisimulation

We present an overall maintenance algorithm in this section. For simplicity, we present an edge insertion algorithm `insert` in Figure 5. Edge deletions are discussed at the end of this section. Our algorithm consists of a split phase and a merge phase. 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. In the split phase, we mark the affected inodes, which will be examined in the merge phase.

Suppose the insertion makes the inode of $n_2$ unstable. 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 an 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 in $\mathcal{S}$ and $\mathcal{Q}$ recursively until they 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 to $\mathcal{S}$ and $\mathcal{Q}$ similar to Lines 03-04.

*(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 the pervious work [12] for acyclic graphs (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 03-04.

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 (Figure 3). An optimization is that apply merging on the inodes that are marked in the split phase.

*Example 5.1.* We illustrate Algorithm `insert` with an example. Reconsider the cyclic data graph is shown in Figure 2(a). Its minimal bisimulation is shown in Figure 2(b). 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

**Procedure** `insert`
**Input:** an insertion of an edge $(n_1, n_2)$ to a graph $G$; 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 $\mathcal{S} \neq \emptyset$
06  **if** $\mathcal{S} \neq \emptyset$ **then** /* split the relevant `SCC` */
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 $n_i$'s child, $n_i \in I_2$ 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 `SCC`s$\}$
13  **if** $\mathcal{Q} \neq \emptyset$ **then** /* split nodes not related to `SCC`s */
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 $\mathcal{I}$ /* [12] */
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 $n_i$'s child, $n_i \in I$ 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 `SCC`s$\}$
21 `Gabow`$(G')$ /* update the SCC information in $G'$ */
22 $(G', B')$ = `bisimilar_cyclic_marked`$(G, B)$ /* merging the marked inodes */
23 **return** $(G', B')$

**Fig. 5.** Insertion for minimal bisimulation of cyclic graphs

in Figure 2(c). Then, we update the `SCC` information of the data graph. By `bisimilar_cyclic_marked`, we obtain the bisimulation at Figure 2(d).

While the previous work [12] produces the same split graph (Figure 2(c)), it returns the bisimulation in Figure 2(c), due to the lack of the handling on `SCC`s as discussed in Section 4. Subsequently, any subgraphs that are connected to the `SCC` (Nodes 17-20), e.g., Node 21, are not merged, as the `SCC`s are not merged.

**Analysis.** The recursive procedure in Lines 05-20 traverses the graph $O(|E|)$. With optimization in [17], 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

The maintenance algorithm presented in Section 5 involves splitting the updated bisimulation into a non-minimal bisimulation. The non-minimal bisimulation is then minimized by merging. As discussed in the previous section, determining if two SCCs are bisimilar can be computationally costly $O(|E|log(|V|))$. In addition, in practice, SCCs may often be non-bisimilar. This motivates us to optimize bisimulation minimization of cyclic graphs by proposing features to prune computations on non-bisimilar SCCs. The main idea is to derive features of SCCs such that two SCCs can be bisimilar *only if* their features are the same or bisimilar. Futhermore, the features are ideally discriminative enough and can be efficiently derived and maintained.

### 6.1 Properties of Bisimulation of Cyclic Graphs

This subsection shows some properties of bisimulation of cyclic graphs. These properties show that a number of classic properties of graphs are not suitable for our feature-based optimization. We establish these properties with proof by contradictions. (The detaials can be found in the technical report [**?**].)

*Property 1.* SCCs with the same cycle height may not be bisimilar. SCCs with different cycle heights can be bisimilar.

*Property 2.* Two SCCs with the same number of simple cycles may not be bisimilar. Two bisimilar SCCs may have the different number of simple cycles.

*Property 3.* Two bisimilar SCCs with different numbers of entry nodes can be bisimilar.

The design of features exploits the following proposition on bisimulation of SCCs. The intuition is that as long as, we find some nodes in a SCC that is not bisimilar to any node in another SCC, the two SCCs will not be bisimilar.

**Proposition 6.2:** *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* $V_1$ *such that it is not bisimilar to any node in* $V_2$. □

### 6.2 Features of SCCs

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 SCC may affect other SCCs. We present some efficient features that gives the merging algoirthm some "lookahead" of SCCs to effeciently conclude that there is some node in a SCC that is not bisimilar to any nodes in other SCC (Proposition 6.2).

**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. 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 [14] and its maintenance is technically intriguing.

**Proposition 6.3:** *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, the longest paths of a cyclic graph cannot be determined in PTIME.

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 6.2). 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 $O(|V|^{1/3}log(|V|))$ [9]. 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

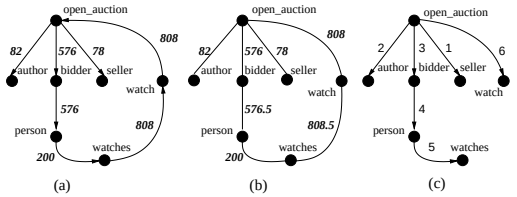



**Fig. 6.** The construction of the canonical spanning tree from a simplified `open_auction`

direction of the edges in the canonical spanning tree may differ from that of the edges in the original graph.

**Proposition 6.4:** *Two* `SCC`*s are bisimilar only if their minimum canonical spanning trees returned by Prim's algorithm are bisimilar.* □

It should be remarked that `SCC`s 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 6.2.* We illustrate the construction of a canonical spanning discussed above with an example shown in Figure 6. Figure 6(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 the edges, shown in Figure 6(b). Then, it is straightforward to compute the spanning tree (shown in Figure 6(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 [8]. Hence, one may be tempted to use circuit bases as a feature to prune computation on non-bisimilar graphs.

**Proposition 6.5:** *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.

### 6.3 Offline versus Online Feature Construction

Since the proposed features can be constructed efficiently, they may be constructed and used during bisimulation computation, i.e., runtime. During runtime, we may incorporate the features with the partial bisimulation constructed

so far for constructing features. Specifically, some nodes in SCCs have been associated with Inode. The id of Inodes together with the label, as opposed to the label alone, to build features.

In comparison, the features may be built the features offline and maintained with each update of the graph. However, given a cyclic graph, there can be exponientially many SCCs to the number of nodes of the graph, in the worst case. To build all possible features offline, we may determine features for each node, in the worst case. This size requirement may sometimes be prohibitive.

## 7 Experimental Evaluation

We present an experimental study on our algorithms. We modified the implementation of Ke *et al.* [12] to implement our algorithms. 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 [22] 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.

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.

Figures 7(a) and 7(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 SCCs of Large and Cyclic, we ran 100 graphs for each s.f. Figures 7(a) and 7(b) show that the runtimes are roughly linear to s.f. At the same s.f.

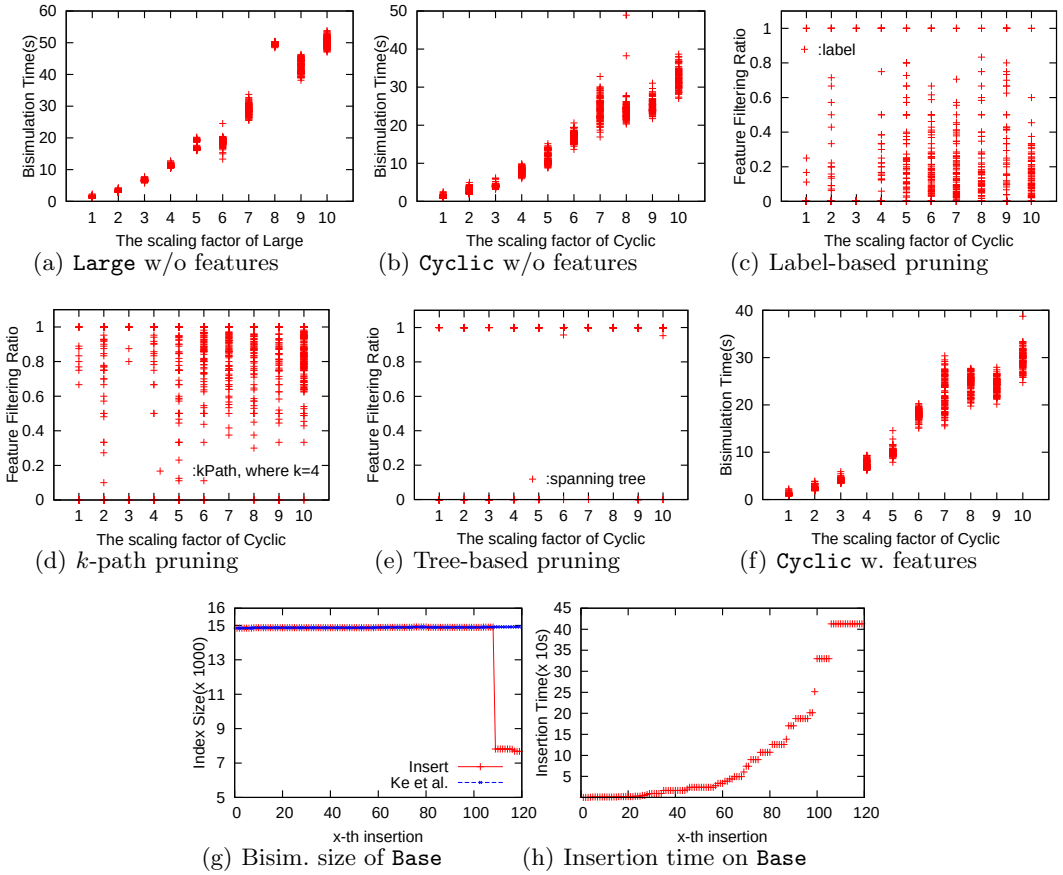



**Fig. 7.** Scalability test of the minimization algorithm on `XMark`; the effectiveness of features; and the efficiency of the maintenance algorithm

(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 7(c), 7(d) and 7(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 7(f) shows the runtime of `bisimilar_cyclic` with features. On average, it is 4% faster than

that without features (Figure 7(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 7(g). Figure 7(g) shows the size of the minimal bisimulation produced by `insert` and Ke *et al.* [12]. 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 [12] 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 7(g), `insert` identifies the two bisimilar `SCC`s that lead to a bisimulation graph roughly 100% smaller than the one produced by [12]. We remark that the performance difference (in terms of bisimulation size) between `insert` and [12] depends on the size of bisimilar `SCC`s and their bisimilar subgraphs are there in a graph.

The runtime of `insert` is shown in Figure 7(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 runtime of [12] 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. First, we presented a bisimulation minimization algorithm that explicitly handles `SCC`s. Second, we presented a maintenance algorithm for minimal bisimulation of cyclic graphs. Third, 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 unnecessary bisimulation computation. Our maintenance algorithm can return smaller bisimulation graphs than previous work, depending the sizeof bisimilar `SCC`s and their bisimilar subgraphs in the data graph.

## References


1. Batagelj, V., Mrvar, A.: Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/
2. Buneman, P., Davidson, S.B., Fernandez, M.F., Suciu, D.: Adding structure to unstructured data. In: ICDT (1997)
3. Buneman, P., Grohe, M., Koch, C.: Path queries on compressed XML. In: VLDB (2003)
4. Chen, Q., Lim, A., Ong, K.W.: D(k)-index: an adaptive structural summary for graph-structured data. In: SIGMOD (2003)

5. Dovier, A., Piazza, C., Policriti, A.: An efficient algorithm for computing bisimulation equivalence. Theor. Comput. Sci. 311(1-3), 221–256 (2004)
6. Fisher, D.K., Maneth, S.: Structural selectivity estimation for XML documents. In: ICDE (2007)
7. Fisler, K., Vardi, M.Y.: Bisimulation minimization and symbolic model checking. Form. Methods Syst. Des. 21(1), 39–78 (2002)
8. Gleiss, P.M., Leydold, J., Stadler, P.F.: Circuit bases of strongly connected digraphs. Working Papers 01-10-056, Santa Fe Institute (2001), `http://ideas.repec.org/p/wop/safiwp/01-10-056.html`
9. Henzinger, M.R., King, V.: Maintaining minimum spanning trees in dynamic graphs. In: ICALP (1997)
10. Kaushik, R., Bohannon, P., Naughton, J.F., Shenoy, P.: Updates for structure indexes. In: VLDB (2002)
11. Kaushik, R., Shenoy, P., Bohannon, P., Gudes, E.: Exploiting local similarity for indexing paths in graph-structured data. In: ICDE (2002)
12. Ke, Y., Hao, H., Ioana, S., Jun, Y.: Incremental maintenance of XML structural indexes. In: SIGMOD (2004)
13. Li, H., Lee, M.L., Hsu, W., Cong, G.: An estimation system for XPath expressions. In: ICDE (2006)
14. Mendelzon, A.O., Wood, P.T.: Finding regular simple paths in graph databases. In: VLDB (1989)
15. Milner, R.: Communication and Concurrency. Prentice Hall (1989)
16. Milo, T., Suciu, D.: Index structures for path expressions. In: ICDT (1999)
17. Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM J. Comput. 16(6), 973–989 (1987)
18. Polyzotis, N., Garofalakis, M.: XCluster synopses for structured XML content. In: ICDE (2006)
19. Polyzotis, N., Garofalakis, M.: XSketch synopses for XML data graphs. ACM Trans. Database Syst. 31(3) (2006)
20. Polyzotis, N., Garofalakis, M., Ioannidis, Y.: Approximate XML query answers. In: SIGMOD (2004)
21. Saha, D.: An incremental bisimulation algorithm. In: FSTTCS (2007)
22. Schmidt, A., Waas, F., Kersten, M., Carey, M.J., Manolescu, I., Busse, R.: XMark: A benchmark for XML data management. In: VLDB (2002)
23. Spiegel, J., Polyzotis, N.: Graph-based synopses for relational selectivity estimation. In: SIGMOD (2006)

# Selectivity Estimation of Twig Queries on Cyclic Graphs

Yun Peng, Byron Choi, Jianliang Xu

*Department of Computer Science, Hong Kong Baptist University, Hong Kong, China*

{ypeng, bchoi, xujl}@comp.hkbu.edu.hk

## Abstract

*Recent applications including the Semantic Web, Web ontology and XML have sparked a renewed interest on graph-structured databases. Among others, twig queries have been a popular tool for retrieving subgraphs from graph-structured databases. To optimize twig queries, selectivity estimation has been a crucial and classical step. However, the majority of existing works on selectivity estimation focuses on relational and tree data. In this paper, we investigate selectivity estimation of* twig queries *on* possibly cyclic *graph data. To facilitate selectivity estimation on cyclic graphs, we propose a matrix representation of graphs derived from prime labeling — a scheme for reachability queries on directed acyclic graphs. With this representation, we exploit the* consecutive ones property (C1P) *of matrices. As a consequence, a node is mapped to a point in a two-dimensional space whereas a query is mapped to multiple points. We adopt histograms for scalable selectivity estimation. We perform an extensive experimental evaluation on the proposed technique and show that our technique controls the estimation error under 1.3% on* XMARK *and* DBLP*, which is more accurate than previous techniques. On* TREEBANK*, we produce* RMSE *and* NRMSE *6.8 times smaller than previous techniques.*

## 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. Up-to-date, there has already been voluminous real-world (possibly cyclic) graph-structured data [3]. 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 support for query optimization technique in databases. In particular, selectivity estimation has been built into the query optimizer of all commercial relational databases. In a nutshell, given a query, we want to determine the number of results of the query, without invoking potentially costly query evaluation. However, the majority of previous research on selectivity estimation, with a few exceptions (see Section 2), focuses on relational and tree-structured data. In this paper, we propose *histogram-based selectivity estimation of twig queries for possibly cyclic graphs.*

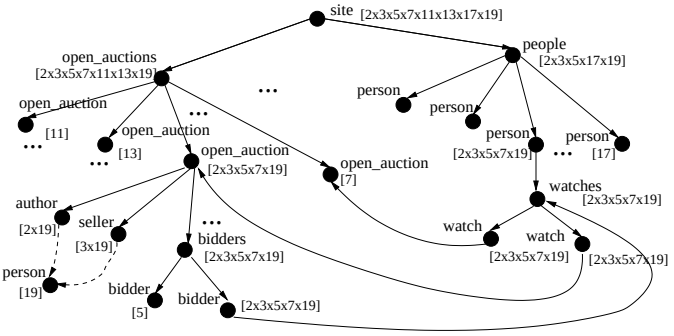



**Figure 1. An example graph of auction information (XMARK) with its prime labeling**

Twig queries have been a popular and classical tool for retrieving subgraphs from a graph-structured database. To facilitate our technical discussions, let us consider a twig query over a simplified XMARK (cyclic) graph [18]. The graph (Fig. 1) encodes auction information, where people watch over auctions and bidders bid items. Consider a sample query `//person[//open_auction//person]`, which selects the `person`s who watch some auctions that are still open and related to some other `person`s. Note that the twig query is recursive, where the query selects `person`s that have some `person` descendants. In XMARK with a scaling factor 1.0, there are 25,500, 12,000, and 13,192 `person`s, `open_auction`s and `open_auction//person`s, respectively. Based on selectivity information, a query optimizer should produce a plan that evaluates `//open_auction//person` prior to `//person`, to minimize intermediate result size.

Recently, there have been studies on selectivity estimation of XML data, *a.k.a.* tree data, and twig queries. In particular, Wu *et al.* [24] propose to adopt histograms for

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. That is, each node of a tree is associated with one interval. The challenge in adopting the interval representation to cyclic graphs (and even directed acyclic graphs) is that multiple intervals may be associated with a node [2], as there may be muliple paths between any two nodes. Thus, the storage requirement of this technique on cyclic graphs can be prohibitive. In addition, it does not appear straightfoward to extend the existing estimation framework [24] to support multiple interval representation either.

Regarding cyclic graphs, there has been a work, namely XSKETCH [16], that exploits (local) minimal bisimulation of a graph for selectivity estimation of path queries. When compared to the histogram approach, bisimulation is not yet available in any commercial database and there has not been a *de facto* external representation of bisimulation graphs. Another drawback of local bisimulation is that the estimation accuracy relies on a strong statistical assumption (uniform distribution) of data.

In this paper, we propose a novel selectivity estimation technique for twig queries on cyclic graphs. The novelties of the technique are twofold. First, different from [16], we undertake a histogram approach to conduct selectivity estimation; we use auxiliary histograms to tackle possibly skewed data and do not make any assumption on the data distribution. To facilitate summarization of data, we propose a prime number labeling scheme (or simply *prime labeling*) to represent (cyclic) graph data, which was originally proposed for tree data [23]. (We defer the discussion on the drawbacks of other alternative representations to Section 2.) With prime labeling, the checking of the descendant-ancestor relationship among nodes of graphs and (later) estimation methods become very simple. Specifically, previous prime labeling scheme essentially associates each node with an exclusive prime number and labels each node with the product of its children's labels and its own prime number. Reachability between nodes is simply mapped to divisibility test of labels. Unlike previous works, our prime labeling requires fewer prime numbers for labeling and is therefore smaller in size.

A known issue of prime labeling is that it often results in very large integers. The second novelty lies in a new binary matrix representation of prime labeling, which further reduces the labeling size. In this way, we bridge selectivity estimation to the work on matrices. In particular, we transform a cyclic graph into a matrix with the *Consecutive Ones Property (C1P)*. Subsequently, a node of a cyclic graph can be represented as an interval of column IDs, (*start*, *end*).[1] Querying is then done by logical operations on the matrix. Nodes are essentially summarized in a two-dimensional histogram. In matrix transformations, new columns are often introduced. We store mappings between equivalent column IDs in a compressed form. Given a query, we translate it into multiple equivalent queries (intervals) with the compressed mappings. Such interval representations of data and queries make histograms a feasible solution for summarization.

The contributions of this paper are as follows.

- To the best of our knowledge, this is the first work on selectivity estimation of twig queries on cyclic graphs. Previous works focus on either twig queries or cyclic graphs but not both.

- We propose a prime labeling scheme to represent cyclic graphs and a binary matrix representation of prime labeling (Section 5). We transform the matrix in order to map a node of a graph to an interval and in runtime, a query to possibly multiple intervals. A two-dimensional histogram is used to summarize the matrix (Section 6). We propose an estimation algorithm with the histograms (Section 7).

- We perform a performance evaluation (Section 8) that verifies our technique controls the estimation error under 1.3% for XMARK and DBLP datasets . In comparison, the previous work XSKETCH/TREESKETCH [16, 17] reports that it controls the error under 5%. On TREEBANK dataset, our implementation produces RMSE and NRMSE that are at least 6.8 times smaller than XSEED's [25].

## 2 Related Work

There have been some recent works on selectivity estimation for path or twig queries on trees or cyclic graphs. The techniques can be roughly classified into two categories: graph-based approach and relational-based approach.

**Graph-based approach.** While graph-structured data model has its root at network data model, it was revisited in Tsimmis project, in which Object Exchange Model (OEM) is proposed. DATAGUIDE [14] is proposed to summarize the paths of OEM. Graphs are considered as NFA and their DATAGUIDEs are DFA of the graphs. DATAGUIDE has been extended to support approximate query processing [8]. Straight-Line Grammar (STL) [6] is a special form of context-free grammar, for summarizing a data graph. To reduce the size of the grammar, [6] proposes to use a wildcard to simplify some non-terminals in a production.

[1] Our interval represents the column IDs of a C1P matrix. In contrast, the multiple intervals of nodes in [2] represents the preorder and postorder numbers of a traversal on the spanning tree of a DAG and the connectivity due to non-tree edges.

Another graph-based approach [16, 17] (XSKETCH and later TREESKETCH) is derived from bisimulation of graphs. XSKETCH supports only path queries on cyclic graphs. TREESKETCH, on the other hand, supports twig queries on acyclic graphs only. In comparison, we support twig queires on cyclic data. [16, 17] propose to adopt bisimulation as the synopses of a data graph. To further reduce the size of bisimulation, a notion of local bisimulation [10] has been applied. To recover the path information from a local bisimulation graph, graph stability is exploited and uniform distribution of nodes is assumed. Unlike their techniques, we do not assume the data exhibits uniform distribution but use auxiliary histograms to summarize skewed data. A recent survey shows that some popular graph (XML) benchmarks contain highly skewed data [13]. Our overall technique adopts histogram, which is by far the most popular selectivity estimation technique.

Correlated subpath tree (CST) [4] stores the count of small twigs (*branches*) in data trees. It has been shown by recent experiments that [16, 17] outperform [4]. XSEED [25] initially derives a compact path summary (kernel) from data trees and adaptively tunes memory budgets of summaries based on query workload. Its experiment showed XSEED outperforms TREESKETCH [17] when 1000 queries are considered. In our experiments, we compare our techniques with XSEED. STATIX [7] proposes to count subtrees in XML, not cyclic graphs, with schema information. In contrast, we do not assume schemas.

**Relational-based approach.** Histograms from relational databases have been adapted to support selectivity estimation of queries on graphs. [24] proposes an interval representation of nodes of a tree. The start and end position of the interval is used as the $x$ and $y$ coordinates of a point in a two-dimensional plane. A two-dimensional histogram and auxiliary histograms are used to summarize the points. Bloom histograms [21], path trees and Markov tables [1,12] have been proposed for path selectivity estimation for tree data. However, it is not clear how these techniques support cyclic graphs, which contain infinitely many paths, for selectivity estimation.

**Alternative representation of graphs.** In this work, we adopt prime labeling [22, 23] as the representation of cyclic graphs, due to its simplicity. In addition to the interval representation discussed earlier, there have been alternative representations. Transitive closure of the graph $G$ consists of an entry $(u, v)$ if $u$ can reach $v$ in $G$. However, its storage is prohibitive $O(|G|^2)$. Adjacency matrix has been a classical representation of graphs. However, determining the ancestor-descendant relationship in an adjacency matrix is relatively complex, which requries taking self-products of the matrix. There has been a host of ad-hoc indexes for reachability queries on graphs, *e.g.*, 2-hop labeling [5]. However, the structures of ad-hoc indexes are often complex and their summarization does not seem to be straightforward.

# 3 Definitions and Preliminaries

We begin our technical discussions with the definitions and 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, $\Sigma$ is a set of tags and $\lambda$: $V \rightarrow \Sigma$ is a function that returns the tag 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 irrelevant.

## 3.2 Twig Queries

Among the queries on graphs, XPATH has been studied more extensively recently than others and it has been an indispensable part of eXtensible Markup Language (XML) — the *de facto* standard for electronic data exchange. Hence, we consider a fragment of structural XPath — *twig queries*. The syntax is given in BNF below:

$$p \quad ::= \quad \epsilon \mid A \mid * \mid // \mid p/p \mid p[q],$$
$$q \quad ::= \quad p \mid q \wedge q \mid q \vee q,$$

where $\epsilon$, $A$, $*$ and $/$ denote the *self-axis*, a tag, a wildcard and the *child*-axis, respectively; $//$ stands for */descendant-or-self::node()/*; and $q$ in $p[q]$ is called a *filter*, in which $\wedge$ and $\vee$ denote conjunction and disjunction, respectively. For $//$, we abbreviate $p_1/\ //$ as $p_1//$ and $//\ /p_2$ as $//p_2$. For simplicity, our technical discussion focuses on // axes, while the extension to / axes can be addressed by introducing an index on the depth of nodes. We use $r[|p|]$ to denote the evaluation of the query $p$ from the node $r$.

**Problem statement.** Let $R$ be the set of nodes of the evaluation, where $R$ = $r[|p|]$. In this paper, given $p$ and $r$, we want to determine $|R|$ efficiently and accurately.

## 3.3 Consecutive Ones Property

Next, we provide the definition of the *Consecutive Ones Property* (C1P), which is useful to summarize the ones (non-zeros) in a matrix. In this paper, we represent a cyclic graph with a *binary matrix*, denoted by $M$. The $u$-th row is denoted as $M[u]$. The entry at the $u$-th row and the $v$-th column, denoted as $M[u][v]$, can be either '0' or '1'.

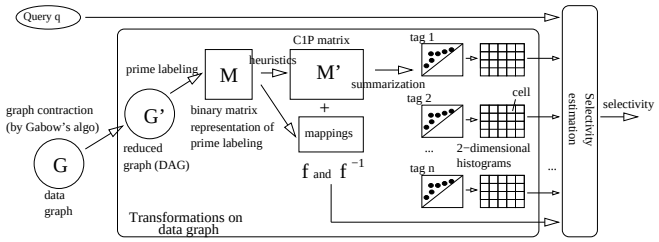



**Figure 2. An overview of our proposed technique**

**Definition 3.1:** 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. ∎

For simplicity, we call a matrix with strong C1P a *CIP matrix*. Since the ones in a row of a C1P matrix are adjacent, we can represent the ones in the row with a start and end column number which can be considered as the $x$- and $y$-coordinates of a data point. In subsequent discussions, we may use the term intervals and data points interchangeably.

# 4 Overview

In this section, we provide an overview of the design of our proposed technique to the selectivity estimation problem.

Consider a cyclic data graph $G$, as shown in Fig. 2. $G$ is first reduced into a DAG ($G'$), on which prime labeling is applied. We design a prime labeling scheme that not only facilitates simple query processing but also selectivity estimation of twig queries. Furthermore, we propose a new binary matrix representation $M$ of prime labeling. Its advantages are twofold. First, prime labeling may require large prime numbers for labeling large graphs since at least each leaf node needs a unique prime number. Second, we can adopt existing work from matrices, for summarization. In particular, we transform the binary matrix $M$ into a C1P matrix $M'$ and represent graph nodes with two-dimensional data points. As a consequence, a well-known estimation technique, two-dimensional histogram, can be used to summarize the data points of each kind of tag, where the histogram's cell size $\rho$ can be tuned for space or estimation accuracy.

Regarding estimating the selectivity of a twig query $q$, we encounter a unique challenge that data points derived from the C1P matrix $M'$ are often highly skewed. When developing the estimation algorithm, we observe that there are sometimes few queries with large errors, which lead to a poor overall accuracy. Hence, we propose additional auxiliary histograms for summarizing skewed data points and do not opt to adopt [24] for selectivity estimation.

In what follows, we discuss the prime labelling and binary respresentation of cyclic graphs in Section 5, matrix transformations in Section 6, and selectivity estimation of twig queries in Section 7.

# 5 Representation of Cyclic Graphs

Most labeling techniques on descendant-or-self axis focus on tree data. It is not clear how these labelings can be modified to support cyclic graphs. For example, path-based labelings do not work in cyclic graphs as there are infinitely many paths and the interval labeling [2] has a high space requirement for cyclic graphs. In this section, we present a new representation of cyclic graphs based on prime labeling [23], to efficiently estimate the descendant-or-self axis on cyclic graphs.

## 5.1 The Original Prime Labeling

Prime labeling was originally proposed for indexing trees. The main idea of prime labeling is that each node is labeled with a product of prime numbers such that the ancestor-descendant relationship between nodes could be determined by using the division of the prime labels. A node $n_1$ is an ancestor of another node $n_2$ if and only if the label of $n_1$ is *divisible* by that of $n_2$. In [23], a unique prime number is assigned to each leaf node. The prime label of an internal node is the product of the prime labels of its children. Such labeling works on trees only. To extend prime labeling to DAGs, [22] requires a unique prime number per node.

## 5.2 Prime Labeling for Cyclic Graphs

To support cyclic graphs, we follow the standard preprocessing to reduce each SCC into a supernode and apply prime labeling on the reduced graph. We propose two modifications on prime labeling: First, the previous work [22] on prime labeling uses excessive prime numbers (one prime number per node). We propose to use fewer number of (unique) prime numbers needed for labeling and hence reduce the overall size of prime labeling. More specifically, we require a new unique prime number for labeling a node $n$ in one of the following scenarios: (i) $n$ is a leaf node; or (ii) all the children of $n$ have more than one parents. Regarding the second scenario, if a new prime number is *not* used for labeling $n$, then it is possible to have a node $n'$ whose label is divisible by $n$'s label but $n'$ is not an ancestor of $n$, since $n'$ can be an ancestor of other parents of $n$'s children. Second, prime labeling needs to support possibly multiple strongly connected components (SCCs) in cyclic graphs. By definition, each node in an SCC can reach any other node in

the SCC. Therefore, the nodes in an SCC can be associated with the same prime label.

Next, we present the definition of prime labeling for cyclic graphs. Let `get_next()` be a special function which returns a prime number that has not been returned before. Assume that a cyclic graph $G$ has been preprocessed by Tarjan's algorithm [20], where each SCC is reduced to a supernode. Denote the reduced graph to be $G'(V', E')$. Each node $n$ is associated with a prime label $\ell$ as defined below.

**Definition 5.1:** The *prime label* $\ell$ 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 all the children of $n$ have multiple parents, then $n.\ell$ = `get_next()` $\times \prod_{c \in C} c.\ell$, where $C$ is the set of $n$'s children.
3. Otherwise, $n.\ell = \prod_{c \in C} c.\ell$.

■

The prime labels of the nodes of a reduced graph $G'$ are assigned in a reverse-topological order, *i.e.*, a bottom-up traversal. The pseudo-code of the prime labeling construction (`prime-construct` in Fig. 3) can be readily dervied from Definition 5.1. It assigns prime labels to the reduced graph ($G'$). In Line 01, we apply Tarjan's algorithm to reduce a cyclic graph $G$ into a DAG $G'$, where an SCC is reduced to a supernode. We initalize the prime label of each node to be 1 in Line 02. Then, we assign the prime labels of nodes in a reverse-topological order (bottom-up traversal). There are two possible cases. (1) If the node $n$ is a leaf node (Lines 04-05), we assign a new prime number to the node. (2) If the node $n$ is not a leaf node, the prime labels of the node is set to the product of the prime labels of its children in Lines 07-08. However, if all the children of the node have multiple parents, we assign an additional new prime number to the prime label of $n$ (Lines 09-10), as argued earlier.

While `prime-construct` and [22, 23] assign prime numbers differently, querying with our prime labeling remains simple. Assume that we have a set of $A$-nodes $S_A$ and $B$-nodes $S_B$. 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 by $M_A$, and then check the divisibility between $M_A$ and the prime label of each node in $S_B$. This requires $O(|S_a| + |S_b|)$ only.

**Example 5.1:** Reconsider the XMARK graph shown in Fig. 1. The prime label of each node is shown in the square bracket. We show a strongly connected component whose nodes have the same label 2×3×5×7×19, as they can reach one another, by definition. The `person` with label 19 is both a `seller` and an `author`. We use a new prime label for the `author` (2) and `seller` (3). Since 2×19 and 3×19 are not divisible, `author` and `seller` are not a descendant of each other. Furthermore, to check the number of `person`s that are a descendant of some `open_auction`s, we can simply check the divisibility between the `person`'s label, *e.g.*, 17, to the label of `open_auction`s, *i.e.*, 2×3×5×7×11×13×19. ■

```
Input: A data graph G
Output: A data graph with prime labeling
01 G' = tarjan(G)
02 initialize the prime label of nodes in G' to 1
03 for each n in G'.V in reverse topological order
04   if n is a leaf node                /* Definition 5.1 */
05     n.ℓ = get_next()
06   else
07     for each c in n.children
08       n.ℓ = n.ℓ × c.ℓ
09     if ∀ n' ∈ n.children. n' has multiple parents
10       n.ℓ = get_next() × n.ℓ
```

**Figure 3. Prime labeling construction** `prime-construct`

## 5.3 Matrix Representation of Cyclic Graphs

Given voluminous graph data, such as biology pathways, social networks and XML, prime labeling may result in very large integers. To address this issue, we propose a binary matrix representation of prime labeling and map integer divisions simply to logical operators of vectors.

**Definition 5.2:** Suppose that the prime label $\ell$ 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. $\ell$ 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 $\vec{\ell}[i_j]$ = 0 otherwise. The size of the vector is the total number of prime numbers used in labeling $G$. ■

A graph is represented as a set of binary vectors which form a matrix. Here, we always discuss binary vectors and matrices. For simplicity, we may omit the term "binary".

With this representation, divisions and multiplications of prime labels can be mapped into logical operators on the vector representation of the prime labels.

**Definition 5.3:** 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 5.3 can be alternatively 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 5.4:** 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 $\neg(\bigwedge_{n \in V} n.\vec{\ell}) \wedge n_2.\vec{\ell} = \vec{0}$. ∎

To end this section, we remark that `prime-construct` (presented in Fig. 3) can be used with minor modifications to compute the binary matrix representation *directly* from cyclic graphs.

## 6 Matrix transformations

In this section, we present the transformation of the binary matrix of prime labeling into a C1P matrix for simple summarization. On one hand, a C1P matrix can be readily summarized by a set of intervals, as discussed in Section 1. On the other hand, converting a matrix into a C1P matrix is intractable [19]. Worst still, there is no polynomial time approximation scheme for determining a C1P submatrix in a given matrix. Therefore, we propose (i) a heuristic algorithm for converting a matrix into a C1P matrix and (ii) two practical optimizations, namely, horizontal decomposition on the matrix and extraction of the largest common subset of non-zeros in the decomposed submatrices, to reduce the size of the matrix passed to the heuristic algorithm.

### 6.1 Transforming to C1P Matrix

The heuristic algorithm uses a C1P detection algorithm proposed by Hsu [9] as a component, which determines if a matrix is C1P or not and has been known to have simple implementations. The overall heuristic algorithm `heuristic_c1p` is presented in Fig. 4.

We assume that the rows of the input matrix $M$ are assumed to be sorted by the number of non-zeros in descending order. The heuristic algorithm is to first process the rows that have more overlapping non-zeros with the first row. The idea is that there may be a higher chance for such rows to share more columns containing non-zeros. Subsequently, we may obtain a smaller C1P matrix.

The details of `heuristic_c1p` are as follows. We first compute the overlappings between the rows in $M$ with the first row — the row with the most number of non-zeros (Lines 01-03). Then, we sort the rows by the amount of overlappings (Line 04) and construct a new C1P submatrix $R$ (Line 05). We may merge a row $M[i]$ into $R$ if one of the three conditions is satisfied (Lines 07-09): (i) Hsu's algorithm (denoted as `c1p_detect`) reports that $M[i]$ can be merged to $R$ to form a C1P matrix. (ii) $M[i]$ does not overlap with $R$. (iii) $M[i]$ is contained in some rows in $R$. We remark that Conditions (ii) and (iii) do not arise in [9]. However, such a row can be readily merged into the C1P matrix $R$.

In Line 11, `column_partition` is the COLUMN-PARTITION algorithm in [9] extended to handle Conditions

**Procedure** `heuristic_c1p`
**Input**: A matrix representation of a cyclic graph $M[\,][\,]$, where the rows of $M$ are sorted by # of non-zeros (descending)
**Output**: The C1P matrix from $M$

01 $r = M[0]$ /* 1st row */
02 **for each** $i$ **in** [1...$m$-1], where $m$ is the number of rows of $M$
03 $M[i]$.`overlap` = $|\{\, j \mid M[i][j] \wedge r[j],\ j \in [1...n]\}|$
04 sort $M$ by the `overlap` attribute of the rows
05 $R = \{r\}$
06 **for each** $i$ **in** [1...$m$-1]
07 **if** (i) `c1p_detect`$(R \cup \{M[i]\})$ **or** /* [9]*/
08 (ii) $M[i] \wedge \bigwedge R = \emptyset$ **or** /* non-overlapping row*/
09 (iii) $\exists\, j$ s.t. $M[i] \wedge R[j] = M[i]$ /* $M[i]$ in $R[j]$*/
10 **then**
11 $R$ = `column_partition`$(R, M[i])$ /* Section 6.1 */
12 **return** $R \oplus$ `heuristic_c1p`$(M - R)$

**Figure 4. Heuristic C1P transformation**

(ii) and (iii). In a nutshell, assuming that $R$ and $r$ form a C1P matrix, `column_partition`$(R, r)$ reorganizes the columns of $R$ and $r$ in partitions such that a C1P matrix can be trivially generated from the partitions. Due to space constraints, we opt to present COLUMN-PARTITION as a black box.

Finally, we recursively call `heuristic_c1p` to process the remaining rows, until the whole matrix is transformed into a C1P matrix (Line 12). A subtle note is that the C1P submatrix $R$ constructed from each call of `heuristic_c1p` is often not mergable to each other. Otherwise, these submatrices may be returned in a single call of `heuristic_c1p`. Hence, we append (denote as $\oplus$) the submatrix, returned from recursive calls, to $R$.

The operator $\oplus$ is a special append operator. Suppose $R_1$ and $R_2$ is a $n_1$ by $m_1$ matrix and $n_2$ by $m_2$ matrix, respectively. $R_1 \oplus R_2$ returns a ($n_1$ + $n_2$) by ($m_1$ + $m_2$) matrix $R'$, where $R_1$ and $R_2$ are placed at the top-left and bottom-right corner of $R'$, respectively. Fig. 5 illustrates $\oplus$ and the heuristics `heuristic_c1p` with a sketch of the run of `heuristic_c1p`. (A real example of a C1P matrix produced by `heuristic_c1p` is presented in Fig. 7.) `heuristic_c1p` generates a C1P matrix recursively.

**Mappings between columns and positions.** In general, a column of a matrix may be duplicated in multiple submatrices returned by `heuristic_c1p`. To avoid confusions, we refer the columns of the C1P matrix to *positions*. Two mappings are needed to record the relationship between the column and its positions. In particular, we store the mappings in two binary relations $f$ and $f^{-1}$, where $f(v_i)$ returns the positions of $v_i$ in $V$ and $f^{-1}(p)$ returns $v_i$ where $p \in f(v_i)$.

**Analysis.** The runtime of Hsu's algorithm (`c1p_detect`

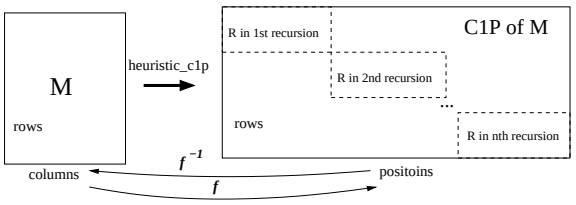



**Figure 5. Schematics of heuristics** `heuristic_c1p`

and `column_partition`) is $O(m+n+r)$, where $m$, $n$ and $r$ are the number of rows, the number of columns and the number of non-zeros. The loops (Lines 02 and 06) iterate through the rows of a matrix. There are at most $m$ recursive calls. Thus, the time complexity of `heuristic_c1p` is $O(m^2 \times (m+n+r))$.

## 6.2 Optimizing Matrix Transformation

This subsection presents two optimizations for matrix transformation that are specific to our approach.

We make two observations on the matrix representation of a cyclic graph, constructed as in Section 5. First, the prime labels are assigned essentially bottom-up, where the rows (the nodes) near the root have relatively more non-zeros. That is, the number of non-zeros of the rows (the nodes) near the root is often very different from those near the leaf nodes. Second, since the nodes in an SCC can reach one another, the rows of an SCC are identical. Hence, we propose two matrix manipulations to reduce the size of the matrix passed to `heuristic_c1p`.

### 6.2.1 Horizontal Matrix Decomposition

The patterns in the rows with many non-zeros are often different from those with few non-zeros. We propose a simple decomposition to separate these rows of a matrix $M$ and summarize them separately. First, note that the order of rows does not carry any information. We sort the rows by the number of non-zeros, in descending order. Denote by $\bar{M}$ and $\sigma$, respectively, the mean and standard deviation of the number of non-zeros for all rows of $M$. Second, we scan the sorted matrix. Let $R$ be the rows scanned thus far and $r$ be the next row in the scan. If the number of non-zeros of $r$ is beyond $\bar{R} - c\sigma$, where $c$ is a constant, *e.g.*, 3, this indicates the remaining rows in the matrix are significantly different from those in $R$. Hence, we decompose the matrix at $r$ and then continue the scan.

### 6.2.2 Common Pattern Extraction

A pattern that appears in *all* rows of a matrix contains little information. Therefore, we extract the largest common pattern of a matrix in a scan of the matrix, which can be incorporated with the decomposition discussed above. In the scan, we maintain the current common pattern $P$ of the scanned rows. Assume $r$ is the next row in the scan. The next largest common pattern is simply defined as $P \wedge r$.

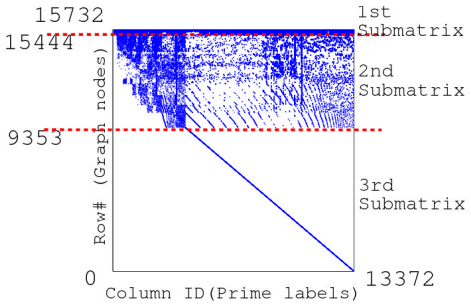



**Figure 6. Matrix representation of** XMARK

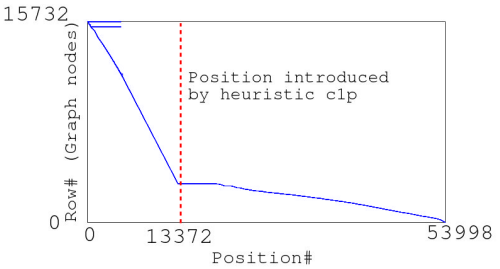



**Figure 7.** C1P **matrix representation of** XMARK

**Example 6.1:** In Fig. 6, we show the matrix representation of XMARK (with the scaling factor 0.01) after sorting the rows by the number of non-zeros (prime numbers). A dot '.' and a blank space ' ' represent a non-zero and zero, respectively. The figure shows that there are three distinguishable submatrices with different non-zero densities. The dotted lines show the decomposition when $\bar{R} - 3\sigma$ is used. Most of the non-zeros of the matrix occur in the topmost submatrix. After we locate the common pattern in the submatrix, we extract it out from the submatrix. We find that the topmost submatrix has a large common pattern, containing 10,684 non-zeros. Finally, we apply `heuristic_c1p` on the decomposed matrices to obtain a C1P matrix shown in Fig. 7. The number of positions needed for 13,372 columns is 53,998. ∎

# 7 Selectivity Estimation

This section presents the details of using two-dimensional histograms to summarize the C1P matrix derived in Section 6 and perform selectivity estimation. We first discuss our data structures associated with the histograms (Section 7.1) and the overall estimation algorithm (Section 7.2) and then highlight its technical details (Sec-

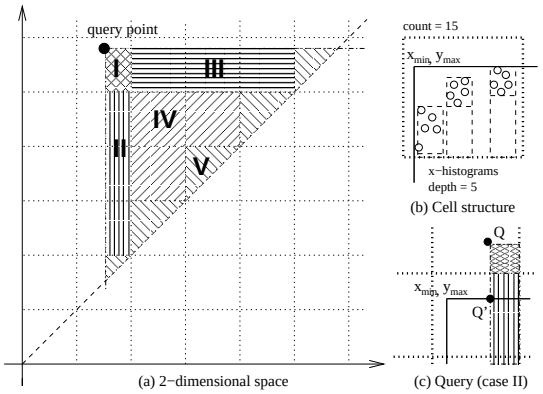



**Figure 8. (a) Two-dimensional histogram; (b) Cell structure; and (c) Intermediate query**

tions 7.3).

### 7.1 Two-dimensional Histograms

As discussed in Section 3.3, a C1P matrix can be represented by a set of (two-dimensional) data points $(x, y)$'s, where $x$ and $y$ are the start and end columns of 1's. We use a two-dimensional histogram for summarizing the data points of each kind of tag. The two-dimensional space is covered by a grid, which consists of non-overlapping cells. An example is shown in Fig. 8(a). The cell size is controlled by a parameter $\rho$. A data point $(x, y)$, where $x \leq y$, resides in the upper diagonal area of the grid. To reduce space, we only consider the cells with data point(s).

Through our experiments, we found that the data points of our benchmark datasets are not evenly distributed on the two-dimensional space. First, the data points are skewed towards the diagonal line. Such phenonmenon has also been found in the data points of the interval approach [24], though the definition of their interval is different from ours. Second, the estimation rules in [24] assume data points are uniformly distributed along the diagonal line, which essentially integrate the area of possible regions containing some answer. In contrast, we associate three auxiliary data summaries to each cell to tackle skewed data points. An example of the cell structure is shown in Fig. 8(b). The auxiliary summaries are discussed below.

1. We introduce a tight bounding rectangle, defined by $(x_{min}, y_{max})$, of the data points of a cell, where $x_{min}$ is the smallest $x$-coordinate and $y_{max}$ is the largest $y$-coordinate of the data points in the cell.

2. We build equi-depth histograms based on the $x$-coordinate, where the depth of the histograms can be specified by a parameter $\varphi$. In addition, we keep the largest and smallest $x$ and $y$ values for each bin of the equi-depth $x$-histogram.

3. For the cells on the diagonal line, we further keep their data points, for partial query evaluation.

**Procedure** `top_down`
**Input**: A twig query $p$, a set of query points $Q$, a data graph $G$
**Output**: the count of $p$ in $G$

01 **case of** $p$:
02 (i) $//A/p'$ /* $A$ is a tag */
03 $Q'$ = `estimate_intermediate`($//A$, $Q$) /*Sec. 7.3.1*/
04 **return** `top_down`($p'$, `equiv`($Q'$), $G$)
05 (ii) $//A$
06 **return** `estimate_count`($//A$, $Q$) /* Sec. 7.3.2 */
07 (iii) $//A[q]/p'$
08 $C_q$ = `bottom_up`($q$, $G$, $G$)
09 $Q'$ = `estimate_intermediate`($//A$, $Q$)
10 $Q'$ = $\{g \mid g \in Q' \wedge$ `at_right_bottom`$(g, C_q)\}$
11 **return** `top_down`($p'$, `equiv`($Q'$), $G$)
12 (iv) $//A[q]$
13 $C_q$ = `bottom_up`($q$, $G$, $G$)
14 **return** `estimate_count_with_Qf`($//A$, $Q$, $C_q$)

**Figure 9. The overall estimation algorithm `top_down`**

### 7.2 The Overall Estimation Algorithm

The estimation exploits a property of data points, which can be readily derived from Definition 5.4. A node $v$ is a descendant of another node $u$ if and only if the interval of $v$ is *contained* in that of $u$. This is equivalent to say that $v$ is a descendant of $u$ if and only if the data point of $v$ is at the bottom-right region of the data point of $u$. Fig. 8(a) shows the region containing the descendants of a query point. While the region is divided into five cases as in [24], our detailed esitmation exploits the auxilary structures to handle skewed data points each of the region.

With the above, we are now ready to present the overall estimation algorithm `top_down`, shown in Fig. 9. In a nutshell, `top_down` estimates the path of the twig query top down and invokes `bottom_up` to estimate the filters (branches) in the query. Some important technical details of `top_down` are given in Section 7.3.

The input of `top_down` is a twig query $p$, a set of query points $Q$ and a data graph $G$. Initially, $Q$ contains the root node, at where the evaluation starts. `top_down` proceeds according to the structural form of the query as follows (We omitted $\wedge$ and $\vee$ for presentation simplicity):

(i) If the query is $//A/p'$ (Lines 02-04), where $//A$ is an intermediate query, we compute *the next queries* (*i.e.*, points)

**Procedure** `bottom_up`
**Input**: A filter query $q$, a set of query points $P$, a data graph $G$
**Output**: the points that have some data points that satisfy $q$
01 **case of** $q$:
02 (i) $//A$
03 **return** `estimate_intermediate_reverse`($//A$, $P$, $G$)
04 (ii) $/p'//A$
05 $P'$ = `estimate_intermediate_reverse`($//A$, $P$, $G$)
06 **return** `bottom_up`($p'$, `equiv`($P'$), $G$)
07 (iii) $//A[q']$
08 $P'$ = `bottom_up`($q'$, $P$, $G$)
09 $P = \{p \mid p \in P \wedge \exists\, p' \in P'\ p'$ is a descendant of $p\}$
10 **return** `estimate_intermediate_reverse`($//A$, $P$, $G$)
11 (iv) $/q'//A[q'']$
12 $P'$ = `bottom_up`($q''$, $P$, $G$)
13 $P = \{p \mid p \in P \wedge \exists\, p' \in P'\ p'$ is a descendant of $p\}$
14 $P''$ = `estimate_intermediate_reverse`($//A$, $P$, $G$)
15 **return** `bottom_up`($//q'$, `equiv`($P''$), $G$)

**Figure 10. Auxiliary procedure for handling filters** `bottom_up`

$Q'$ from $Q$ with `estimate_intermediate` (to be detailed in Section 7.3.1). Then, we proceed to estimate $p'$. Since columns may be represented by multiple positions, we need to process all the equivalent query points of $Q'$ determined by `equiv` (to be detailed in Section 7.3.1).

(ii) If the query is the last step (Lines 05-06), we generate *the selectivity count* with `estimate_count` (to be detailed in Section 7.3.2).

(iii) Suppose the query contains a filter $q$ ($//A[q]/p'$) (Lines 07-11). We determine the points $C_q$ whose bottom-right region contains *some* points satisfying $q$ (Line 08). In a nutshell, `bottom_up` is mostly symmetric to `top_down` and returns a set of points that satisfy the filter $q$. (Its details will be presented in the next subsection.) Then, we estimate the next query points $Q'$ (Line 09) as in Case (i) but we only keep the query points that have some points in $C_q$ in their bottom-right region (Line 10). Next, we estimate $p'$ recursively, as in Case (i).

(iv) If the filter occurs in the last step ($//A[q]$), we need to generate the selectivity count, similar to Case (ii). However, we invoke `bottom_up` to find the query points $C_q$, similar to Case (iii). The difference between Line 06 and Line 14 is that when we generate the count, we only include the points that have some points in $C_q$ in their bottom-right region.

**Example 7.1:** A partial run of `top_down` on an example query `//a//b//c` is shown in Fig. 11. The estimation starts with the root node (Fig. 11(a)). The root node becomes a query point that searches for $a$-descendant nodes, with `estimate_intermediate`. The shaded cells illustrate the

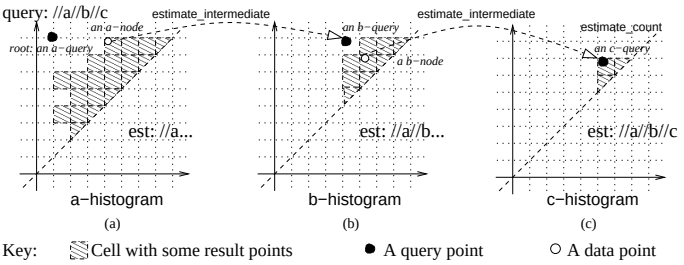



**Figure 11. A schematics of the overall estimation algorithm**

cells that contain some $a$-nodes. The selected $a$-nodes become the next query points — $b$-queries. In Fig. 11(b), we show one $b$-query. Similarly, we determine the cells that contain some $b$-descendant nodes and subsequently $c$-queries. In Fig. 11(c), we show one $c$-query. Since `//c` is the last step, `estimate_count` is invoked to count the number of $c$-descendant nodes in the shaded cells. ∎

#### 7.2.1 Details of `bottom-up`

This subsection provides the details of the handling of filters in the overall estimation algorithm `top_down` (Fig. 9). The filters are handled by Algorithm `bottom_up` shown in Fig. 10. The input of `bottom_up` is a filter $q$ specified in the form of twig query, a set of intermediate query points $P$ and a data graph $G$.

We first discuss `estimate_intermediate_reverse`, which is used in `bottom_up`. We skip its pseudo-code because it is straightforward. `estimate_intermediate_reverse`($q$, $P$, $G$) returns a set of points that satisfy $q$ in a graph $G$ and has a point $p \in P$ in its bottom-right region.

Next, we focus on the structural recursion in `bottom_up`.

(i) If the filter query is $//A$, *i.e.*, the last step (Lines 02-03), we simply return the set of points (of cells) that contain some $A$-nodes which have some $p \in P$ in their bottom-right region.

(ii) If the filter query is not the last step ($//p'//A$), we first determine the points that satisfy $//A$ (Line 05). Then, we recursively determine the points that satisfy $//p'$ based on the result of $//A$ (Line 06).

(iii) If the filter query contains yet another filter query $q'$, we invoke `bottom_up` recursively to first determine the points $P'$ that satisfy $q'$ first (Line 08). We keep only the points in $P$ that have some points in $P'$ in their bottom-right region (Line 09). Then, we determine the points of $//A$ as in Case (i) (Line 10).

(iv) This case (Lines 11-15) is similar to Case (iii). The difference is that after we determine the points $P''$ for $//A[q']$, we use $P''$ to determine the points for $//p'$ recursively.

**Example 7.2:** Recall from Algorithm `top_down` that `bottom_up` is invoked as `bottom_up`($q$, $G$, $G$) (at Lines 08 and 13) and consider an example where $q$ = `//a//b`. Note that `bottom_up` processes $q$ bottom-up. Initially, we encounter Case (ii). We determine the nodes that satisfy `//b` and has a descendant in $G$. Hence, we obtain a set of all $B$-nodes in $G$ as $P'$ (Line 05). Next, we evaluate `//a` with all $B$-nodes (Case (i)). We obtain a set of $A$-nodes that have some $B$-descendant nodes. These nodes will be returned to `top_down` to filter query points. ∎

## 7.3 Estimation Details with Histograms

This subsection provides the technical details of the overall algorithm, specifically, `estimate_intermediate`, `estimate_count` and `equiv`.

### 7.3.1 Generation of Intermediate Queries

Given a query point, `estimate_intermediate` generates a set of next query points from five distinguishable cases in the bottom-right region of the query point. The five cases are visualized in Fig. 8(a).

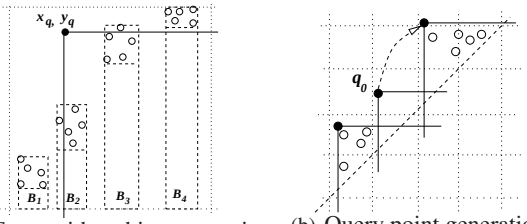
To illustrate how `estimate_intermediate` works, we present the estimation details with an example query `a//b//...`. Suppose the query point in Fig. 8(a) is an $a$-query and the histogram summarizes $b$-nodes. With reference to Fig. 8(a), we present the generation of $b$-queries below:

- *Cases I, II and III.* Suppose an $a$-query point is $(x_q, y_q)$ and $(x_{min}, y_{max})$ represents the bounding rectangle of a cell of these cases. The $b$-query point is (`max`($x_q$, $x_{min}$), `min`($y_q$, $y_{max}$)).
- *Cases IV and V.* The $b$-query point is simply ($x_{min}$, $y_{max}$) of a cell of these cases.

### 7.3.2 Generation of Result Count

`estimate_count` generates a count from the histogram. Similarly, assume that the query dot in Fig. 8(a) is an $a$-query (generated from `estimate_intermediate`) and the histogram summarizes $b$-nodes. We present the generation of the count of $b$-nodes of the query $a//b$ as follows:

- *Cases I.* The count of the result points of a query point ($x_q$, $y_q$) in the cell is estimated to be the the sum of the count of the bins that contain data points with an $x$-coordinate larger than $x_q$ *and* a $y$-coordinate smaller than $y_q$. We illustrate this with Fig. 12(a). The estimated count is 10 (from $B_2$ and $B_3$).



(a) Est. with $x$-histograms in grid (b) Query point generation

**Figure 12. Estimation with $x$-histograms and query point generation**

- *Case II.* The count is estimated to be the sum of the count of the bins with some data point whose $x$-coordinate is larger than $x_q$.
- *Case III.* This is similar to the above case except that we check the $y$-coordinates and $y_q$.
- *Case IV.* We simply return the count of the cell.
- *Case V.I.* If the query point is not in the diagonal cell, we simply return the count of the cell.
- *Case V.II.* If the query point is in the diagonal cell, we check the bins with some data point whose $x$-coordinate is larger than $x_q$ as follows: (i) If the bin's largest $y$ is smaller than $y_q$, we simply include the count of the bin. (ii) If the bin's smallest $y$ is larger than $y_q$, we skip the bin. (iii) Otherwise, we *evaluate* the query with the bin. Evaluation is invoked because as ($x_q$, $y_q$) approaches the diagonal line, there are fewer data points in the bottom-right region, where the error introduced can be relatively large. Suppose Fig. 12(a) is a diagonal cell. The estimated count is 5 (from $B_2$) + 4 (from $B_3$) = 9.

`estimate_count_with_Qf` is similar to `estimate_count` except that it considers a set of points $C_q$, which satisfy a filter $q$. In addition to checking the bins and data points with ($x_q$, $y_q$), `estimate_count_with_Qf` includes the bins and points in estimation only if there are some nodes in $C_q$ in their bottom-right region.

### 7.3.3 Generation of Equivalent Query Points

A query point $(x_q, y_q)$ in general has many equivalent query points in the two dimensional space, since a column may be mapped to multiple positions (at the end of Section 6.1) with `equiv`. We now discuss the details of `equiv`. The equivalent query points are generated in two steps. First, we determine the set of column IDs that involve the query point:

$$C = \{c \mid i \in [x_q, y_q], f^{-1}(i) = c\} \quad (1)$$

Second, we compute the intervals that can be constructed by the column IDs:

$$Q = \{(x', y') \mid \forall j \in [x', y']. \exists j = f(c), c \in C\}. \quad (2)$$

To optimize the generation of query points, *i.e.*, $Q$, we propose to skip generating query points that have empty results. The main idea is illustrated with Fig. 12(b). First we assume that the positions of a column ID are sorted in ascending order offline. We sort $C$ obtained from Equation (1). We scan through the positions of $C$ in parallel. When we obtain a query $q_0$: ($x_0$, $y_0$) that has empty result, we probe the histogram to obtain the grid that contains the data point with the next $x_{min}$, where $\nexists x.\ x_0 < x < x_{min}$ and ($x$, $y$) is a data point. Finally, we skip all positions of columns in $C$ that are smaller than $x_{min}$.

**Compression of mappings between columns and positions.** The mappings $f$ and $f^{-1}$ between column IDs and positions can be potentially large. In query point generation, the mappings are scanned, as just discussed. We compress the mappings such that the scan can be efficiently supported in the compressed domain. In essence, instead of storing the equivalent positions of column IDs, we store the difference (delta) between each pair of adjacent positions. We replace repetitive deltas with an ID and their occurrence. For example, the positions (2,3,4,6,8,10,12,14) are compressed to (2,#1×2 ,#2×5), where #1=1 and #2=2. The positions can be trivially regenerated in a scan through the compressed deltas.

**Offline `equiv` computation.** The next optimization on $f$ and $f^{-1}$ is to precompute `equiv` for all data points and use histograms to summarize the equivalent points, as opposed to computing `equiv` on-the-fly. It is possible because $f$ and $f^{-1}$ depend only on the data graph, not query workloads. In this case, a node is represented by multiple intervals.

## 8 Experimental Evaluation

In this section, we present an extensive experimental evaluation that verifies the accuracy of our proposed technique and the effectiveness of proposed optimizations. We performed an experimental comparison with XSEED [25] on tree data. Since the implementation of XSKETCH/TREESKETCH [16, 17] is not supported by recent operating systems, we perform an indirect comparison with them.

**Experimental settings.** We ran our experiments on a server with a Dual 4-core 2.93GHz CPU and 30GB memory running SOLARIS OS (CENTOS release 5.4). Our implementation was written in `Java JDK` 1.6. We implemented equi-depth histograms for grid cells. The default value of the depth of a bin is 10% of the points in a grid cell. We tested equi-width histograms as well but they exhibited a similar performance to equi-depth histograms in our preliminary experiments.

**Table 1.** XMARK **Characteristics**

| XMARK s.f. | 0.1 | 0.4 | 0.7 | 1.0 | DBLP | TREEBANK |
|---|---|---|---|---|---|---|
| Avg. bindings | 3.1k | 14.1k | 22.9k | 35.4k | 338k | 3k |

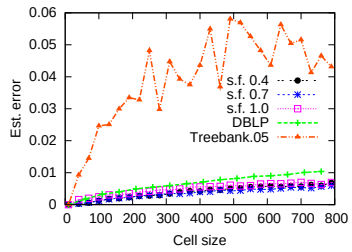
**Benchmark datasets.** We used XMARK [18], DBLP [15] and TREEBANK [11] to obtain a set of large graphs for evaluation. The scaling factor (s.f.) of XMARK was ranged from 0.4 to 1.0. We set the default s.f. value at 1.0. The DBLP used contains 3.3 million nodes. We note that XSEED supports TREEBANK by extracting up to 5-percentile vertices and hence we followed such an extraction. In addition, since XSEED supports trees only, we ignore the IDREFs in XMARK for the experimental comparisons with XSEED

**The definition of metrics.** In our experiments, we used the error metrics used in [16] and [25]. The definitions of these metrics can be described as follows. Let $n$ be the number of positive queries, $a$ be the real result count of a query and $e$ be the estimation value. The estimation error is defined to be $(\sum_{i=1}^{n} \frac{|a_i - e_i|}{e_i})/n$. Similar to [16], we applied a sanity bound $s$ to avoid high percentages of low-count queries. We set $s$ to 10-percentile as in [16]. Two alternative error definitions, root mean square deviation (RMSE) and normalized RMSE (NRMSE), were also adopted [25]. RMSE is defined as $\sqrt{(\sum_{i=1}^{n}(e_i - a_i)^2)/n}$ and NRMSE is defined as RMSE/$\bar{a}$, where $\bar{a}$ is $(\sum_{i=1}^{n} a_i)/n$.

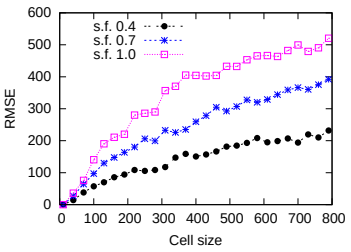
**Query workload.** We implemented a query generator based on the description in Polyzotis *et al.* [17]. However, since our proposed technique does not involve the synopses of XSKETCH/TREESKETCH, our query generator generates twig queries by sampling the data graph, as opposed to the synopses. On the XMARK, DBLP and TREEBANK datasets, we generated 1,000 positive twig queries, where the query results are larger than 0. The twig queries have one branch on average. The length of the main path ranges from 2 to 5. The number of branches ranges from 1 to 3. This workload is similar to the CP workload reported in [25]. Some characteristics of query workloads are shown in Table 1.

### 8.1 Experiments on overall performance

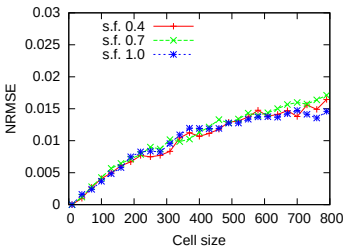
**Scalability tests.** The estimation errors of the queries on various XMARK graphs are shown in Figs. 13(a)-(e). The $x$-axis of Figs. 13(a)-(c) is the cell size. From Fig. 13(a), we note that the estimation error increases as the cell size increases (from 0% to 0.7%), as fewer details are captured by larger cells. Our technique is less accurate in DBLP and TREEBANK but the error is still lower than 1.3% and 6%, respectively. Fig. 13(b) shows that RMSE of our implementation increases with the data graph size. However, the normalized RMSE of our implementation is roughly a constant as the data graph size increases, shown in Fig. 13(c). This



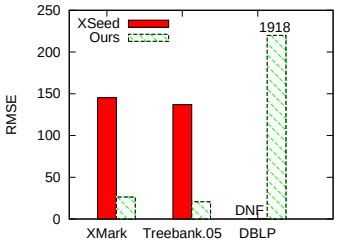

(a) Estimation error of our method



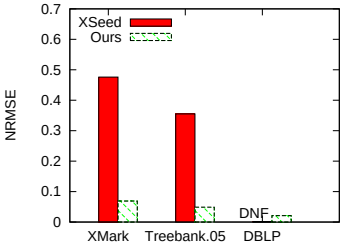

(b) RMSE of our method (XMARK)



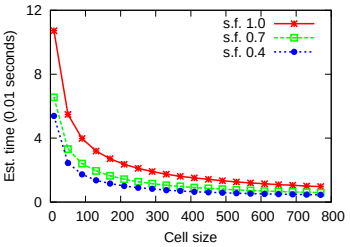

(c) NRMSE of our method (XMARK)



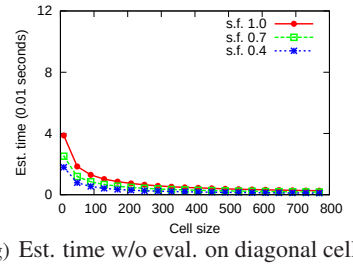

(d) RMSE comparison



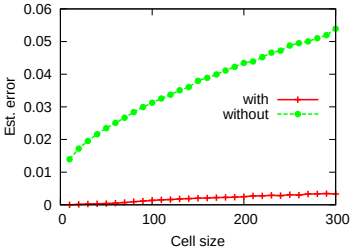

(e) NRMSE comparison



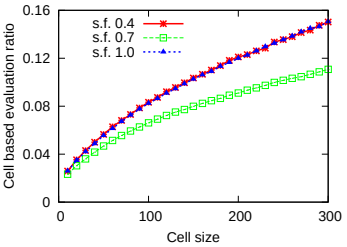

(f) Est. time with eval. on diagonal grid cells



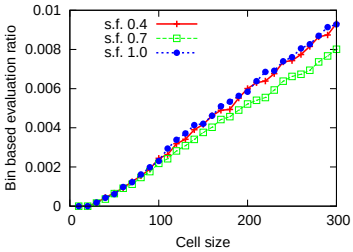

(g) Est. time w/o eval. on diagonal cells



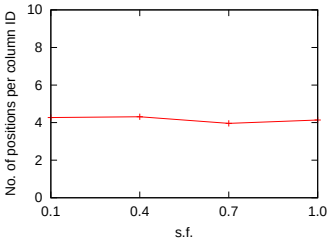

(h) w. vs w/o eval. on diagonal cells



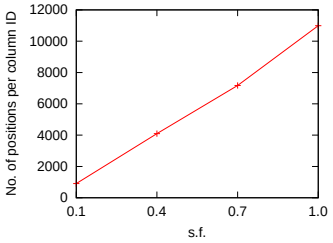

(i) Cell-based eval. to est. ratio



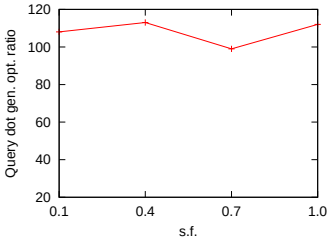

(j) Bin-based eval. to est. ratio



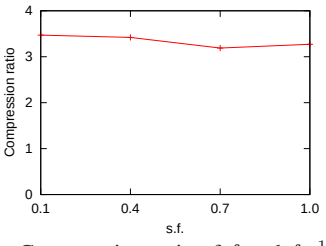

(k) C1P matrix size w. optimization



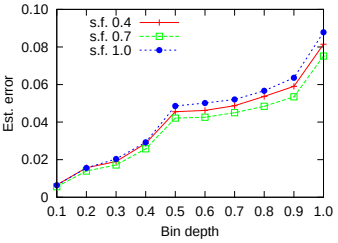

(l) C1P matrix size w/o. optimization



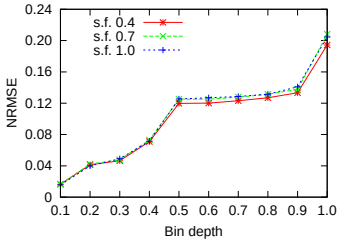

(m) Optimization on query dot generation



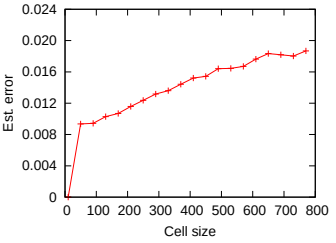

(n) Compression ratio of $f$ and $f^{-1}$



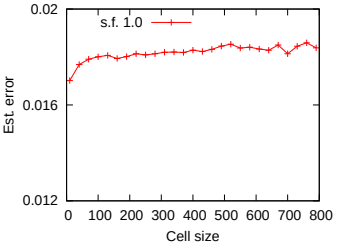

(o) Estimation error as a function of $\varphi$




(p) NRMSE as a function of $\varphi$




(q) Path queries on XMARK graph




(r) Twig queries on XMARK tree

**Figure 13. Performance results on synthetic dataset (XMARK) and real datasets (DBLP and TREEBANK)**

is because the absolute result counts are larger in XMARK datasets with larger s.f.'s.

Next, we set the cell size to 800 and use XMARK s.f. 1.0, unless otherwise specified.

We ran the query workload with our implementation and XSEED's. The results given in the two error metrics are presented in Figs. 13(d)-(e). Regarding XMARK, our method's RMSE and NRMSE are more accurate than XSEED's by a factor of 7.1 and 6.9, respectively. On TREEBANK, our method is roughly 6.8 times more accurate than XSEED, in terms of both RMSE and NRMSE. XSEED's evaluation does not finish on DBLP, which is needed for computing XSEED's errors. Our implementation gives a large RMSE on DBLP since the average number of bindings of DBLP is relatively large (Table 1).

**Experiment on estimation time.** Figs. 13(f)-(g) show the estimation time with and without evaluation on diagonal cells. In any case, the estimation time is less than 0.12 seconds. On average, the estimation with evaluation on diagonal cells is (on average) 2.8 times slower than the estimation without evaluation. However, due to evaluation on diagonal cells, the estimation error reduces from 5% to 0.3% when the cell size is 300 (Fig. 13(h)). In Figs. 13(f) and (g), we did not include the time for `equiv` as it is the same for both methods. For example, the time for `equiv` on XMARK s.f. 1.0 was approximately 0.01 seconds among various cell sizes.

## 8.2 Experiments on optimizations

**Optimizations on diagonal cells.** In Fig. 13(h), the error reduction due to evaluation on diagonal cells is large. In this experiment, we analyze how much evaluation is there in the overall estimation. We computed the ratio between the number of points involved in evaluation and those in estimation. Fig. 13(j) shows the ratio is smaller than 1%. The reasons are that most of the points involved in estimation are not in diagonal cells and the evaluation is not performed on the entire diagonal cell, due to $x$-histograms.

**Experiment on evaluation vs estimation ratio.** To support the argument that the $x$-histograms in diagonal cells effectively reduce the amount of evaluation, we tested the ratio between the number of points involved in evaluation and that in estimation when the $x$-histograms in cells are *not* even used. The result is shown in Fig. 13(i). Without the $x$-histograms, the ratio reaches 16%, when the cell size is 300. In comparison, Fig. 13(j) shows that the ratio does not reach 1% when $x$-histograms in cells are used.

**Matrix transformations.** We tested the size of the C1P matrix obtained from the transformation proposed in Section 6.1 with and without the optimizations in Sections 6.2.1-6.2.2. With the optimizations, Fig. 13(k) shows that the ratio of the increase of the matrix size is approximately 4.1 as s.f. increases. While the size of the matrix increases by a *constant* factor, the C1P matrix is much simpler (recall Figs. 6-7). In contrast, without optimizations, the ratio increases linearly with the s.f. as shown in Fig. 13(l).

**Query point generation.** We determined the ratio between the number of query points in estimation with and without the query point optimization proposed in Section 7.3.3. The result is shown in Fig. 13(m). It shows that we reduce the number of query points generated by a factor over 100. That is, given a query, there are many equivalent query points that do not contribute the result counts.

**Compression of $f$ and $f^{-1}$.** Next, we tested the compression performance presented in Section 7.3.3. The size of $f$ and $f^{-1}$ is important to estimation time as each generation of equivalent query points requires a scan on the compressed $f$ and $f^{-1}$. The result is plotted in Fig. 13(n). The figure shows that the compression ratio is roughly 3.1 for various s.f.'s.

**The depth of the bin in $x$-histograms.** In previous experiment, we set the depth to be 10% of the number of dots of a cell. To show the effect of the depth of $x$-histograms on the estimation accuracy, we performed an experiment by varying the bin's depth from 10% to 100%. The cell size is 800. The results are shown in Figs. 13(o)-(p). As expected, when the depth increases, the estimation error and NRMSE gradually increase. Due to space constraints, we skip the results on RMSE as we observed a similar trend.

## 8.3 Indirect comparison with XSketch and TreeSketch

Although the implementation of XSKETCH/TREESKETCH [16, 17] has been available, it was developed on a legacy `gcc`, which is no longer supported. Some `gcc` libraries used have no longer been available. Therefore, we could only compare the numbers reported from [16, 17]. We compared the estimation error of XSKETCH/TREESKETCH and ours on XMARK dataset s.f. 1.0.

As discussed, XSKETCH supports path queries only for cyclic graphs. XSKETCH generates queries based on the popularity of tags in their synopses, which is absent in our method. Hence, we generated path queries based on the popularity of tags in data graphs. As shown in Fig. 13(q), our estimation error has not reached 2% when the cell size is smaller than 800. When the cell size is smaller than 200, our estimation error has been controlled under 1%. In comparison, the estimation error of XSKETCH reported in [16] is well-controlled under 10%.

Next, we compared the results reported from TREESKETCH [17]. TREESKETCH estimates the selectivity of twig queries but on acyclic graphs only. As in

TREESKETCH, we did not consider IDREF in XMARK. The twig queries were generated as described in the beginning of this section. Fig. 13(r) shows that our technique controls the estimation error around 1.6%. In comparison, TREESKETCH controls the estimation errors of twig queries on XMARK tree under 5%.

## 9 Conclusion

In this paper, we propose a histogram-based selectivity estimation of twig queries on cyclic graphs. To the best of our knowledge, previous works only focus on either twig queries or cyclic graphs but not both. Specifically, we propose a new matrix representation of cyclic graphs by our prime labeling scheme. Next, we derive a heuristic transformation of the matrix to a C1P matrix for summarization. As a result, a data node is represented by an interval and subsequently a two-dimensional data point. A query is then represented by multiple points in runtime. Two-dimensional histograms are used to summarize data points and auxiliary structures are introduced to tackle skewed data points. We present a selectivity estimation algorithm on the histograms. Our experiments with XMARK and DBLP show that the estimation error is well-controlled under 1.3%, which is more accurate than XSKETCH/TREESKETCH and XSEED. On TREEBANK, we produce RMSE and NRMSE 6.8 times smaller than XSEED's.

As for future works, (i) we are incorporating this technique with queries with filters on data values; and (ii) we are investigating graph partitioning to optimize the computation of the binary matrix, which is currently maintained in the main memory.

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

[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] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick. Reachability and distance queries via 2-hop labels. *SIAM J. Comput.*, 32(5):1338–1355, 2003.

[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] W.-L. Hsu. A simple test for the consecutive ones property. *J. Algorithms*, 43(1):1–16, 2002.

[10] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. Exploiting local similarity for indexing paths in graph-structured data. In *ICDE*, page 129, 2002.

[11] Language and Information in Computation at Penn. Penn treebank project. Available at http://www.cis.upenn.edu/ treebank/.

[12] 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.

[13] Z. Lin, B. He, and B. Choi. A quantitative summary of xml structures. In *ER*, pages 228–240, 2006.

[14] J. McHugh and J. Widom. Query optimization for xml. In *VLDB*, pages 315–326, 1999.

[15] G. Miklau. UW XML repository. Available at http://www.cs.washington.edu/research/xmldatasets/.

[16] N. Polyzotis and M. Garofalakis. Xsketch synopses for xml data graphs. *ACM Trans. Database Syst.*, 31(3):1014–1063, 2006.

[17] N. Polyzotis, M. Garofalakis, and Y. Ioannidis. Approximate xml query answers. In *SIGMOD*, pages 263–274, 2004.

[18] 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.

[19] J. Tan and L. Zhang. The consecutive ones submatrix problem for sparse matrices. *Algorithmica*, 48(3):287–299, 2007.

[20] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[21] 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.

[22] 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.

[23] X. Wu, M. L. Lee, and W. Hsu. A prime number labeling scheme for dynamic ordered xml trees. In *ICDE*, page 66, 2004.

[24] 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.

[25] N. Zhang, M. Ozsu, A. Aboulnaga, and I. Ilyas. Xseed: Accurate and fast cardinality estimation for xpath queries. In *ICDE*, pages 168–197, 2006.

# A Tag Means a Lot Than It is in a Folksonomic System

Ho Keung Tsoi
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
hktsoi@comp.hkbu.edu.hk

## Abstract

*Folksonomic system allows users to use tags to describe items, these tags do not just exist in the form of textual description, they actually bear more meaning underneath, such as user preference. In this paper, we first show the distribution of preferences and semantic categories across a folksonomic system, and then develop a hybrid design to cope with the cold-start problem.*

*We speculate the semantic categories formed in user perspective and item perspective in a folksonomic system are different. They represent different preferences and meaning and are believed to be crucial in recommender algorithm design. Through a dimensionality reduction technique, the Latent Dirichlet Allocation, we demonstrate our speculation is correct.*

*In this regards, we design a hybrid strategy accordingly. Our strategy consists of two stages. First we enhance the user's tag profile by WordNet, so as to provide more sound information for later use. The second stage is to find a winning cluster, that maximizes the user's preference. The evaluation reveals our design outperform other existing approaches. This verifies our idea of leveraging various users' interests in the recommendation process, is capable of yielding a better result. Since this strategy can stimulate user's preference, it can enhance user experience as well as solving the cold-start problem.*

## 1 Introduction

How do you judge the person at first sight? By his appearance or by his dressing? You do not have too much information about this individual in the first meet, and hence you cannot do too much. This is also the case in providing recommendation to novel users, the lack of prior information hinder us from understanding the new comers.

Folksomonic system [23] has become popular and growing rapidly in recent years. A folksomonic system allows Internet users to assign keywords – so called tags, to annotate resources. The role of these tags is to help users to manage, navigate and explore resources. Living examples of this system include Flickr[1], Last.fm[2] or Delicious[3].

Different analysis of tagging pattern and motivation in folksonomic systems have been done by peer researches. They show the types of tags used in the social tagging process can be classified in the categories of *Personal*, *Factual* and *Subjective*[3], and a semantic space of social tags will gradually be evolved from the folksonomic system, this semantic classification of tags formed by social tagging has some self-organizing characteristic[20]. As for the motivation of applying tags, study has shown that it is driven by the purpose of sharing and personal information management[15].

To this end, various tag recommendation algorithms have been deviated from these folksonomic systems. Tag recommendation can facilitate users to browse and search resources, as well as to manage and retrieve their own resources. Contemporary tag recommendation algorithms include cluster-based[19], memory-based[1], content-based[29] or collaborative filtering[27] approach. These approaches rely heavily on available prior information, such as rating, to find a matching relevant candidate to return to user. When a novel user is encountered that prior information is yet available, the recommender system struggles to generate a recommendation. This is know as the "cold-start" problem [18].

However, stimulating user's preference should be the primary goal of recommendation. If a user doesn't interested in a system, he will not use the system anymore, not to mention to accomplish the above tasks. But the aforementioned algorithms either suffering from the cold-start problem, or overlook the essence of user's preference, we hope to seek a solution that balances the two sides. In this paper, we design a cluster-based algorithm to give solution to the cold-start

[1] http://www.flickr.com
[2] http://www.lastfm.com
[3] http://delicious.com

problem, while reserving user preference in the process.

The remainder of this paper is organized as follows. Section 2 is the literature review. In Section 3 we show the dataset we use throughout this paper. In Section 4, we present an analysis, and state the difference of semantic categories built in user- and item-space. In Section 5, a brief introduction of algorithms to be experimented and our strategy is presented. A comparison and evaluation of these algorithms are examined in Section 6. In Section 7 we summarize and discuss the algorithm and future works.

## 2 LITERATURE REVIEW

A folksonomy can be described as a four-tuple: a set of users, $U$; a set of resources, $R$; a set of tags, $T$; and a set of assignments, A. The data in the folksonomy is denoted as $D$ and is defined as: $D = \langle U, R, T, A \rangle$. The assignments, $A$, are represented as a set of triples containing a user, tag and resource defined as: $A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\}$. Therefore a folksonomy can be regarded as a tripartite hyper-graph with users, tags, and resources represented as nodes and the assignments represented as hyper-edges connecting one user, one tag and one resource[24].

As such, Graph Theory has always been adopted to provide recommendation in folksonomy. A graph-based ranking algorithm for interrelated multi-type object is proposed[14]. The task of Personalized Tag Recommendation is modeled as a "query and ranking" problem. When a user issues a tagging request, both the document and the user are treated as a part of the query. This algorithm ranks tags by considering both relevance to the document and preference of the user. Likewise, some authors are inspired by the algorithm *PageRank*, and use *authoritative* tags to enrich user query[9]. Each folksonomic user is maintained a profile in their approach, as well as a knowledge base consisting of two graphs called *Tag Resource Graph* and *Tag User Graph*. These graphs register the tags exploited in the folksonomy and the way they label involved resources, or the way they are registered in the user profiles. When user submit a query, *authoritative* tags are suggested to user and enrich user profile automatically. FolkRank [17], a enhancement of PageRank-like algorithm that takes into account the structure of folksonomies to search in the system.

The idea of embedding content information in the recommendation process is not novel. The authors in [29] describe a movie recommendation system built purely on the keywords assigned to movies via collaborative tagging. Recommendations for the active user are produced by algorithms based on the similarity between the keywords of a movie and those of the tag-clouds of movies the user rates. According to [8], content-based recommender not only can recommend items, but also be used to infer user interests. They use a multivariate Possion model for naive Bayes text classification adapted to infer user profiles from both static content, as in classical content-based recommender, and tags provided by users to freely annotate items. The benefit of using content information includes solving the cold-start problem. Researches in [12] propose a probabilistic model for inferring the mist probable tags from the text of the book. They combine a Relevance model developed for Information Retrieval, and the Collaborative Filtering approach, to generate tags from the content of books.

Besides content information, contextual information such as navigational pattern, and browsing behaviors also play a major role in recommendation process. [32] conduct a study to determine which context information sources can predict user's interests effectively. In particular, they evaluate *social*, *historic*, *task*, *collection*, and *user interaction*. Their result shows the context overlaps outperform any isolated source, and suggests designers can improve Website suggestion by these findings. [21] suggests items to users based on inferences made about user interests gleaned from their task environment, such as recently-viewed Web pages or the contents of active desktop applications. Underneath the obvious contextual information like the link structure of the Web, implicit connection in a folksonomic system can link related users together. Authors in [5] formulate user-induced links in collaborative tagging system as follows: if two documents that are maintained in the collection of the same user and/or assigned similar sets of tags can be considered as related from the perspective of the user. They then demonstrate that this kind of induced-link achieves much higher accuracy than existing hyperlinks. Contextual information also includes emotional context and the like. [13] present a SPA system, which elicits user's preference through a rich interaction through highly dynamic environments, networked game for example. This approach is particularly useful in social software systems, as it can easily acquire information via user activities and tasks. As shown, the task of generating recommendation is not limited to the scope of floksonomy.

Due to the textual nature of tags, each tag bears a semantic meaning. This leads to researches focusing on the semantic dimension to produce recommendation. WordNet dictionary [30] and ontologies from open linked data published on the Web [6] are the tools to support this task. To recommend items which are about similar contents, [10] find semantic relations between tags on different semantic sources and calculate their semantic similarity. In addition to applying semantic calculus directly on the tags, clustering tags at a higher level can also be done. [20] use self-organizing map to determine the tags' semantic dimension in social tagging system. This higher representation of tags gives more representative, and can amend the weaknesses of traditional methods such as experts or statistics.

Also because of this very nature of tags, the issue of re-

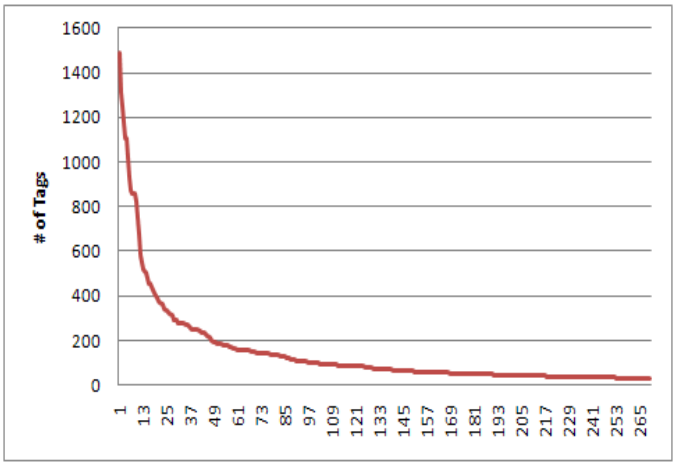



**Figure 1. Tag assignments distribution of each user**

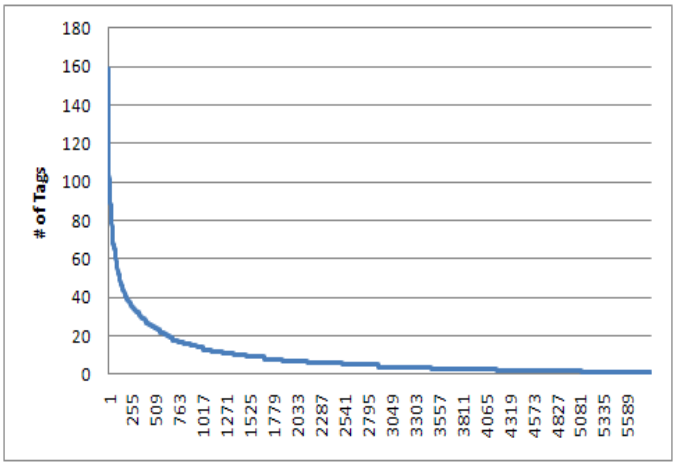



**Figure 2. Tag assignments distribution of each movie**

dundancy and ambiguity has to be solved. The free annotation in a folksonomic system permits unsupervised tagging, and users can use tag in a way they desire. Consequently, a single tag has many different meanings or results in ambiguity and redundancy in which several tags have the same meaning. However, since traditional evaluation metrics such as Precision and Recall cannot account the effect of these phenomena, [11] use a cluster-based approach to define ambiguity and redundancy and provide evaluation on real world datasets. They show this evaluation strategy can more reveal the utility of tag recommender.

## 3 DATASET

The work of this paper is based on the MovieLens[4] dataset, and the movie's information obtained from IMDB[5]. MovieLens is the movie recommender system maintained by GroupLens Research. Each movie in the dataset has a link led to that movie's description on IMDB. Since we are interested in user tag profile, we trimmed the data as follows to counteract the effect of skewed distribution[28]. Figure 1 and Figure 2 show the distribution of our dataset. They reflect the long-tail phenomena in a folksonomic system – a majority of users/items use very few tags.

In the original dataset, we extracted a set of users who applied at least 30 tags (include duplicated tags). The set of tags belong to these users and the set of movies related to these users are considered. As a result, we have the following data.

For each of the related movie, we additionally crawled the movie's description from the associated IMDB link[6]. Keywords are then extracted from the textual description by comparing against the standard stop-word list.

**Table 1. Summary of the data set used**

| | MovieLens |
|---|---|
| # users | 271 |
| # distinct tags | 6,409 |
| # movies | 5,840 |

## 4 THE DIVERGENCE OF SEMANTIC CATEGORIES

There are two approaches to discover semantic categories given a folksonomic system. One is to treat each user as a document, and the list of tags *assigned by* this user as words; alternatively, the list of tags *assigned to* an item can be regarded as a document, and thus the tags are words to describe this item. At this point, we have user-space and item-space. We further define that the semantic categories in user-space reveal the general preferences of users, while those in item-space reveal the objective description of items.

We come to these definitions due to the following observations. In user-space, a user assigns tag based on individual preference. A user might repeatedly use the same tag on different items, for instance. So the bag of tags used by this user can indicate his general preference. Whereas in the item-space, the item's tags are given by more than one user, and therefore those most frequently occurring tags can lighten the effect of individual bias, and hence objective. A common way to examine the semantic categories

[4] http://movielens.umn.edu/
[5] http://www.imdb.com/
[6] Example link: http://akas.imdb.com/title/tt0114709/synopsis

is through the technique of dimensionality reduction. Algorithms like Latent Semantic Indexing[31], Latent Dirichlet Allocation[7] and Self-Organizing Map[26] are capable of achieving this purpose. In particular, we rely on Latent Dirichlet Allocation (hereafter, LDA) for our subsequent analysis and algorithm development.

### 4.1 The Analysis

We use LDA to examine the divergence of semantic categories among user-space and item-space. For illustrative purpose, we specify the number of laten topics to be 30. The result can be found in Table 2.

For each topic, we use the top three representative tags to represent the topic, the representativeness of a tag is in turn measured by the *term frequency*[10] w.r.t. the topic. It is seen that the semantic categories found in the two spaces not always agree with each other. For example, the cluster {action, Comedy, Drama} in user-space reflects the preference of watching comedic drama movie; the {Betamax, James Bond, 007} in item-space indicates the 007's series movies. This implies dissimilarity in the two spaces. The result is expectable in the sense that the semantic categories formed in user-space reflect the general preferences of users, and that in the item-space tells the factual dimension of resources. In the subsequent section, we will show how our algorithm addresses these findings.

## 5 TAG RECOMMENDATION

Based on the findings from previous analysis, we design a strategy to account for them. In this section, we present our approach, and show how we will deal with the issues. Then some of the example algorithms are selected and briefly described, so as to provide readers a brief understanding of the contemporary development of tag recommendations.

### 5.1 Our Approach

Our goal is to provide recommendation to a novel user in a fashion such that maximizes the user's preferences. We reference to the design of existing tag recommendation algorithms, and try to combine their advantages into one. Our idea is to utilize user's tag profile to generate recommendation. By tag profile we mean the tags a user applied. A new user has a tag profile of length one as he first uses the system; this user has a tag profile of length two as he applies the second tag, so on and so forth for instance.

To begin with, we use LDA to discover the semantic categories in both user-space and item-space. Here the item-space contains not only the tags assigned by users, but also the keywords extracted from the movie's description link. The keywords extraction is done by removing words from the standard stop-word list, and the remaining words become the keywords of that movie. Then the similarity of each cluster in user-space to each cluster in item-space are determined using *symmetric Jacaard coefficient* :

$$sim(C_{i^{th}}^{user}, C_{j^{th}}^{item}) = \frac{\mid T_i \cap T_j \mid}{\mid T_i \cup T_j \mid}$$

$$T_i \in C_{i^{th}}^{user}$$

$$T_j \in C_{j^{th}}^{item}$$

where $T$ is the tagset, $C_{i^{th}}^{user}$ denotes the $i^{th}$ cluster in user-space, and $C_{j^{th}}^{item}$ denotes the $j^{th}$ cluster in item-space. This equivalents to bridge the gap between user-space and item-space. Whereas the traditional LDA concerns only either one, and neglect the other, which lead to the resulting clusters do not cover all the possible semantic categories.

Now, lets get back to the user's tag profile. For a new user, immediately after he has entered the first tag, we enrich his profile to five tags using WordNet [25]. We return $N$ recommended tags which have the highest similarity value to the querying tag using Wu and Palmer metric[33], where $N$ depends on the length of the current tag profile. Formally, we enrich the user's tag profile in the initial stage in the following manner:

$$N = \begin{cases} (5 - \mid Profile \mid), & \text{if } \mid Profile \mid < 5 \\ 0, & \text{if } \mid Profile \mid \geq 5 \end{cases}$$

After the initial enrichment, we have a tag profile of length at least five. The tag profile is then used for finding the most relevant cluster in the user-space, and the relevance is defined as the value of overlap between the tag profile and cluster. With the $C_{i^{th}}^{user}$ located, we return both $C_{i^{th}}^{user}$ and the most similar cluster in item-space $C_{j^{th}}^{item}$ found in previous step.

The last step is to decide which cluster to be the final recommendation. Again, we use the maximum overlap for our measurement.

$$WinningCluster = \arg\max(C_{i^{th}}^{user} \cap Profile, C_{j^{th}}^{item} \cap Profile)$$

The rationale behind our strategy is to maintain the tag profile to have certain length, so that we can make use of this piece of information in the recommendation process. Using WordNet, we can enrich the profile with semantically correlated tags. The clusters in the user-space and item-space, on the other hand, represent users' general interests and objective factual information respectively. Choosing among these two groups in the last step captured the importance of user preference, as it has the largest degree of

**Table 2. The Latent Topics found in User-Space and Item-Space**

| User-Space | Item-Space |
|---|---|
| action,Comedy,Drama | serial killer,martial arts,beautiful |
| boring,PG13,afternoon section | Owned,Crime,dvd |
| Can't remember,Friday night movie,Didn't finish | Nudity (Full Frontal - Notable),lesbian,Musical |
| dvd,DIVX,Want | Tumey's DVDs,imdb top 250,black and white |
| AFI 100,Disney,AFI 100 (Laughs) | library,gay,erlend's DVDs |
| girlie movie,Hitchcock | Bruce Willis,psychology,ghosts |
| seen more than once,overrated,James Bond | Oscar (Best Picture),documentary,Oscar (Best Cinematography) |
| Tumey's DVDs,USA film registry,Tumey's To See Again | based on a book,adapted from:book,Fantasy |
| less than 300 ratings,avi,violent | 70mm,World War II,history |
| movie to see,National Film Registry,ClearPlay | comic book,holocaust,super-hero |
| classic,Criterion,history | classic,imdb top 250,National Film Registry |
| Nudity (Topless),Nudity (Topless - Brief),Nudity (Full Frontal - Notable) | anime,In Netflix queue,Japan |
| erlend's DVDs,Sven's to see list,based on book | Betamax,James Bond,007 |
| Bibliothek,seen at the cinema,watched 2006 | less than 300 ratings,To See,Sven's to see list |
| aliens,drugs,remake | boring,Johnny Depp,Adventure |
| 70mm,Betamax,DVD-Video | action,aliens,Eric's Dvds |
| corvallis library,hw drama | dvd-r,library vhs,Scary Movies To See on Halloween |
| imdb top 250,netflix,oppl | dvd,sci-fi,Futuristmovies.com |
| on computer,funny,ohsoso | comedy,funny,chick flick |
| anime,need to own,breakthroughs | drama,biography,christmas |
| atmospheric,Golden Palm | erlend's DVDs,atmospheric,Criterion |
| Futuristmovies.com,documentary,space | directorial debut,time travel,seen more than once |
| owned,adapted from:book,based on a TV show | zombies,movie to see,Angelina Jolie |
| Johnny Depp,Brad Pitt,Arnold Schwarzenegger | movie to see,Below R,parody |
| In Netflix queue,Disney,Christmas | Disney,Animation,Pixar |
| ummarti2006,2.5,cars | remake,Brad Pitt,Based on a TV show |
| Oscar (Best Picture),psychology,toplist08 | Nudity (Topless - Notable),VHS,Jackie Chan |
| based on a book,directorial debut,black and white | Can't remember,own,based on a play |
| World War II,Jack Nicholson,Clint Eastwood | ClearPlay,drugs,PG13 |
| own,Eric's Dvds,Ei muista | Nudity (Topless),Nudity (Topless - Brief),netflix |

agreement to the user's tag profile. The visual presentation of our idea is presented in Figure 3. The following subsections introduce the traditional approaches in tag recommendation.

### 5.2 Collaborative Filtering

In Collaborative Filtering[1, 27], an object is suggested to a user *u* if it was rated as relevant by a group of users having a profile similar to the one of *u*. The profile can be established by user's rating, and the relevance is measured by similarity metric such as cosine-similarity. Formally, similarity between users *i* and *j*, denoted by *sim(i,j)* is given by

$$sim(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\parallel \vec{i} \parallel_2 * \parallel \vec{j} \parallel_2}$$

where · denotes the dot-product of the two vectors.
Variation of such approach includes Pearson Correlation[22], similarity-weighted average of the rating[4] are used for distance metric. This algorithm is intuitive and efficient, but sophisticated readers should spotted the problem of cold-start. Because of the nature of this algorithm relies heavily on user profile, it hardly provides recommendation to novel user or user who doesn't rate.

### 5.3 Association Rule Mining

Association rule mining finds interesting associations and correlation relationships among large set of data items. It has a form $T_1 \longrightarrow T_2$, where $T_1$ and $T_2$ are items (Tags in our case), this indicates $T_1$ implies $T_2$. Association rules show attribute value conditions that occur frequently together in a given dataset. A typical and widely-used example of association rule mining is Market Basket Analysis[2]. The three key measures for association rules are support, confident and interest. The support is simply the number of transactions that include all items in the antecedent and consequent parts of the rule. i.e., an estimate of the joint probability $P(item_1, item_2)$. Confidence is an estimate of the conditional probability $P(item_1 — item_2)$. Interest (a.k.a. Lift) is the ratio of Confidence to Expected Confidence ($\frac{P(item_1,item_2)}{P(item_1)P(item_2)}$).

In the context of tag recommendation, if many resources with tags $Tag_1$ are typically also annotated with tags $Tag_2$, then a new resource with tags $Tag_1$ may also be meaningfully annotated with tags $Tag_2$[16]. But the nature of skewed distribution [28] of tags in a folksonmic system, prohibited the association rule to yield a better performance. If one prefers to maintain a large coverage of tags, the Confidence (so as the Support) parameters have to be lowered.

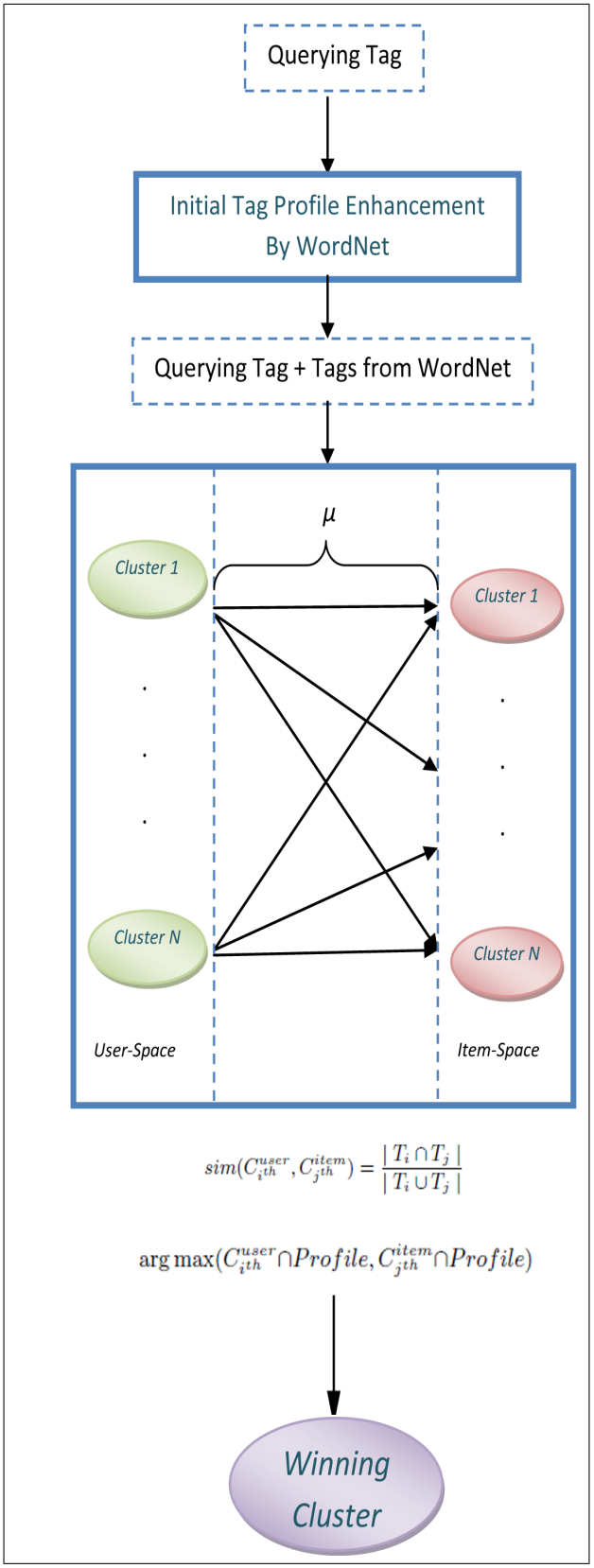



**Figure 3. Graphical representation of our strategy**

### 5.4 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus and it assumes there are $k$ underlying latent topics. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a multinomial distribution over words. Using the terminology in our context, multiple users are annotating resources, and the resulting topics reflect a collaborative shared view of the resource and the tags of the topics reflect a common vocabulary to describe the resource.

The generative process of LDA can be formalized as follows:

1. Choose $N \sim$ Possion($\xi$)
2. Choose $\theta \sim$ Dir($\alpha$)
3. For each of the $N$ words $w_n$:
   (a) Choose a topic $z_n \sim$ Multinomial($\theta$)
   (b) Choose a word $w_n$ from $p(w_n|z_n,\beta)$,a multinomial probability conditioned on the topic $z_n$.

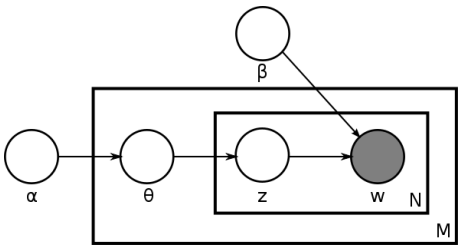



**Figure 4. Graphical model representation of LDA, adopted from [7]**

The parameter $\theta$ indicates the mixing proportion of different topics in a particular resource. $\alpha$ is the parameter of a Dirichlet distribution that controls how the mixing proportions $\theta$ vary among different resources. $\beta$ is the parameter of a set of multinomial distributions, each of them indicates the distribution of tags within a particular topic. Learning a LDA model from a collection of resources $D$ = $\{t_1, t_2, ..., t_3\}$involves finding $\alpha$ and $\beta$ that maximize the log likelihood of the data $l(\alpha,\beta)$= $\sum_{d=1}^{M} \log P(w_d|\alpha,\beta)$. This parameter estimation problem can be solved by the variational EM algorithm.[7]

### 5.5 WordNet

WordNet is an online lexical database designed for use under program control. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each representing a lexicalized concept [25].

To utilize WordNet as a support tool for the tag recommendation[10], each tag is associated with a semantic knowledge and the recommendation is produced by returning tags with the highest similarity value to the querying tag, where the similarity between tags can be measured by the semantic similarity using the formula proposed by Wu and Palmer [33]. The advantage of this approach is that the recommended tags are lexically correlated to the querying tag, but its downside is that it takes no user preference into account.

## 6 COMPARISON AND EVALUATION

In particular, we evaluate Association Rule Mining(*UXASSO*, *IXASSO*)[16], Latent Dirichlet Allocation (*UXLDA*, *IXLDA*)[7, 19], WordNet(*WN*)[33], and our hybrid one. The results are summarized in Figure 5 to Figure 6. The standard metrics in Information Retrieval, i.e. Precision, Recall are adopted. The horizontal axis of the graphs depict the number of tags in user profile. For each run, 200 iterations with the same length of tag profile are performed, and the averaged values are presented in the graphs.

To demonstrate the effect on different semantic categories found in different spaces, we evaluate the same algorithm with two perspectives, namely user-space (hereafter, UX) and item-space (hereafter, IX). And because we are concerning about the cold-start problem, each user we withhold all but one of his tags, and withhold one tag less for another round until his tag profile length reaches five. This setup can help us to simulate the cold-start problem and to determine how the number of tags in the user profile influent the recommendation process.

Besides using the dataset as described in Section 3, we use the following parameter(s) in the experiment. In item-space association rule, the Support and Confidence are set to 10% and 80% respectively; whereas in user-space association rule, these values are 20% and 80%. Since we would like to obtain a considerable amount of rules, and hence larger coverage, we have to lower the Support values to achieve so in a skewed[28] dataset. The number of topics of LDA in both ix and ux are set to 500. As for the measurement of semantic distance in WordNet, we adopt the Wu and Palmer distance metric [33]. Table 3 summarized these settings.

From the graphs, it can be seen that the experimented algorithms can be classified into two distinct classes. One is sensitive to the current user tag profile, another one is not. We first go through the sensitive one, followed by the opposite group.

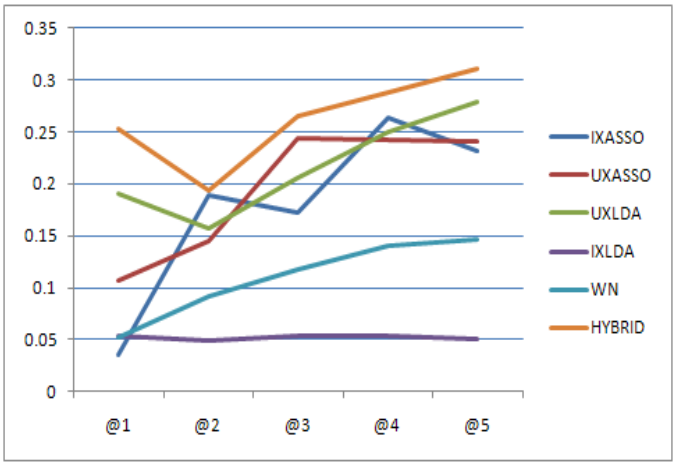



**Figure 5. Average precision of different algorithms with variable profile length**

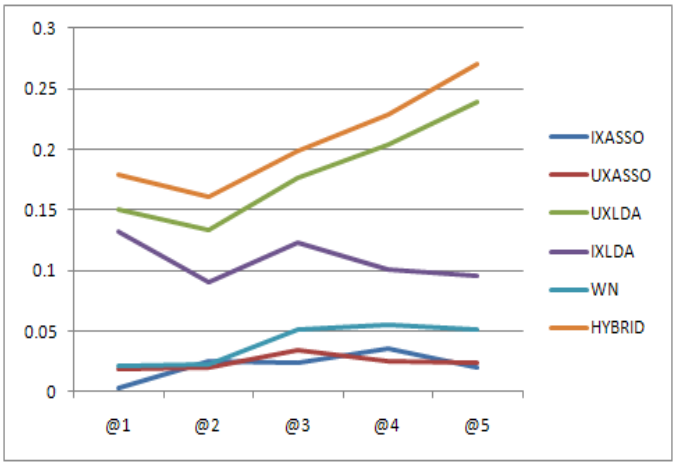



**Figure 6. Average recall of different algorithms with variable profile length**

| | Support | Confidence | No. of Topics | Similarity Metric |
|---|---|---|---|---|
| ixasso | 10% | 80% | / | / |
| uxasso | 20% | 80% | / | / |
| uxlda | / | / | 500 | / |
| ixlda | / | / | 500 | / |
| wn | / | / | / | Wu [33] |
| hybrid | / | / | 500 | Wu[33] |

**Table 3. Parameters of different algorithms**

### 6.1 Sensitive to Tag Profile

All of the experimented algorithms are sensitive to the user's tag profile, except the item-based Latent Dirichlet Allocation. These algorithms have a general trend in common, the averaged precision values increase as the the number of available tags in user's tag profile increase. That is, if we know more about user's interest, we can provide better recommendations.

In this paragraph, we would like to draw your attention to both *UXLDA* and our *HYBRID* approach. Interestingly, when you look at Figure 5 to Figure 6, you might find there are two lines running almost parallel to each other, and the one representing our strategy shifted upward. This indicates our design surpasses *UXLDA*, while the latter one in turn topped the rest of the experimented strategies. The advantages of *UXLDA* is that it emphasizes on the user's preference. Users with variety of preferences able to find a cluster fitting their taste. As in the case in user-based Association Rule, recommending tags that stimulate user's preference yield a better performance. Though *UXLDA* and *UXASSO* are doing similar tasks, *UXLDA* can get rid of the problem of low coverage and hence excel *UXASSO*.

Bear in mind that some individuals would prefer objective tags and others have their specific preferences, our *HYBRID* design further improves *UXLDA* by considering both factors in once. And the initial tag profile enhancement stage of our design plays an important role. It magnifies the user's preference, and is crucial to our subsequent decision of what to deliver to user. The algorithm comes to a *dilemma point* when there are exactly two tags in the user's tag profile, because at this stage, the two tags have equal weight, if these two tags have contradictory meaning, we cannot tell with confident that this user prefers either sides, and hence the performance dropped slightly. This is also the case for *UXLDA*. But this issue is rectified as the tag profile grows.

In the WordNet approach, the algorithm suggests *Top k* recommended tags to the user. The ranking is done by finding the most similar tags to the tags available in tag profile using Wu and Palmer[33] metric. We set the $k$ to be five in our case. Assuming a user has a consistent preference, he

will use more or less the same set of tags to annotate objects. For example, if a user is optimistic, then it is likely for him to use tags such as 'great', 'funny', or 'happy' to describe an object. This explains the increasing precision values, because WordNet provides tags which are lexically correlated to the querying tags (user's preference).

Let's then take a look at the Association Rule. This algorithm generally outperforms WordNet, regardless user-space or item-space. In the beginning stage, when there is only one tag available, *UXASSO* performs better than *IXASSO*. This phenomenon is reasonable in the sense that the rules discovered in *UXASSO* reflect the user's preference. By suggesting tags with high Confidence, the probability of touching user's interest is relatively high. The low precision in *IXASSO* can be attributed to the fact that the rules generated from item-based transactions are generally objective, seldom biased towards individual's preference, which is in line with our assumption.

The distinction of these two approaches become blurred as more and more tags are available in the user's tag profile. This is especially the case when the number of tags reached five. We suppose this is caused by the low coverage in Association Rule. Because the distribution of tags is sparse, it is not easy to construct a rule given certain values of Confidence and Support. Only a small portion of tags, which are frequently co-used by the same user or co-exist in the same item, formed the basis of the rules. This leads to the result of low coverage no matter it is user-based or item-based. Consequently, the algorithm reaches a saturation point once the length of tag profile approach five, in this case for instance.

### 6.2 Insensitive to Tag Profile

The only experimented algorithm falls in this class, is the item-based Latent Dirichlet Allocation. As shown in Figure 5, the averaged precision values of this algorithm remain steady as the number of tags in user tag profile increases. As its name implies, the semantic categories or clusters formed in *IXLDA* reveal only the semantic categories of items, which is objective and descriptive in nature. It is good for discovering the semantic dimensions among items, but insufficient to stimulate user's interest.

As observed from the graphs, we interpret that there is only a few portion of overlapping between the user's preference (i.e. the tags in the user profile) and the item's semantic categories. The more availability of tags in the user's tag profile, the more obvious is the user's preference. However, there is no positive correlation between the length of tag profile and the averaged precision values. The precision values do not grow as the user's tag profile grows. We draw the conclusion that the objective information of items, or the factual categories of tags, merely occupying a small proportion of the whole set of user's preference, and this information is not enough to generate recommendations that fit all kinds of user preferences.

## 7 CONCLUSIONS

We demonstrated that taking the user's preference into account can improve the recommendation results. In a typical recommendation algorithm design, designers always focus on either user-based or item-based information, and overlook the difference between them. As we shown in our analysis, difference does exist.

We recognize the pattern found in user-based tag transactions as their preferences indicator. When a folksonomic system is mature, users in this system composite different niches, each representing certain preferences. An individual can find a niche to suit his interest. In contrast, the tag transactions in item-based are contributed by various users from different niches and thus preferences, resulting in avoiding a particular preference from dominating over others. We interpret these outcomes are objective, and unbiased. This follows the idea of classifying tags into *Personal*, *Subjective* and *Factual* categories. With *Personal*, *Subjective* classes belong to user-space, and *Factual* class belongs to item-space.

Upon verifying the dissimilarity among the user and item perspective, we examined the performance of our design, which takes the observations into account. We use a competitive strategy to find a winning cluster to user, that is, with cluster from user-space and item-space on hand, the winning cluster is the one which has the largest agreement with user tag profile. The recommendation generated in this way can leverage different interests, and therefore give a better result. The preliminary tag profile lengthening stage of our strategy enable us to maximize user's preference, which is essential to deal with the cold-start problem, as it doesn't have prior knowledge of the user.

Given the evident of different conceptual meaning found in user-space and item-space, as well as the benefit of considering them together, a simple tweak to the existing algorithms in this approach can outperform those only considered a single side, while the users of these systems can be more stimulating.

## References


[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.

[2] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993*

*ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

[3] 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.

[4] X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak, and N. Oliver. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 532–539, New York, NY, USA, 2009. ACM.

[5] C.-m. Au Yeung, N. Gibbins, and N. Shadbolt. User-induced links in collaborative tagging systems. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 787–796, New York, NY, USA, 2009. ACM.

[6] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (ldow2008). In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1265–1266, New York, NY, USA, 2008. ACM.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[8] M. de Gemmis, P. Lops, G. Semeraro, and P. Basile. Integrating tags in a semantic content-based recommender. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 163–170, New York, NY, USA, 2008. ACM.

[9] P. De Meo, G. Quattrone, and D. Ursino. A query expansion and user profile enrichment approach to improve the performance of recommender systems operating on a folksonomy. *User Modeling and User-Adapted Interaction*, 20(1):41–86, 2010.

[10] F. Durao and P. Dolog. Extending a hybrid tag-based recommender system with personalization. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1723–1727, New York, NY, USA, 2010. ACM.

[11] J. Gemmell, M. Ramezani, T. Schimoler, L. Christiansen, and B. Mobasher. The impact of ambiguity and redundancy on tag recommendation in folksonomies. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 45–52, New York, NY, USA, 2009. ACM.

[12] S. Givon and V. Lavrenko. Predicting social-tags for cold start book recommendations. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 333–336, New York, NY, USA, 2009. ACM.

[13] G. Gonzalez, J. L. de la Rosa, M. Montaner, and S. Delfin. Embedding emotional context in recommender systems. In *ICDEW '07: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 845–852, Washington, DC, USA, 2007. IEEE Computer Society.

[14] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 540–547, New York, NY, USA, 2009. ACM.

[15] M. Heckner, M. Heilemann, and C. Wolff. Personal information management vs. resource sharing: Towards a model of information behaviour in social tagging systems. In *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)*, San Jose, CA, USA, May 2009.

[16] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[17] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Folkrank: A ranking algorithm for folksonomies. In *Proc. FGIR 2006*, 2006.

[18] A. B. J. A. Konstan, J. Riedl and J. Helocker. Recommender systems: A grouplens perspective. In *Recommender Systems: Papers from the 1998 Workshop*. (AAAI Technical Report WS-98-08), pages 60C64. AAAI Press, 1998.

[19] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 61–68, New York, NY, USA, 2009. ACM.

[20] B. Li and Q. Zhu. The determination of semantic dimension in social tagging system based on som model. *Intelligent Information Technology Applications, 2007 Workshop on*, 1:909–913, 2008.

[21] H. Lieberman. Letizia: An agent that assists web browsing. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 924–929, 1995.

[22] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46, New York, NY, USA, 2007. ACM.

[23] A. Mathes. Folksonomies – cooperative classification and communication through shared metadata. *Computer Mediated Communication, LIS590CMC*, December, 2004.

[24] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semant.*, 5(1):5–15, 2007.

[25] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[26] B. S. Penn. Using self-organizing maps to visualize high-dimensional data. *Comput. Geosci.*, 31(5):531–544, 2005.

[27] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[28] S. Sen, J. Vig, and J. Riedl. Learning to recognize valuable tags. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 87–96, New York, NY, USA, 2009. ACM.

[29] M. Szomszor, C. Cattuto, H. Alani, K. O?Hara, A. Baldassarri, V. Loreto, and V. D. Servedio. Folksonomies, the semantic web, and movie recommendation. In *4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0*, 2007.

[30] E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180, New York, NY, USA, 1993. ACM.

[31] C.-P. Wei, C. C. Yang, and C.-M. Lin. A latent semantic indexing-based approach to multilingual document clustering. *Decis. Support Syst.*, 45(3):606–620, 2008.

[32] R. W. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 363–370, New York, NY, USA, 2009. ACM.

[33] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

# Delay Cascade in Queueing Network of Cardiovascular Care

Li Tao

## Abstract

*Queueing networks are common in complex service systems in general and healthcare systems in particular. Queues, which are arisen from any service stations with limited resources, may have mutual and cascading effects of delay. That means, a small delay in one place may result in long delays elsewhere. Understanding such kind of cascading delays in queueing network will be helpful for understanding the dynamic patterns of waiting lists and efficient healthcare waiting list management. In this paper, we invest delay cascade, or how delays disseminate through queue links in healthcare queueing network. Prior to a further investigation of the dynamic patterns (e.g., when and how the delay cascade will happen) in cardiovascular care, we 1) first identify whether two interacted units have significant relationships in terms of waiting list by Structure Equation Modeling based on the cardiovascular care data collected from fiscal period 2002/03-2005/06 in Ontario, Canada; 2) develop a series Markovian queueing network model to analyze the dynamic processes of patient flows and cascading effects of delay among waiting lists in a simple cardiac care unit network. Parameterized by the real-world data, our preliminary simulation results show that the delay cascade is an important factor in the spreading of wait in the whole healthcare system and its impact on wait is varied depending on the workload of one unit.*

## 1 Introduction

Long waiting list is a notorious problem in public health care [2][4][29] for its massive negative effects, such as increasing mortality and morbidity, decreasing patients' satisfaction, inducing unexpected patient behaviors which thwart regular performances of service providers [14] and etc. In order to balancing demands and resource allocations while avoiding unduly long waiting lists, previous studies on waiting lists focus mainly on some intuitionally impact factors, like modeling patient flow, optimizing resource allocation, improving management strategies [6][18][21], but the questions that whether and how the delays are spreading over all the system are not well answered.

Healthcare system can be regarded as a directed queueing network system. Because it contains an arbitrary, but finite number of queues which located in the units that are connected by functional or temporal relation. Customers, or patients travel through the network and are served at the nodes. Wait may happen not only due to the inadequate capability of a node [6][21][18], unpredictable patient behaviors [12][11][24], but also may result from the delays transferred or cumulated from other mutual interacted nodes. In other words, small delay in one node may result in long delays elsewhere in queueing network.

Cascading delays are common in healthcare queueing network. For example, thirty minutes delay of patient A in Magnetic Resonance Imaging (MRI) test will influence all the patients behind him wait thirty more minutes in the same queue. As well, this delay may also influence the waiting lists of other nodes involved in the sequential path of this patient. For example, with doctor's arrangement, patient A should take MRI test on 9:00 am and take Angiography test on 10:00 am by schedule. Due to the delay of thirty minutes in MRI test, the start time of Angiography is not 10:00 am but 10:20 am. Therefore, delays in the form of wait time spreads across the entire queueing network like a virus. We call this spreading of a piece of wait time along links in a queueing network a delay cascade.

In order to investigate how delays propagate in healthcare queueing network, we start from a simple queueing network with two sequential connected key units– Angiography diagnosis test unit and Coronary Artery Bypass Graft (CABG) treatment unit. Based on the real world data collected from fiscal period 2003/04-2004/05 in Ontario, Canada, we explore the following questions:

(1) Do the wait times of interactive units have a kind of covariance relationship from empirical data?

(2) Does delay spread along links in the queueing network? In other words, does delay influence the variations of wait time in interactive units?

(3) What are the properties of delay dissemination in queueing network (e.g., how the delay of one node impact the delays in subsequent nodes, or length of queues)?

Based on the motivation above, our work in this paper includes:

(1) Adopts Structural Equation Modeling (SEM) approach to qualitatively identifying the causal relations between two interacted units in terms of wait list based on real world data. SEM is a statistical technique for testing and estimating causal relations using a combination of statistical data. By SEM, we want to discover whether the waiting lists of interacted units have causal relations.

(2) Propose a series Markovian queueing network model to demonstrate the delay cascade effect which may be account for the causal relationship of waiting lists of interacted units. Queueing theory has been proven to be very useful to quantitatively study waiting lists [19]. As a specific kind of queueing model, queueing network, which analyzes several queues in a network-like structure, may be suitable for studying healthcare waiting lists because a healthcare system exactly has such kind of network-like structure. By queueing network model, we discover that the one delay may influence the wait time within a unit and across the queueing network.

The remainder of this paper is organized as follows. In the next section, we will talk about the related work. Section 3 is the problem statement. Section 4 analyzes the causal relation of waiting lists by SEM. Section 5 describes our queueing model to study the delay cascade effects in queueing network model. Section 6 is the preliminary experiments. Finally, Section 7 concludes the whole paper.

## 2 Related Work

### 2.1 Characterizing Cascading Effects

Cascade effects have been widely studied in many research areas. In ecosystem, trophic cascade has been studied to understand the population relations of predator and prey in a food web [17]. Cascade failures are characterized and modeled to explain why small initial shocks will trigger the entire system collapses in electrical power network, traffic network and Internet [13]. In social network, cascade effects have been investigated to figure out how information disseminates through social links in social networks and the underline mechanisms [9]. In industrial control system, how to control the cascade time delay has attracted long time attention [20].

### 2.2 Queueing Models of Waiting Lists

Queueing theory, as a useful way to analyze queues, has developed for many years. Fomundam and Herrmann [16] have surveyed a range of queueing models applied to waiting list analysis, resource utilization analysis, and healthcare system design (e.g., appointment systems). They have also considered modeling systems at different scales, including individual units and regional healthcare systems. Schoenmeyr et al.[27] have developed a nontrivial queueing model to predict the dynamics of the operating and recovery rooms under the constraints of capacity (i.e., recovery beds, surgery case volume, recovery time and other parameters). Zhao and Lie [31] have applied the queueing model with a Markov chain and a discrete event simulation to describe the patient flow in an emergency department aiming at intelligent scheduling and reducing emergency department crowding. Creemers and Lambrecht [11] have constructed a queueing model to assess the impact of service outages, to approximate patient flow times, and to evaluate a number of practical applications.

In view of the fact that many healthcare systems include several arbitrary but finite queues in the real world, Cochran [10] has proposed a multi-stage approach to balancing inpatient bed unit utilization within an entire hospital. He has used queueing network and discrete event simulation models to define a step-by-step procedure for analyzing bed planning. Creemers and Lambrecht [12] have developed a decomposition based queueing network model and a Brownian motion based queueing network model to assess the performance in terms of patient flow times at the orthopaedic department in Middelheim hospital. AuYeung et al. [5] have developed a multiclass Markovian queueing network model of patient flows in the accident and emergency department of a major London hospital.

However, so far, there are few works to study the effects and mechanisms of delay cascade in healthcare system. And neither do they apply queueing theory to investigate the effects of delay cascade on waiting lists in the cardiovascular care system based on the real world data.

## 3 Problem Statement

A cardiovascular healthcare system can be regarded as a network by functional and temporal relationships. Based on the cardiovascular treatment guidelines [1][3], a cardiovascular healthcare system can be simplified as a directed graph $G$ with $N$ nodes and $K$ directed links. $G$ is described by the $N \times N$ adjacency matrix $e_{ij}$. If there is a functional or temporal interaction from unit $i$ to unit $j$ (e.g., linked consultation unit and Holter monitor unit means the later one almost visited after consultation), then $e_{ij} = 1$; Otherwise $e_{ij} = 1$.

Patients, according to doctors' suggestions, go through several nodes sequentially. In the patient journey, some key nodes (MRI, PTCA, CABG, and etc.) with scarce resource need appoint before execution. In this paper, we focus on

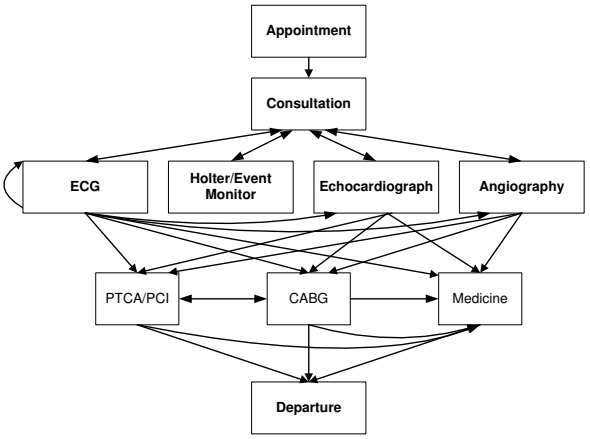



**Figure 1:** A cardiovascular unit network consisting of the units commonly encountered in cardiovascular patient pathway. Among them, Angiography, PTCA/PCI, and CABG are the most important ones due to their high costs and long waiting lists. (ECG: Electrocardiogram; PTCA: Percutaneous transluminal coronary angioplasty; PCI: Percutaneous coronary intervention; CABG: Coronary artery bypass grafting)

the delay cascade in those which need appoint and schedule in advance. The definition of wait time is:

$$w_{pk} = TE_{pk} - TJ_{pk} - TP_{pk} \tag{1}$$

Here, $w_{p,k}$ is the wait time of patient $p$ at node $k$. $TE_{p,k}$ is the actual end-time of a treatment of patient $p$ at node $k$. Similarly, $TJ_{p,k}$ is the time when patient joins the queue of node $k$ and $TP_{p,k}$ is the actual perform-time of this treatment. We assume that the actual perform-time of patient $k$ follows normal distribution, $TP_{p,k} \sim \mathcal{N}(\mu_k, \sigma_k^2)$.

Based on the assumption that the working time of units is 10 hours a day, the delay of patient $p$ at node $k$ is calculated by:

$$d_{p,k} = \begin{cases} 0, \ if \ TE_{p,k} - \widetilde{TS_{p,k}} - \widetilde{TP_{p,k}} < 0 \\ TE_{p,k} - \widetilde{TS_{p,k}} - \widetilde{TP_{p,k}}, \ else \end{cases} \tag{2}$$

Here, $d_{p,k}$ is the delay of patient $p$ at node $k$. $\widetilde{TS_{p,k}}$ is the scheduled start-time for patient $p$ to receive the treatment provided by node $k$. $\widetilde{TP_{p,k}}$ is the expected perform-time of a treatment of patient $p$ at node $k$. We assume that the expected perform-time of patient $k$ also follows normal distribution, $\widetilde{TP_{p,k}} \sim \mathcal{N}(\mu_k, \sigma_k^2)$. $\widetilde{TS_{p,k}}$ is calculated by the equation 3.

The cascade delay may happen in the units with appointment service mechanism and will spread over the queueing networks because:

(1) Patients sequentially travel through several units to receive various services. The movements of patients lead the units have a network-like topology.

(2) Gains of treatment time (e.g., the expected treatment time for patient A is 2 hours while the actual execution time is 1 hour, then it gains 1 hour in this case) do not shorten the wait time of the subsequence patients because the start time of their treatments have been scheduled.

(3) Delay of one patient may propagate to the current queue as well as the queues of sequential units which he will visit in plan. The processes of these two types of delay cascades are shown in figure 2 and figure 3.

Due to lack of complete empirical data related to the units shown in figure1, we start from a graph including two units (i.e., angiography and bypass surgery, which have been identified as the most important and resource-intensive cardiovascular procedures [8][25]), to show the waiting list dynamic process. Specific research questions to be answered are as follows:

(1) Does the queues of interacted units have some kinds of relationships in the real world? If the queues are positively or negatively related, then we need to find out the mechanism underline such kind of relation.

(2) Does delay in one unit result in delays elsewhere? What is the dissemination mechanism of delay in queueing networks?

(3) Can we explain the relationship of waiting lists by cascading delay? To validate how the delay at one unit influence the delays and waiting lists elsewhere, we propose a queueing models to reflect the dynamics and effects statistically.

## 4 Relationship Identification of Interactive Waiting Lists

In this section, we want to explore whether the queues of interactive units have some kinds of relationships by Structure Equation Modeling. SEM is a very general, chiefly linear, chiefly cross-sectional statistical modeling technique. Major applications of SEM include factor analysis, path analysis, regression and correlation structure models [7][22].

### 4.1 Hypothesis

There are many factors reported influence the wait queues for key treatment services. On the supplier-side, capacity of suppliers has been recognized to partially explain the significant regional disparities in access to coronary angiography after accounting for clinical need [30].

$$\widetilde{TS_{p,k}} = \begin{cases} (\widetilde{TS_{p-1,k}} + \widetilde{TP_{p-1,k}})/10 + 1, \ if \ (\widetilde{TS_{p-1,k}} + \widetilde{TP_{p-1,k}} + \widetilde{TP_{p,k}})/10 < \widetilde{TP_{p,k}} \\ \widetilde{TS_{p-1,k}} + \widetilde{TP_{p-1,k}}, \ else \end{cases} \tag{3}$$

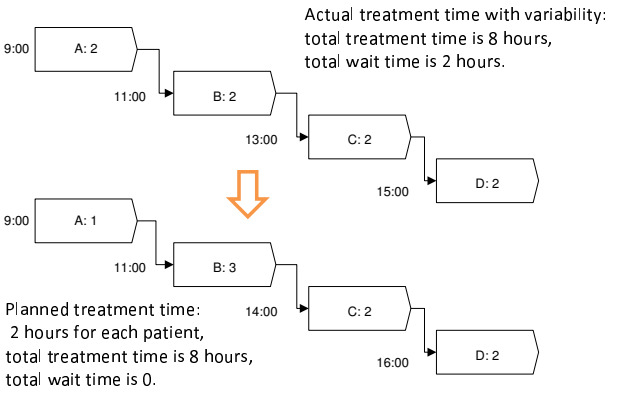



**Figure 2:** Illustration of delay cascade within a unit. The planned treatment time is two hours per person. Actually, the treatment patient B waste one more hour, then this delay will cascade to patient C and D. Although the patient A has gain one hour, it is meaningless to shorten the total wait time because the start time of all the patients are prearranged.

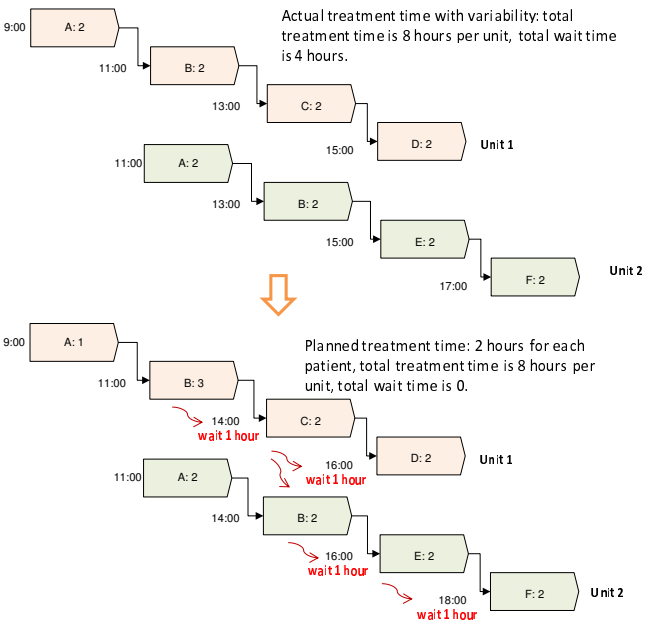



**Figure 3:** Illustration of delay cascade across units. The planned treatment time is two hours per person. Due to the one hour delay of patient B in unit 1, patient E and F will wait one more hour because their treatment are later than patient B according to the prearrangement of unit 2.

On the demander-side, gender and age are two important factors influence the cardiac mortality, mortality, symptoms and ways of diagnosis and treatment, whereafter affect the actual capacity of one service. For example, according to the literatures, women and men are different in perception of symptoms of acute coronary syndromes [15], different in cardiac surgery operation risks [26], as well as in ways of diagnosis tests and treatment [4]. Age, an indicator which accompanies with the structural and functional changes of different body organs, is a significant risk factor in cardiac surgery, strongly correlating with morbidity and mortality [23] [28].

The aim of this section is not only to verify that whether and how gender, age, and actual capacity of a service have direct or indirect influence on waiting list, but also to verify the hypothesis that there may exist a relation between interactive waiting lists. Because units may share patients (i.e., several units may serve the same patient) and may have causal dependencies (i.e., a service may result in another service, for example, consultation often along with testing, and medicine prescription often goes behind). Overall speaking, our hypotheses for SEM verification are:

(1) Age and gender as two observed variables have direct influence on the actual capacity of a service.

(2) Age and gender as two observed variables have direct influence on the waiting list of a service.

(3) Actual capacity of a service has direct influence on the waiting list of a service.

(4) Two interactive waiting lists which are regarded as hidden variables have a causal relation.

(5) The two hidden variables are measured by two observed variables, average median wait time and the time of 90% completed, respectively.

In this case, we use the real world data to see the causal relation of waiting lists of angiography service and CABG service. The hypothetic CABG waiting list causal model of our case is shown in figure 4. In this model, the rectangles denote the observed variables, and ellipses express the unobserved latent variables. The circles named as $z_i$ and $e_i$ are error variables which indicate that factors other than the latent variables affect the result of a measurement related to latent variables and observed variables respectively. The single headed arrows describe the causal connections among variables. And the double headed arrows indicate that the two connected variables have a kind of covariance correlation.

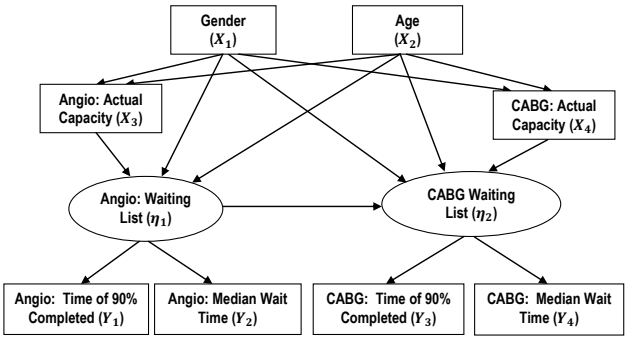



**Figure 4:** The hypothetic CABG waiting list causal model. In this model, we assume that the angiography waiting list and the CABG arrivals' demography have direct influence to CABG waiting list. The number of CABG arrivals is omitted in this model due to its unfolded obvious affect for waiting list. (Angio: Angiography; CABG: Coronary artery bypass grafting; $X_i$: Observed exogenous variables; $Y_i$: Observed endogenous variables; $\eta_i$: Unobserved endogenous variables)

### 4.2 Data Analysis

To validate the hypothesized model, we collect data about angiography and CABG of Ontario in fiscal year of 2003/2004 to 2004/2005 [4]. With the aim of analyzing the impact of age, we divide the population into four age groups: A1(20-39), A2(40-64), A3(65-74), A4(75+). The reliability of the instrument is evaluated using Cronbach's Alpha coefficient. With SPSS 16.0, the Cronbach's Alpha coefficient for the overall observed factors is 0.725 which means the data have high reliability. In the end, in order to eliminate the impact of non-uniform dimensions in analysis, the real world data has been standardized to z-scores.

In this study, the estimated model is displayed in figure 5. The assessment of the model fit is shown in table 1. All analysis were conducted by using the SPSS 16.0 for Windows computer package and Amos 16.0.

**Table 1:** Goodness of fit indices for CABG waiting list causal model.

| Fit Index | Value |
|---|---|
| $\chi^2$ | 17.974 |
| $df$ | 10 |
| $p$ value | 0.055 |
| $GFI$ | 0.859 |
| $NFI$ | 0.899 |
| $TLI$ | 0.851 |
| $RMSEA$ | 0.186 |

Note: $GFI$ = Goodness of Fit Index, $NFI$ = Normed Fit Index, $TLI$ = Tucker-Lewis Index, $RMSEA$ = Root Mean Square Error of Approximation.

As table 1 shows, goodness-of-fit statistics exhibit that the hypothesized model is not well fit the data but acceptable. Because the $\chi^2$ is relatively small and the probability level $p$=0.055 is not bad. $GFI, NFI, TLI$ are close to 0.9 and $RMSEA$ is close to 0.1.

However, from the result, we observe that although the "Angio Waiting List" has the highest effect (because it has the largest regression weight) on "CABG Waiting List", the influence is not significant because the critical ratio is less than 1.96 and the probability is greater than 0.05. The reason of this phenomenon may hidden in the data we have used. Because the data of angiography and CABG is a special case which cannot represent the general situations. In section 5 and section 6, we will further model and investigate the wait time relationship in terms of delay cascade in queueing networks.

## 5 Modeling Delay Cascade in Queueing Network

Our proposed series queueing network model is composed by two M/D/1 service stations aiming to investigate the waiting lists dynamics among units is shown in figure6. Here, M denotes the Markovian arrival rate, D denotes the deterministic service completion time, and 1 denotes the single server. Some basic assumptions of this model are:

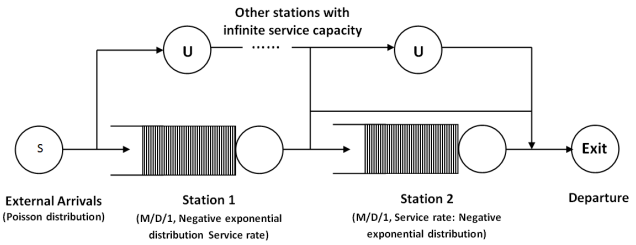



**Figure 6:** A series queueing network model. Node $station1$ and $station2$ are two serial connected units always have queues. $U$ denotes service stations without queues.

(1) An open queueing system with only one entrance. That means, the system has infinity input at the root node (e.g., register unit in healthcare system) but the rest of nodes do not have external arrivals except flow from other nodes (e.g., unit nodes will not consider the referral patients from other healthcare systems).

(2) The external arrivals of angiography is Poisson distribution with parameter $\lambda_k$.

(3) The arrival rate of station 2 is proportional to the arrival of station 1 by state transition parameter $\xi$ ($0 < \xi \leq 1$). There are no external arrivals in the second station.

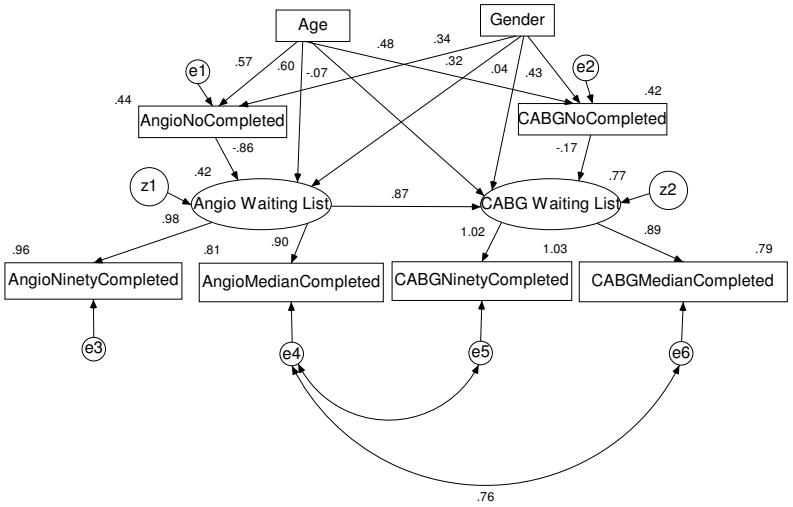



**Figure 5:** Output path diagram for hypothesized CABG waiting list causal model. Numbers beside directed line are standardized regression coefficients. Numbers above the rectangle are the multiple squared covariance of the variables. (Angio: Angiography; CABG: Coronary artery bypass grafting)

(4) Station 1 and 2 always have waiting queues while other service stations (denoted as "U" in figure6) have no queues to eliminate the influence of other units.

(5) Consider First In First Out service discipline. That means, there is no service priority in this model.

(6) Let $\mu_i$ denote the average departure rate of workstation $i$. $\mu_i$ can be considered as negative exponential distribution.

Let $N_i(t)$ denotes task number (e.g., including patient in queue and in process in our scenario) in state $i$ on time $t$. And let $p(a, b : t) = P\{N_1(t) = a, N_2(t) = b\}$, where $p(a, b : t)$ is the state probability that system has $a$ tasks (patients) in station 1 and $b$ tasks (patients) in station 2. Then we can draw the following equations to characterize the queueing process.

When $\Delta t \to 0$, $p(a, b) = \lim_{t\to\infty} p(a, b : t), a, b \geq 0$. The solving process is omitted in this paper.

## 6 Experiments

In this section, we examine the delay dissipation in our sequential queueing network model. According to the cardiac guidelines, the execution time of angiography is around 1 hours, while the time of CABG surgery is normally varied from 2 to 6 hours. The assumptions in our simulation are in accordance with those in section 3 and section 6. In our simulation, the parameters of station 1 (unit 1) are $\mu = 1$ and $\sigma = 0.5$, and $\mu = 4$, and $\sigma = 2$ for station 2 (unit 2). The transfer rate from unit 1 to unit 2 is 0.2, which approximately equals to the rate of service provision from CABG to angiography in 2004/05.

$$
\begin{aligned}
p(0,0:t+\Delta t) &= (1+\lambda\Delta t)p(0,0:t) \\
&\quad + \mu_2\Delta t p(0,1:t) \\
&\quad + (1-\lambda\xi\Delta t)p(1,0:t) \\
&\quad + o(\Delta t), \\
p(a,0:t+\Delta t) &= \lambda\Delta p(a-1,0:t) \\
&\quad + (1-\lambda\Delta t-\mu_1\Delta t)p(a,0:t) \\
&\quad + (1-\lambda\xi\Delta t)p(a+1,0:t) \\
&\quad + \mu_2\Delta t p(a,1:t) \\
&\quad + o(\Delta t)), a>0, \\
p(0,b:t+\Delta t) &= \mu_1\xi\Delta t p(1,b-1:t) \\
&\quad + (1-\lambda\Delta t-\lambda\xi\Delta t-\mu_2\Delta t)p(0,b:t) \\
&\quad + (1-\mu_i\xi\Delta t)p(1,b:t) \\
&\quad + \mu_2\Delta t p(0,b+1:t) \\
&\quad + (1-\mu_1\xi\Delta t+\mu_2\Delta t)p(1,b+1:t) \\
&\quad + o(\Delta t), b>0, \\
p(a,b:t+\Delta t) &= \lambda\Delta p(a-1,b:t) \\
&\quad + (1-\lambda\Delta t-\lambda\xi\Delta t-\mu_1\Delta t-\mu_2\Delta t)p(a,b:t) \\
&\quad + \mu_1\xi\Delta t p(a+1,b-1:t) \\
&\quad + (1-\mu_1\xi\Delta)p(a+1,b:t) \\
&\quad + \mu_2\Delta t p(a,b+1:t) \\
&\quad + (1-\mu_1\xi\Delta t+\mu_2\Delta t)p(a+1,b+1:t) \\
&\quad + o(\Delta t), a,b>0
\end{aligned}
$$

The total wait time of all the patients in unit 1 and 2 are shown in figure 7 and figure 8. We can see that the trend of the wait time for each unit is exponential increase. This reflect the accumulative features of delay in a unit. Specifically, the delays for patients who visit both unit 1 and 2 are shown in figure 9. From this figure, we note that delay definitely cascades in a unit because the delay in unit 1 almost linearly increase. However, it is interesting to note that delay in unit 2 per person will reach to a relatively stable state after a phase of increase. In other words, unit 2 is more robustness than unit 1 to maintain its performance. The properly reason is that the density of workload is smaller than that of unit1. Because the arrivals of unit 2 are less than unit 1 because they proportionally come from unit 1. In addition, less patients will be scheduled to serve in unit 2 than unit 1 because the execution time of unit 2 is much longer than that of unit 1. That means, unit 2 may have some spare time between two scheduled cases. We also think this is a possible reason to explain why the waiting lists of angiography and CABG have no significant relationship from empirical data analysis in section 4.

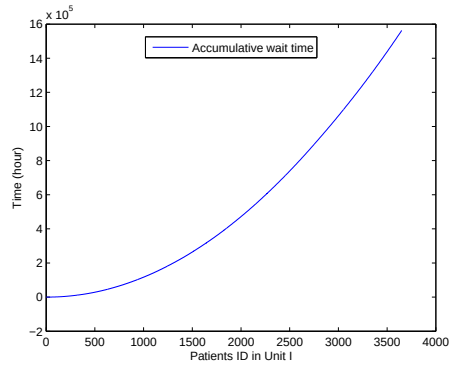



**Figure 7:** The accumulative delays of patients in unit 1.

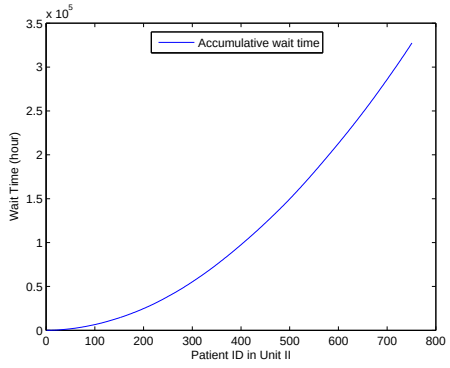



**Figure 8:** The accumulative delays of patients in unit 2.

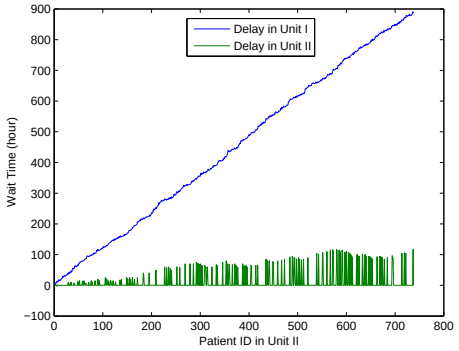



**Figure 9:** The delays respect to the patients who visits two units successively.

## 7 Conclusion

The main contribution of this paper includes two aspects: (1) Employ Structural Equation Modeling to discover the waiting lists relationship of angiography unit and CABG unit based on real world data. Although the results show that there is a biggest regression weight between two waiting lists, the influence of angiography waiting list is not significant on that of CABG. One possible reason for this result is the speciality of the data in our case, which calls for more general data to represent the majority wait time phenomenon in healthcare system. (2) Propose a serial queueing network model to study the effects of delay cascade in cardiac queueing networks. The simulations demonstrate that delay will cascade in the whole queueing network. Delay has accumulative effect which leads to the total wait time increases exponentially in a unit. As well, we discover that the delay cascade has more notable influence on units with heavy service workload than those with light workload per day.

## References


[1] Arrhythmia diagnostic tools. texas heart institute at st.luke's episcopal hospital. Texas Heart Institute at St.Luke's Episcopal Hospital, http://www.texasheart.org/PatientCare/Centers/CCAE/Diagnosis.cfm.

[2] *The Reform of Health Care Systems: A Review of Seventeen OECD Countries*. OECD, 1994.

[3] ACC/AHA guidelines for the management of patients with ST-elevation myocardial infarction-executive summary: A report of the american college of cardiology/american heart association task force on practice guidelines. American Heart Association. http://circ.ahajournals.org/cgi/content/full/110/5/588, 2004.

[4] D. A. Alter, E. A. Cohen, X. Wang, K. W. Glasgow, P. M. Slaughter, and J. V. Tu. Cardiac procedures, in: Jack v. tu

and s. patricia pinfold and paula mccolgan and andreas laupacis, editor. access to health services in ontario: Ices atla 2nd edition. Toronto: Institute for Clinical Evaluative Sciences. http://www.ices.on.ca, 2006.

[5] S. W. M. AuYeung, P. G. Harrison, and W. J. knottenbelt. A queueing network model of patient flow in an accident and emergency department. In *20th Annual European and Simulation Modelling Conference*, pages 60–67, October 2006.

[6] C. Brecht, D. Erik, and J. Belien. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 2009.

[7] B. M. Byrne. *Structural Equation Modeling with AMOS: Basic Concepts, Applications, and Programming (2nd)*. Routledge, 2009.

[8] R. Carroll, S. Horn, B. Soderfeldt, B. James, and L. Malmberg. International comparison of waiting times for selected cardiovascular procedures. *Journal of the American College of Cardiology*, 25:557–563, March 1995.

[9] M. Cha, A. Mislove, B. Adams, and K. P. Gummadi. Characterizing social cascades in flickr. In *Proceedings of the first workshop on Online social networks*, pages 13–18, 2008.

[10] J. K. Cochran. A multi-stage stochastic methodology for whole hospital bed planning under peak loading. *International Journal of Industrial and Systems Engineering*, 1(1-2):8–36, 2006.

[11] S. Creemers and M. Lambrecht. Isr technical report: Healthcare queueing models. Katholieke Universiteit Leuven. http://ideas.repec.org/p/ner/leuven/urnh dl123456789-164227.html, 2008.

[12] S. Creemers and M. R. Lambrecht. Modeling a healthcare system as a queueing network: The case of belgian hospital. Open Access publications from Katholieke Universiteit Leuven. http://ideas.repec. org/p/ner/leuven/urnhdl123456789-120530.html, 2007.

[13] P. Crucitti, V. Latora, and M. Marchiori. Model for cascading failures in complex networks. *Physical Review E*, 69(4):045104, 2004.

[14] R. F. Davies. Waiting lists for health care: A necessary evil. *Canadian Medical Association*, 160(10):1469–1470, May 1999.

[15] H. A. DeVon, C. J. Ryan, A. L. Ochs, and M. Shapiro. Symptoms across the continuum of acute coronary syndromes: Differences between women and men. *American Journal of Critical Care*, 1(17):14–24, August 2008.

[16] S. Fomundam and J. Herrmann. Isr technical report: A survey of queuing theory applications in healthcare, 2007.

[17] K. T. Frank, B. Petrie, J. S. Choi, and W. C. Leggett. Trophic cascades in a formerly cod-dominated ecosystem. *Science*, 308(10):1621–1623, June 2005.

[18] S. H. Jacobson, S. N. Hall, and J. R. Swisher. Discrete-event simulation of health care systems. *Patient Flow: Reducing Delay in Healthcare Delivery*, 2006.

[19] J. L. Jain, S. G. Mohanty, and W. Bohm. *A Course on Queueing Models*. Chapman & Hall /CRC, Boca Raton, 2006.

[20] M. Jankovic. Control of cascade systems with time delay - the integral cross-term approach. In *Proceedings of the 45th IEEE Conference on Decision & Control*, pages 2547–2552, 2006.

[21] J. Jun, S.H.Jacobson, and J.R.Swisher. Application of discrete-event simulation in health care clinics: A survey. *The Journal of the Operational Research Society*, 50(2):109–123, 1999.

[22] P.-W. Lei and Q. Wu. Introduction to structural equation modeling: Issues and practical considerations. *Educational Measurement*, 26(3):33–43, Fall 2007.

[23] A. Mortasawi, B. Arnrich, U. Rosendahl, I. Frerichs, A. Albert, J. Walter, and J. Ennker. Is age an independent determinant of mortality in cardiac surgery as suggested by the euroscore. *BMC Surgery*, 2(8), October 2002.

[24] J. Pirolo, A. Ray, G. Scoville, and B. Amland. Utilization of discrete event simulation in the prospective determination of optimal cardiovascular lab processes. In *Proceedings of the 2009 Winter Simulation Conference*, pages 1916–1926, 2009.

[25] T. J. Ryan. International comparison of waiting times for selected cardiovascular procedures: A commentary on the long queue. *Journal of the American College of Cardiology*, 25:564–566, March 1995.

[26] R. Samalavicius, I. Misiuriene, G. K. G. Norkunas, and A. Baublys. Impact of gender on outcome following coronary artery bypass grafting surgery. *ACTA Medica Lituanica*, 16(3-4), 2009.

[27] T. Schoenmeyr, P. F. Dunn, D. Gamarnik, R. Levi, D. L. Berger, B. J. Daily, W. C. Levine, and W. S. Sandberg. A model for understanding the impacts of demand and capacity on waiting time to enter a congested recovery room. *Anesthesiology*, 110(6):1293–1304, 2009.

[28] G. Spencer, J. Wang, L. Donovan, and J. V. Tu. Report on coronary artery bypass surgery in ontario, fical years 2005/2006 and 2006/2007. Institute for Clinical Evaluative Sciences, In collaboration with the Cardiac Care Network of Ontario. http://www.ices.on.ca/webpage.cfm?site_id=1&org_id=68, 2008.

[29] 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.

[30] H. C. Wijeysundera, T. A. Stukel, A. Chong, M. K. Natarajan, and D. A. Alter. Impact of clinical urgency, physician supply and procedural capacity on regional variations in wait times for coronary angiography. *BMC Health Services Research*, 10(5), 2010.

[31] L. Zhao and B. Lie. Modeling and simulation of patient flow in hospitals for resource utilization. In *49th Scandinavian Conference on Simulation and Modeling*. Oslo University College, October 2008.

# A Rotation-invariant Script Identification based on BEMD and LBP

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

## Abstract

*Script identification is very important to develop the scripts OCR systems. In this paper, we proposed a new algorithm for script identification based on the global and local texture of document images. The BEMD method is used to decompose the image to some components (IMFs) and then the Local Binary Patterns (LBP) method is used to detect the features. Experiments shown the recognition rate based on BEMD-LBP is as well as the LBPV and wavelet based energy feature in 0 angles. At the same time, for the different angles rotation script, the BEMD-LBP feature present some robust adaptive to the rotation script.*

## 1. Introduction

Script recognition is a basic research topic in document analysis, and is also a difficult and time cost problem. The most feature extractors for script rely on the assumption that character can be defined by the local statistical properties of pixel gray levels. Several script analysis systems have been developed.

There have been many script recognition systems, they can be classified into three main approaches, statistics-based [12], [17], [18], texture-based [14], [15] and token-based [13]. The statistics-based approaches identify scripts through the analysis of the distribution of the upward concavity [12] or horizontal projection profile [17], [18]. Texture-based approaches identifies scripts based on the texture detected by some features, such as Gabor filters [14], Gray-level co-occurrence matrix[16] and wavelet-based energy[19]. In [13], the character tokens which specific to different scripts are used for script identification.

The statistics-based approach, which detects upward concavities or horizontal projection profile in an image, is highly sensitive to noise and image quality[15,16]. These approaches usually use the connected components, which should segment the characters before the follow processing. On the other hand, the scripts often have a distinctive visual appearance, so the script document can be considered as a texture. From this, the problem of script identification can be changed into the texture classification problem. For the texture-based approaches, it is not need to extract individual characters, and there is no script-dependent processing.

There have been some texture-based approaches proposed for the script identification, for example, Gabor filter banks[14], wavelet energy[19] and wavelet gray-level co-occurrence matrix[16]. Recently, Empirical mode decomposition (EMD), developed by Huang [1], has been used for the texture analysis and face recognition [2]. EMD 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. EMD has present some better quality than Fourier, wavelet and other decomposition algorithms in extracting intrinsic components of textures because of its data driven property [2, 4].

In this paper, we proposed a new algorithm for script identification based on the global and local texture of document images. The key point is using texture analysis to extract the features. The BEMD method is firstly used to decompose the image to some components (IMFs) and then the Local Binary Patterns (LBP) method is used to detect the features. Experiments shown the BEMD-LBP method can identify scripts accurately and is robust to the text line simultaneously compared with other texture-based approaches.

## 2. Review of BEMD

Empirical Mode Decomposition (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.

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

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

if SD is larger than a threshold $\varepsilon$, 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:

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

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

## 3. A new BEMD based on self-similar

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

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

$x_{mn}$ 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,

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

Where

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

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.

### 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.

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*×*N*. The size of the extend block is *M*×*M*. After extending, the extended image is (*N*+2*M*)×(*N*+2*M*) with middle *N*×*N* block the original image. The original image *I* is divided to *M*×*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. LBP based on BEMD

To classify the rotation script images, we proposed to use the LBP to extract the local features from the IMFs. Local Binary Patterns (LBP) is introduced as a powerful local descriptor for microstructures of images [20]. The LBP operator labels the pixels of an image by thresholding the 3 × 3-neighborhood of each pixel with the center value and considering the result as a binary string or a decimal number.

### 4.1 Local Binary Patterns (LBP)

The LBP operator was originally developed for texture description. The operator assigns a label to every pixel of an image by thresholding the 3x3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor. Figure 1 shows an example of the LBP operator[24].
The form of the resulting 8-bit LBP code can be defined as follows:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n \qquad (8)$$

where $i_c$ corresponds to the gray value of the center pixel $(x_c, y_c)$, into the gray values of the 8 neighborhood pixels, and function s(x) is dedined as:

$$s(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x \leq 0 \end{cases} \qquad (9)$$

From the above processing, the LBP present that it will be not affected by any monotonic gray-scale transformation which preserves the pixel intensity order in a local neighborhood. Each bit of the LBP code has the same significance level and that two successive bit values may have a totally different meaning.

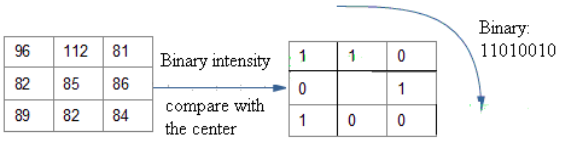



Figure 1 The LBP operator

To deal with textures at different scales, the LBP operator was later extended to use neighborhoods for different sizes [20]. The local neighborhood is extended to as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled allows any radius and number of sampling points[22]. If a sampling point is not in the center of a pixel, it will be rebuilt by bilinear interpolation. The notation (P;R) is defined as the pixel neighborhoods which means P sampling points on a circle of radius of R. Figure 2 shows an example of circular neighborhoods.

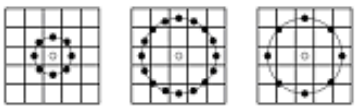


Figure2 The circular (8,1) (16,2) (8,2) neighborhoods

Another extension to the original operator is the definition of so called *uniform patterns* [20]. A local binary pattern is called *uniform* if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular [22]. For example, the patterns 11111111 (0 transitions), 00011000 (2 transitions) and 11100011 (2 transitions) are uniform, the patterns 11001001 (4 transitions) and 01010111 (6 transitions) are not. In the LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all non-uniform patterns are assigned to a single bin[21].

### 4.2 LBP based on BEMD

In this paper, a script image is firstly decomposed by BEMD into several sub-images IMFs, and then, the LBP is used to extract those IMFs. We can use them as a set of the features to classify the script characters.
BEMD is based on the local characteriristic scale of the data, which is able to perfectly analyze the nonlinear and nonstationary signals. The details in the global and local information of the different script are extracted.
LBP has several properties that favor its usage in rotation-invariant script identification. Because of the invariance of the LBP features, the LBP can be suit for the considerable gray-scale variations in images and no normalization of input images is needed. Secondly, the LBP features are very fast to compute. Thirdly, LBP is a nonparametric method, which means that no prior knowledge about the distributions of images is needed. The operator does not require many parameters to be set.
We use the following notation for the script features:
Firstly, the original image I is decomposed to its IMFs $d_k$:

$$I(n) = \sum_{k=1}^{K} d_k(n) + r(n) \qquad (10)$$

Secondly, the $LBP^u(P,R)$ is used to detect the uniform patterns of the IMFs. The subscript represents using the operator in a (P;R)neighborhood. Superscript *u* stands for using only uniform patterns.

Thirdly, the different $LBP_i^u(P,R)$ for $IMF_i$ are combined with weighting rules:

$$LBP^u(P,R) = \sum_{i=1}^{n} w_i * LBP_i^u(P,R) \quad (11)$$

Where $LBP_i^u(P,R)$ indicate the LBP corresponding to the $IMF_i$ . Superscript $u$ reflects the use of rotation invariant 'uniform' patterns. (P;R) is used for pixel neighborhoods which means P sampling points on a circle of radius of R. $w_i$ indicate the corresponding weights, the sum of the those weights is 1.

### 4.3 The classifier based on SVM

Support Vector Machines (SVM) has become a hot research topic in machine learning because of its excellent statistical learning performance. It has been widely applied to pattern recognition. Simply, the principle of constructing the optimal separating hyperplane is that the distance between each training sample and the optimal separating hyperplane is maximum.[20]

Let $\{(x_i, y_i)\}$ $(1 \le i \le N)$ be a linearly separable set. Where, $x_i \in R^d$ , $y_i \in \{-1,1\}$, and $y_i$ are labels of categories. The general expression of the linear discrimination function in *d*-dimension space is defined as $g(x) = w*x + b$, and the corresponding equation of the separating hyperplane is as follows: $w*x + b = 0$.

Normalize $g(x)$ and make all the $x_i$ meet $g(x) \ge 1$, that is, the samples which are closed to the separating hyperplane meet $|g(x)| = 1$.Hence, the separating interval is equal to $2/\|w\|$, and solving the optimal separating hyperplane is equivalent with minimizing $\|w\|$.The object function is as follows:

$$\text{Min}\,\Phi(w) = \frac{1}{2}\|w\|^2 \quad (12)$$

Subject to the constraints:

$$y_i(w \bullet x_i + b) \ge 1, i=1,\ldots,N \quad (13)$$

When adopting Lagrangian algorithm and introducing Lagrangian multipliers $\alpha = \{\alpha_1,\ldots,\alpha_N\}$ , the problem mentioned above can be converted into a quadratic programming problem and the optimal separating hyperplane can also be solved.

Where, $w = \sum_i \alpha_i y_i x_i$ ,

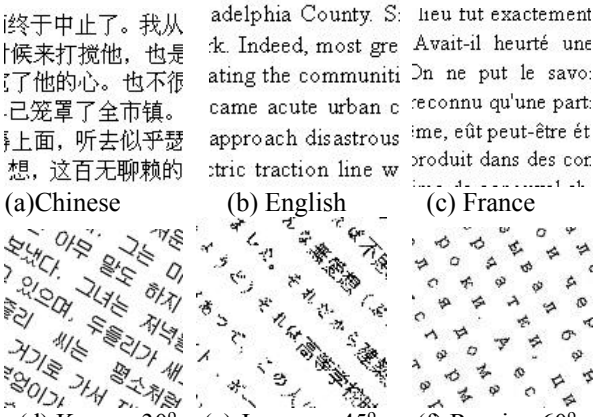
$x_i$ is the sample only appearing in the separating interval planes. These samples are named support vectors and the classification function is defined as follows:

$$f(x) = \text{sgn}\left(\sum_i \alpha_i y_i x_i \bullet x + b\right) \quad (14)$$

For our experiments, we use the RBF kernel because it offers better discrimination than the linear kernel, while using less parameter than the polynomial kernel.

## 5. Experimental results

The proposed algorithm for script identification was tested on a database containing six different script types (English, France, Chinese, Japanese, Russian and Korean). Each script has 250 samples for training, and 500 samples for testing. Some examples of the images are shown in Figure. 3. Each document is scanned on the gray level of 0-255, each 128*128 pixels in size, extracted for each script class.



(a)Chinese (b) English (c) France

(d) Korean 30° (e) Japanese 45° (f) Russian 60°

Figure 3 some samples of the script image

The classifier is based on SVM described as above[23]. The proposed feature is compared with the wavelet based energy feature [19] and the LBPV feature [21].

Table 1 shows the result of the identification. As the table shown, the recognition rate based on BEMD-LBP is as well as the wavelet based energy feature in 0 angles. For the English and the France, which are both Latin word, the similarity of their textures is large than other scripts, the recognition rate of BEMD-LBP and LBPV are lower than the wavelet based energy feature. At the same time, for the different angles rotation script, the BEMD-LBP feature present some robust adaptive to the rotation script. The wavelet energy features shown to be sensitive to the script rotation.

| Angles | English | France | Chinese | Japanese | Russian | Korean | Average |
|---|---|---|---|---|---|---|---|
| $0^o$ BEMD-LBP | 92.51 | 91.53 | **97.38** | **96.24** | **96.52** | **98.56** | 95.45 |
| $0^o$ wavelet[19] | **95.31** | **94.63** | 94.21 | 93.58 | 93.73 | 94.71 | 94.36 |
| $0^o$ LBPV[21] | 88.55 | 89.42 | 94.60 | 93.78 | 94.77 | 95.13 | 92.70 |
| $30^o$ BEMD-LBP | 74.55 | 73.38 | **84.38** | **83.53** | **82.13** | **86.58** | 80.75 |
| $30^o$ wavelet[19] | **75.14** | **74.27** | 78.54 | 77.43 | 75.39 | 80.27 | 76.84 |
| $30^o$ LBPV[21] | 70.43 | 69.26 | 82.24 | 82.61 | 71.25 | 81.51 | 76.21 |
| $45^o$ BEMD-LBP | **50.54** | **49.36** | **58.53** | **57.87** | **55.44** | **60.78** | 55.42 |
| $45^o$ wavelet[19] | 40.23 | 39.76 | 44.38 | 43.93 | 42.35 | 43.57 | 42.37 |
| $45^o$ LBPV[21] | 43.14 | 42.48 | 49.14 | 49.76 | 48.22 | 50.50 | 47.20 |
| $60^o$ BEMD-LBP | **70.16** | **72.61** | **79.40** | **78.53** | **77.50** | **79.11** | 76.21 |
| $60^o$ wavelet[19] | 65.37 | 67.93 | 68.30 | 73.75 | 72.42 | 76.73 | 70.75 |
| $60^o$ LBPV[21] | 65.10 | 64.83 | 75.58 | 69.46 | 74.22 | 77.20 | 71.06 |
| $90^o$ BEMD-LBP | 88.45 | 87.73 | **93.56** | **92.10** | **91.83** | **94.27** | 91.32 |
| $90^o$ wavelet[19] | **89.53** | **88.27** | 90.40 | 92.33 | 88.52 | 89.18 | 89.70 |
| $90^o$ LBPV[21] | 85.14 | 84.57 | 90.21 | 85.68 | 85.94 | 90.44 | 86.99 |

Table 1 Recognition rate for different angles script by proposed method and wavelet energy [19], LPBV[21]

## 6. Conclusion

Script identification is very important for development of multi-script OCR systems. In this paper, a new global-local feature is proposed to identify the word-wise printed script. Firstly, the BEMD decompose the image to different IMFs, which present different scale information of the original image. And then the LBP method is used to detect the local information of IMFs. The experimental shown the combined BEMD-LBP features is robust adaptive to the rotation script compared to wavelet energy features.

In the following work, we will improve the features to be adaptive to the Latin font. On the other hand, this new texture feature will be applied to classify the nature texture images, and other classers will be used for the 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] A.L. Spitz, "Determination of Script and Language Content of Document Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 3, pp. 235-245, Mar. 1997.

[13] J. Hochberg, L. Kerns, P. Kelly, and T. Thomas, "Automatic ScriptIdentification from Images Using Cluster-Based Templates," IEEETrans. Pattern Analysis

and Machine Intelligence, vol. 19, no. 2,pp. 176-181, Feb. 1997.
[14] T.N. Tan, “Rotation Invariant Texture Features and Their Use in Automatic Script Identification,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 7, pp. 751-756, July 1998.
[15] A. Busch, W.W. Boles, and S. Sridharan, “Texture for Script Identification,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pp. 1720-1732, Nov. 2005.
[16] P.S. Hiremath, S. Shivashankar, "Wavelet based co-occurrence histogram features for texture classification with an application to script identification in a document image" Pattern Recognition Letters, pp. 1182–1189, vol.29, 2008
[17] U. Pal and B.B. Chaudhury, “Identification of Different Script Lines from Multi-Script Documents,” Image and Vision Computing, vol. 20, no. 13-14, pp. 945-954, 2002.
[18] A.M. Elgammmal and M.A. Ismail, “Techniques for Language Identification for Hybrid Arabic-English Document Images,” Proc. Sixth Int’l Conf. Document Analysis and Recognition, pp. 1100-1104, 2001.
[19] Li Zeng, Yuanyan Tang ,Tinghuai Chen, "Multi-scale wavelet texture-based script identification method " JISUANJI XUEBAO. Vol. 23, no. 7, pp. 699-704. July 2000
[20] T. Ojala, M. Pietikäinen, and T. T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with Local Binary Pattern,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, 2002.
[21] Zhenhua Guo, Lei Zhang, David Zhang "Rotation invariant texture classification using LBP variance(LBPV) with global matching" Pattern Recognition, 706–719 vol 43. 2010
[22] Guoying Zhao and Matti Pietikainen, "Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 915-928, 2007
[23] Library for Support Vector Machines http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[24] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen, "Face recognition with local binary patterns",Lecture Notes in Computer Science, ECCV ,pp.469-481,2004