PROCEEDINGS

The HKBU 4th Computer Science Postgraduate Research Symposium

July 3, 2006

PG Day 2006

Department of Computer Science
Hong Kong Baptist University

The 4th HKBU-CSD Postgraduate Research Symposium (PG Day) Program

<table>
<tr><th colspan="3">July 3 Monday, 2006</th></tr>
<tr><th>Time</th><th colspan="2">Sessions</th></tr>
<tr><td>09:45</td><td colspan="2">On-site registration</td></tr>
<tr><td>10:00-10:05</td><td colspan="2">Welcome: Prof. YiuWing Leung, Acting Head of Computer Science Department (LMC 514)</td></tr>
<tr><td>10:05-11:35</td><td colspan="2">Session A: (Chair: ChuiYing HUI)
Pattern Recognition</td></tr>
<tr><td></td><td colspan="2">• Protecting Face Biometric Data with Error-Correcting Codes
YiCheng Feng
• Hand-Written Chinese Character Based on SVM
Jianjia Pan
• Application of SVM in the Pattern Recognition
LiMin Cui</td></tr>
<tr><td>11:35-14:00</td><td colspan="2">Noon Break</td></tr>
<tr><td>14:00-16:30</td><td>Session B1: (Chair: Zhenyu HE) (LMC 514)
Intelligent Informatics</td><td>Session C1: (Chair: Minji WU) (LMC 510)
Networking</td></tr>
<tr><td></td><td>■ A New Quantization Scheme for Multimedia Authentication
HaoTian Wu
■ Value Directed Compression for POMDP Planning with Belief State Analysis
Xin Li
■ A Semi-supervised SVM for Manifold Learning
ZhiLi Wu
■ Personalized Spam Filtering with Classifier Emsemble
ChiWa Cheng
■ Agent based Testbed for Relax Criteria Negotiation
KaFung Ng</td><td>■ Lightweight Piggybacking for Packet Loss Recovery in Internet Telephony
WingYan Chow
■ The Client-Based Framework For Privacy-Preserving Location-Based Data Access
Jing Du
■ An Adaptive and Intelligent Controller for Clustered Web Servers
KaHo Chan
■ Location Estimation in Library Environment based on Enhanced Fingerprint Approach
ManChung Yeung</td></tr>
<tr><td>18:30</td><td colspan="2">Best Paper & Best Presentation Awards Announcement via Email</td></tr>
</table>

TABLE OF CONTENTS

Session A: Pattern Recognition

Session B: Intelligent Informatics

Session C: Networking

Protecting face biometric data with error-correcting codes

Feng Yicheng

June 23, 2006

Abstract

Biometric security has been largely regarded and researched within the latest 20 years. However, researchers focus in face recognition have not paid enough attention to the security of face biometric data. We propose a cryptographic algorithm to protect face recognition process against attack, which applies the error-correcting codes. The original face feature vectors are transformed to new vectors so as to satisfy the requirements of error-correcting codes, and then coded and encrypted for protection.

1 Introduction

Reliable user authentication has become more and more important. Various methods have been implemented to enhance user authentication security, including passwords, PINs, and biometrics. Comparing with passwords and PINs, biometric is much more convenience and secure. The information contained in biometric can range from several hundred bytes to over a million bytes, quite larger than the information contained in passwords or PINs. And biometrics are physical characters which is highly linked with people. So biometrics-based authentication system enhance higher security than passwords or PINs. On the other hand, biometric is very convenience because it just exits on human's body. Various biometrics-based authentication systems have been widely applied in different domains, including fingerprint, iris, face and so on.

It's very important to protect the security of biometric systems. From last 20 years this issue is largely and adequately considered by different researchers[1, 2, 3]. However, most of the researchers only concern themselves with security of biometric systems using fingerprint or iris, especially fingerprint. The security of face recognition system receives very little attention. In our paper we want to proposed a method which can be applied to face recognition to protect its biometric data, and do not affect the performance of the original system much.

2 Review

The basic architecture of applying biometrics into smartcard authentication system is to encrypt the biometric data with a private key and store it in the smartcard. In authentication, the data is decrypted and then compared with the one the applicant presents. However, this structure is not secure because the public key is easy to get. As a result, the biometric data will be revealed if the attacker gets the public key and uses it to decrypt the stolen smartcard.

In order to solve this problem, we may store the hash form of the biometric template but not the raw data in the smartcard. In authentication, when the biometric data is extracted by sensor, it is also hashed with the same algorithm and then compared with the one stored in smartcard. So attackers can only get the hash of the template even if the cryptographic system is compromised, which is useless. However, the matching between biometric data in hash form is so sensitive to variation that even if the template changes a little, the matching will fail. In order to solve this problem, the error correcting codes are applied. The error correcting codes modify the variation of biometric data before authentication, thus we can apply hash function into biometric authentication systems. There are many research papers which have used error correcting codes. [3, 2, 4, 5] The diagram of this process is shown in figure 1:

In enrollment, the biometric data is first scanned by

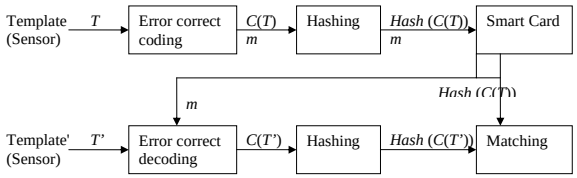


Figure 1: The flowchart of the error-correcting security enhanced system.

the sensor and then transformed to a feature vector T. An error-correcting coding algorithm is done, with the coded data $C(T)$ and a released message m come out. The coded data is encrypted with some one way function such as hashing, the result is denoted as $Hash(C(T))$, and it is stored in database with the released message. In authentication, the feature vector T' is scanned and extracted and sent to do an error-correcting decoding process with the released message m. The decoded result is denoted as $C(T')$. The decoded data is encrypted (denoted as $Hash(C(T'))$) and compared with the data stored in database ($Hash(C(T))$). The error correcting coding/decoding process is used to correct the variation of T' from T by the information contained in the released message m. Different error-correcting codes have different error correcting abilities.

There are many schemes follow this approach, such as the error-correcting scheme by Davida $etal$. [3], the fuzzy commitment scheme by Juels and Watternberg [2], the fuzzy vault scheme by Juels and Sudan [4] and the finger vault scheme by Clancy $etal$. [5]. The first two schemes leads to some data leakage, and the error-correcting abilities are not high. The fuzzy vault scheme has strong security level, but it is not suitable for face biometrics.

While we apply this approach to face recognition, there is one problem we have to face, thus, the characteristic of the face feature vectors is not suitable, or the variation of face feature vectors extracted by PCA/LDA algorithms is too large, larger than the error-correcting abilities of most common error-correcting codes, for example, Hamming codes [9], Reed-Solomon codes [6], BCH codes [7], and so on. These codes can only correct variation with Hamming distance, while the face feature vectors have variation of Euclidean distance. The most suitable code may be the sphere packing codes [8], which correct the original vector to the nearest codeword and is applied by the fuzzy commitment scheme [2]. However, for the large variation of face biometric data, the distance between code words should be set very large, thus, the number of code words will be limited and the security level of this coding algorithm is weak.

To solve the variation problem, we are thinking of transforming the original face feature vectors to some new vectors, which can remain the characteristic of classification, thus, vectors in different classes have large distance while vectors in the same class have only little distance. However, the distance which is used to distinguish different classes is not Euclidean but Hamming distance, so that we can apply most of the common and useful error-correcting codes such as RS or BCH codes to our protecting scheme.

3 Variation problem solving

Consider the simplest condition, the feature vector dimension is only 2, and there are only 2 classes. Thus, there are two classes of points we have to distinguish, which is shown in figure 1. The weights of the two classes are P and Q. In order to do this, a point A is randomly chosen, and the distances between A and points in the two classes are computed. A threshold t is given. If $AP > AQ$, we set the rule:

$$If \quad AB > t, \quad B \quad belongs \quad to \quad class \quad 1.$$
$$If \quad AB < t, \quad B \quad belongs \quad to \quad class \quad 2.$$

In this way, we can distinguish the two classes with the help of the random generated point A. It is easy to see that when A lies on the line PQ, and t equals AO while O is the center of PQ, this distinguish scheme can get the almost best performance. We may call the generated point A "distinguish point".

Then consider the popular condition, while there are m classes of feature vectors with each feature vector of length n. Treat the feature vectors as points in n-dimensional space, we can still use the same way to distinguish these points. Distinguish points are generated and

distances are computed. Because the points of the same class will be close in the feature space, thus, the distances of these points to some distinguish point should also be approximate. Distances of different classes may be much different (this is depending on the location of the distinguish point). More points are generated for distinguishing, the distinguishing ability will be better.

Use these points to distinguish the original feature vectors and to transform them into new vectors. For each feature vector v, compute its distance d to each generated distinguish point A_i. A threshold t is used to judge the result:

$$If \quad d \leq t, \quad m_i = 0;$$
$$If \quad d > t, \quad m_i = 1.$$

After the distances to all distinguish points are computed and judged, we get a binary string $m_1 m_2 m_3 m_4 m_5 \dots m_p$, in which p is the number of distinguish points. Thus, the original feature vector v is transformed into a new binary vector $m_1 m_2 m_3 \dots m_p$. For two feature vectors in the same class, the transformed binary vectors should be nearly the same because the computed distances are approximate. For two feature vectors in different classes, it is much possible that the transformed vectors are very different when p is very large, such as 200 1000. large p means that it is much possible that there are many distinguish points locating in good positions that they can distinguish the two vectors well.

The process is as follows:

- Face image scanned and feature vector v extracted.
- p random points $A_1, A_2, A_3 \dots A_p$ are generated.
- Compute the distances from v to Ai.
- Use threshold t to judge the distances, and get a binary value m_i.
- Construct binary string $m_1 m_2 m_3 m_4 m_5 \dots m_p$.

4 Parameter choosing and scheme modification

4.1 Variation considering modification

The variation problem is actually highly weakened after vector transformation. This is because we use the distances to distinguish points to representing the original vector, and use some thresholds to quantize these distances. After this quantization small variation will be eliminated. However, there may be one condition that the variation of a feature vector just covers the threshold t. For example, feature vectors v_1 and v_2 belongs to the same class, but the distance between v_1 and distinguish point A is a little smaller than t, but distance between v_2 and A is a little larger than t. Then the quantization can not solve this variation.

To solve this problem, a variation range r is defined and the rule for transformation is modified a little:

$$If\ d < t - r/2,\ m_i = 0;$$
$$If\ d > t + r/2,\ m_i = 1;$$
$$If\ t - r/2 \leq d \leq t + r/2,\ m_i = \phi.$$

The transformed vector is then not binary string, but a binary string with some bits of value ϕ. In comparison, the bits with value ϕ in the vector are not considered, thus, even the corresponding bits in the other vector are not ϕ, it is not treated as error. With this modification, we can solve the variation problem described above.

4.2 Points generation

From the above discussion we can see that the location of points will affect the distinguish ability of the transformed vectors. If you want to distinguish two different feature vector P, Q, it is better to choose distinguish point that locates in the extened line of PQ. Assume the database has face images of m classes. Compute the average of each class and get vectors or points $P_1, P_2, P_3 \dots P_m$. For class 1, we should distinguish it with class 2, 3, 4... m. So there are $m - 1$ points and thresholds chosen:

$$A_1 = a_1P_1 + (1-a_1)P_2, t_1 = (a_1 - 0.5)|P_1P_2|.$$
$$A_2 = a_2P_1 + (1-a_2)P_3, t_2 = (a_2 - 0.5)|P_1P_3|.$$
$$A_3 = a_3P_1 + (1-a_3)P_4, t_3 = (a_3 - 0.5)|P_1P_4|.$$
$$\dots$$
$$A_{m-1} = a_{m-1}P_1 + (1-a_{m-1})P_m,$$
$$t_{m-1} = (a_{m-1} - 0.5)|P_1P_m|.$$

From the equation, we can see that A_1 lies on the extended line of P_1P_2, A_2 lies on the extended line of P_1P_3, and so on. Thus, A_1 can distinguish class 1 with class 2, A_2 can distinguish class 1 with class 3, and so on. All these $m-1$ points can then distinguish class 1 with all the other classes.

4.3 Thresholds decision

The thresholds t of the scheme should be carefully chosen to enhance the performance of the authentication. The simplest way to decide the thresholds is to use the same t for all the distinguish points, but this too simple setting may cause the whole system far away from the best performance. Actually we can use many ways to set the thresholds. For example, we may compute the average distance from the distinguish point to the feature vectors, and set it as the threshold to that point. However, these two ways do not show how the thresholds affect the error rate. They are just determined by people's will, but do not by certain reason. In our scheme, we consider a special way making more sense to decide the thresholds, which will get a better performance than the previous two ways.

Assume there are totally m classes with their average feature vectors $O_1, O_2, O_3 \dots O_m$ (treated as points in feature space). In our scheme we generate mp random points $A_1, A_2, A_3 \dots A_{mp}$. The corresponding mp thresholds are determined by

$$t_i = |O_qA_i|, \quad (q = int((i-1)/m) + 1.)$$

Thus, for each average feature vector O_q, $|O_qA_{(q-1)m+1}|$, $|O_qA_{(q-1)m+2}|$, $|O_qA_{(q-1)m+3}|$ $\dots |O_qA_{qm-1}|$ are set as the corresponding thresholds.

After this setting, consider a feature vector (point) M in class 1. M belongs to class 1 means that M is close to O_1, thus,

$$|MA_i - O_1A_i| < \varepsilon, \quad i = 1, 2, \dots p.$$

in which ε is a small scalar. If

$$r/2 > \varepsilon,$$

then

$$|MA_i - t_i| = |MA_i - O_1A_i| < r/2, \quad i = 1, 2, \dots p,$$

thus, the first p bits of the transformed vector should be ϕ. Then all the feature vectors in class 1 will be transformed to binary strings with first p bits ϕ, thus, the same. As the same, the $p + 1_{st}$ to $2p_{st}$ bits of the binary strings transformed from class 2 will be the same as ϕ, the $2p + 1_{st}$ to $3p_{st}$ bits of the binary strings transformed from class 3 will be the same, and so on. In other words, this thresholds setting method can make sure p bits in the transformed binary vectors from the same class to be the same, thus, decreases the FRR.

5 Transformation scheme design

In the previous discussion we have proposed two advanced scheme to transform the original face feature vectors into new ones for error-correcting process. One is the points generation scheme, the other one is the thresholds decision scheme. In this chapter, we describe the whole process of these two schemes, both applying the variation considering modification.

5.1 Points-manual-choosing scheme

In this scheme we use the previous points choosing method, and also applies the variation range setting. The whole recognition process is as follows:

1. Enrollment:
   - Feature vectors are extracted from the face images of the training data.
   - For each class, compute the average vector with the training data.
   - For class i, Use these average vectors, generate the $m-1$ points and corresponding thresholds for class i.

- Set a variation range parameter r.
- Using these $m-1$ distinguish points, thresholds and r, transform the average vector of class i to a binary string with some bits with value ϕ.
- The transformed string is stored in database with the generated points, thresholds and r.

2. Authentication:
   - Extract the face feature vector from the applicant's face.
   - If the applicant claims that he is person i, then take out the $m-1$ distinguish points corresponding to class i from database with the corresponding thresholds and r.
   - Transform the extracted feature vector to a binary string with the distinguish points.
   - Compare the string with the stored one, which represents person i (using Hamming distance).

5.2 Thresholds specifying scheme

1. Enrollment:
   - Feature vectors are extracted from the face images of the training data.
   - For each class, compute the average vector with the training data.
   - Generate mp distinguish points randomly, in which m is the number of class and p is a parameter waiting for decision.
   - Use these distinguish points and average vectors to compute the corresponding mp thresholds.
   - Set a variation range parameter r.
   - Using these distinguish points, thresholds and r, transform the average vectors to binary strings with some bits being ϕ.
   - The transformed strings are stored in database with the generated points, thresholds and r.

2. Authentication:
   - Extract the face feature vector from the applicant's face.
   - Transform the feature vector to a binary string with the stored distinguish points, thresholds and r.
   - Compare the string with the stored one, which represents person i (using Hamming distance).

6 Experiment results

In our experiment we have tried 4 kinds of authentication schemes in order to compare the performance of these scheme: 1) the original authentication scheme; 2) authentication scheme with the basic transformation; 3) authentication scheme with the manual-points-generation transformation; 4) authentication scheme with the thresholds specifying transformation. We uses the traditional LDA authentication algorithm and ORL database [] for testing.

The authentication result of the traditional LDA authentication is shown in figure 2. The result of the basic transformation is shown in figure 3. Results for the two advanced modifications are shown in figure 4 and 5. Because different schemes have different kinds of thresholds, we mainly compare the cross-over error rate. From figure 2 we can see that the cross-over error rate of the original LDA algorithm is about 6%. Figure 3 shows the error rate of the basic transformation is about 7%. From figure 4, error rate of the manual-points-generation transformation is the best one, about 3%. In figure 5 we can see that the thresholds specifying transformation scheme can get a cross-over error rate of about 6%. As a result, we can see that after transformation, the manual-points-generation scheme can get the best performance, even better than the original LDA authentication. This is because it uses a different classifier, which is different with different class. The second one is the thresholds specifying scheme, whose error rate is nearly the same with the original LDA algorithm. The basic transformation may increase the cross-over error rate of about 1%. Thus, both the manual-points-generation scheme and the thresholds specifying scheme are much feasible and the manual-points-generation scheme is best. However, this scheme needs large capacity of database, which is unnecessary in the thresholds specifying scheme.

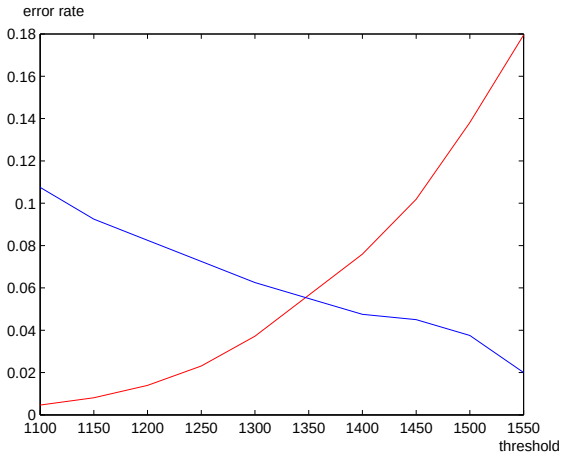


Figure 2: Authentication result for the original LDA algorithm.

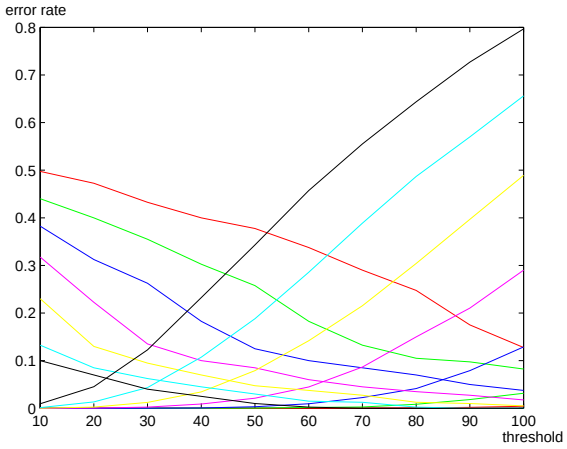


Figure 3: Authentication result for the basic transformation scheme.

7 Conclusion

In this paper, for the purpose of security enhancement, we have proposed two kinds of methods to transform the original face feature vector to new binary vectors, which can be used in error-correcting coding/decoding process and protection. The proposed two schemes well preserve the characteristic of distribution of the original face feature vectors, in which one scheme can almost remain the performance, the other one can even enhance it. After transformation, we can do error-correcting coding/decoding and encryption to protect the biometric data.

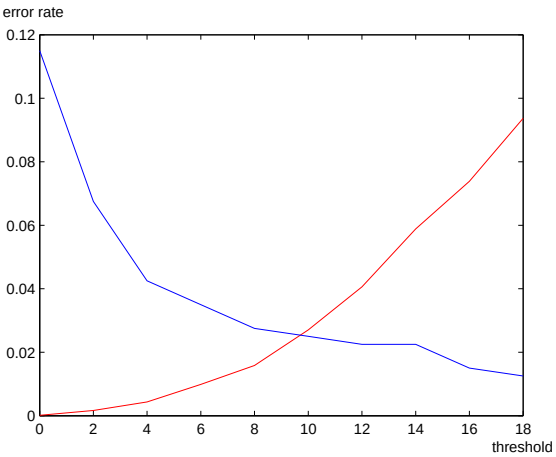


Figure 4: Authentication result for the manual-points-choosing scheme.

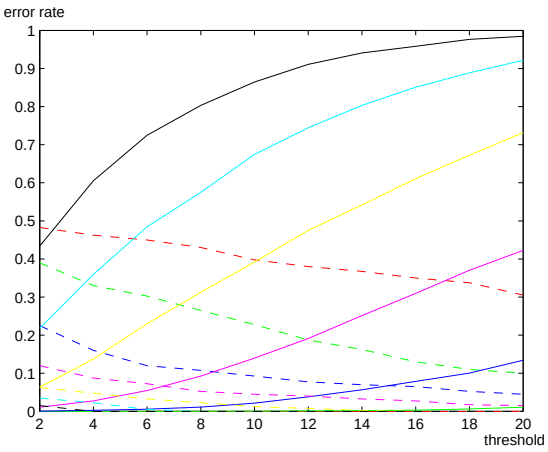


Figure 5: Authentication result for the thresholds specifying scheme.

References

[1] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *End-to-End Security*, vol.40, 2001.

[2] A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," *Sixth ACM Conference on Computer and Communications Security*, pages 28-36, ACM Press. 1999.

[3] G.I. Davida, Y. Frankel, and B.J. Matt, "On enabling secure applications through off-line biometric identification," *IEEE Symposium on Privacy and Security*, pp. 148-157, 1998.

[4] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," *Proceedings of IEEE Internation Symposium on Information Theory*, p.408, 2002.

[5] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," *Proceedings of IEEE Internation Symposium on Information Theoryin Proc. ACMSIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop*, pp.45-52, 2003.

[6] J.I. Hall, "Generalized Reed-Solomon Codes," *Notes on Coding Theory*, 2003.

[7] J.I. Hall, "Cyclic Codes," *Notes on Coding Theory*, 2003.

[8] J.I. Hall, "Sphere Packing and Shanron's Theorem," *Notes on Coding Theory*, 2003.

[9] J.I. Hall, "Hamming Codes," *Notes on Coding Theory*, 2003.

Handwritten Chinese Character Recognition Based on SVM

Jianjia Pan, and Y.Y.Tang
Department of Computer Science
Hong Kong Baptist University
jjpan@comp.hkbu.edu.hk

Abstract

***Abstract**: In recent years, the thorniest question that the off-line handwritten of Chinese character distinguished in pattern recognition region, has obtained the many research results. But, the handwriting Chinese character recognition was still considered was one of most difficult questions of writing recognition domain. This paper presents an application of SVM in small-set off-line handwritten Chinese characters recognition. Paper introduces the basic theory of SVM theories and algorithms, then discusses the theories and algorithms related to multi-class classification of SVM. Then software LibSVM is proposed for handwritten Chinese characters training. The results are also compared with Euclidean Distance classifier. This indicates that the SVM method can improve recognition rate and therefore has more practicability.*

Ⅰ. Introduction

Pattern recognition of handwritten words is a difficult problem, not only because of the great amount of variations involved in the shape of characters, but also because of the overlapping and the interconnection of the neighboring characters. Furthermore, when observed in isolation, characters are often ambiguous and require context to minimize the classification errors. The existing development efforts have involved long evolutions of differing classification algorithms, usually resulting in a final design that is an engineering combination of many techniques.

The problem of handwritten word recognition consists of many difficult sub problems and each requires serious effort to understand and resolve. One of the most important problems in segmentation-based word recognition is assigning character confidence values for the segments of words to reflect the ambiguity among character classes. Many design efforts for character recognition are available, based nominally on almost all types of classification methods such as neural nets, linear discriminate functions, fuzzy logic, template matching, binary comparisons, etc. The choice of one or another nominal method for evaluating features is as important as the choice of what features to evaluate and method for measuring them.

As a new machine learning method, Support Vector Machines is an effective method for pattern recognition. Support Vector Machines (short for SVM), AT&T Bell laboratory V. Human and Vapnik expounds one new machine learning method according to statistical learning theory. It has initially displayed performance better than the methods before, in solving small sample learning question, non-linear and high dimensional pattern recognition, SVM displays many unique superiorities. Its basic thought may summarize as: first transform the input space to a high dimensional space through the nonlinear transformation. Then get the most superior linear classification surface in this new space, and the realization of this kind of nonlinear transformation is through the definition of suitable inner product function.

According to the smallest structure risk criterion, in the premise that the training sample classify-cation error is minimum, it would enhance the promoted ability of the classifier as far as possible. Thinking from the implementation angle, the training support vector machine

equally to solve a quadratic programming question with linear constraint, to get the maximum distance between hyperplanes of two patterns in the separation characteristic space, Moreover it can ensure the obtained solution be global optimum. So the classifier of hand-written characters based on support vector machine would absorb the distortion of the character, thus it will have good exudes and the promoted ability.

Ⅱ .Theory basic

1. SVM theory

Support Vector Machines (SVM) has become a hot research topic in the international machine learning field because of its excellent statistical learning performance. It has been widely applied to pattern recognition, regression analysis, function approximation, et al, and has also been successfully applied to load forecast and malfunction diagnosis of electric power system. Simply, SVM can be comprehended as follows: it divides two specified training samples which belong to two different categories through constructing an optimal separating hyperplane either in the original space or in the projected higher dimensional space. The principle of constructing the optimal separating hyperplane is that the distance between each training sample and the optimal separating hyperplane is maximum.

There are two conditions, linearly separable situation and linearly inseparable situation, under which the principle of SVM is introduced as follows separately.

Under the linearly separable condition, a binary classification task is taken into account. 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:

$$\text{Min} \quad \Phi(w) = \frac{1}{2}\|w\|^2 \tag{1}$$

Subject to the constraints:

$$y_i(w \bullet x_i + b) \ge 1, \ i = 1,\ldots, N \tag{2}$$

When adopting Lagrangian algorithm and introducing Lagrangian multipliers $\alpha = \{\alpha_1, ..., \alpha_N\}$, 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:

$$f(x) = \text{sgn}\left(\sum_i \alpha_i y_i x_i \bullet x + b\right) \tag{3}$$

Under the linearly inseparable condition, on the one hand, SVM turns the object function into as follows through introducing slack variable ξ and penalty factor C

$$\text{Min} \quad \Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{N} \xi_i\right) \tag{4}$$

On the other hand, SVM converts the input space into a higher dimensional space through linear transform in which the optimal separating hyper-plane can be solved.

Additionally, the inner product calculation under the linearly separable condition is turned into $K(x_i, x_j) = \Phi(x_i) \bullet \Phi(x_j)$,

where $K(x_i, x_j)$ is defined as inner product in Hilbert space and it is named kernel function here. Thus the final classification function is represented as follows.

$$f(x) = \text{sgn}\left(\sum_i \alpha_i y_i K(x, x_i) \bullet x + b\right) \tag{5}$$

2. SVM multi-class methods

The multi-class classification problem refers to assigning each of the observations into one of k classes. As two-class problems are much easier to solve, many authors propose to use two-class classifiers for multi-class classification.

The Support Vector Machines (SVM) was originally designed for binary classification. How to effectively extend it for multi-class classification is still an on-going research issue. Currently there are two types of approaches for multi-class SVM. One is by constructing and combining several binary classifiers while the other is by directly considering all data in one optimization formulation. Up to now there are still no comparisons, which cover most of these methods.

The formulation to solve multi-class SVM problems in one step has variables proportional to the number of classes. Therefore, for multi-class SVM methods, either several binary classifiers have to be constructed or a larger optimization problem is needed. Hence in general, it is computationally more expensive to solve a multi-class problem than a binary problem with the same number of data. Up to now experiments are limited to small data sets. In the follow, there is a decomposition implementation for two 'all-together' methods: [1], [2] and [3] .We then compare their performance with three methods based on binary classification: 'one-against-all', 'one-against-one', and DAGSVM.

Note that it was pointed out in [4], that the primal forms proposed in [1],[2] are also equivalent to those in [4][5] . Besides methods mentioned above, there are other implementations for multi-class SVM, for example [6][7]. However, due to the limit of space here we do not conduct experiments on them. An earlier comparison between one-against-one and one-against-all methods is in [8].

The earliest used implementation for SVM multi-class classification is probably the one-against-all method. It constructs *k* SVM models where *k* is the number of class. The *i*th SVM is trains with all of the examples in the *i*th class with positive labels, and all other examples with negative labels. Thus given *l* training data $(x_1, y_1), \ldots, (x_l, y_l)$, where $x_i \in R^n$, $i=1,\ldots,l$ and $y_i \in \{1,\ldots,k\}$ is the class of x_i ,the *i*th SVM solves the following problem:

$$\min_{\omega^i, b^i, \zeta^i} \quad \frac{1}{2}(\omega^i)^T \omega^i + C\sum_{j=1}^{l} \zeta_j^i \qquad (6)$$

$$(\omega^i)^T \phi(x_j) + b^i \geq 1 - \zeta_j^i, \text{ if } y_j = i,$$

$$(\omega^i)^T \phi(x_j) + b^i \leq -1 + \zeta_j^i, \text{ if } y_j \neq i,$$

$$\zeta_t^{ij} \geq 0 \text{ ,j=1,…,}l.$$

Where the training data x_i are mapped to a higher dimensional space by the function Φ and C is the penalty parameter.

Minimizing $(\omega^i)^T \omega^i / 2$ means that we would like to maximize $2/\left\|\omega^i\right\|$, the margin between two groups of data. When data are not linear separable, there is a penalty term which can reduce the number of training errors. The basic concept behind SVM is to search for a balance between the regularization term $C\sum_{j=1}^{l} \zeta_j^i$ and the training errors.

After solving (6) , there are *k* decision functions: $(\omega^1)^T \phi(x) + b^1, \ldots, (\omega^k)^T \phi(x) + b^k$.
We say *x* is the class which has the largest value of the decision function:
Class of

$$x \equiv \arg\max_{i=1,\ldots,k} ((\omega^i)^T \phi(x) + b^i) \quad (7)$$

Practically we solve the dual problem of (6) whose number of variables is the same as the number of data in (6). Hence *k l*-variable quadratic programming problems are solved.

Another major method is called the one-against-one method. It was introduced in [9], and the first use of this strategy on SVM was in [10][11]. This method constructs k(k-1)/2 classifiers where each one is trained on data from two classes. For training data from the *i*th and the *j*th classes, we solve the following binary classification problem:

$$\min_{\omega^{ij}, b^{ij}, \zeta^{ij}} \quad \frac{1}{2}(\omega^{ij})^T \omega^{ij} + C\sum_t \zeta_t^{ij} \qquad (8)$$

$$(\omega^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \zeta_t^{ij} \text{ ,if } y_t = i,$$

$$(\omega^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \zeta_t^{ij} \text{ ,if } y_t = j,$$

$$\zeta_t^{ij} \geq 0.$$

There are different methods for doing the future testing after all k(k-1)/2 classifiers are constructed. In [10], suggested use the following method: if $\text{sign}((\omega^{ij})^T \phi(x_t) + b^{ij})$ says *x* is in the *i*th class, then the vote for the *i*th class is added by one. Otherwise, the *j*th is increased by one. Then we predict *x* is in the class with the largest vote. The voting approach described above is also called the 'Max Wins' strategy. In

case that two classes have identical voted, thought it may not be a good strategy, now we simply select the one with the smaller index.

Practically we solve the dual of (8) whose number of variables is the same as the number of data in two classes. Hence if in average each class has *l*/*k* data points we have to solve k(k-1)/2 quadratic programming problems where each of them has about 2*l*/*k* variables.

The third algorithm discussed is the Directed Acyclic Graph Support Vector Machines (DAGSVM) proposed in [12]. Its training phase is the same as the one-against-one method by solving k(k-1)/2 binary SVMs. However, in the testing phase, it uses a rooted binary directed acyclic graph which has k(k-1)/2 internal nodes and *k* leaves. Each node is a binary SVM of *i*th and the *j*th classes. Given a test sample *x*, starting at the root node, the binary decision function is evaluated. Then it moves to either left or right depending on the output value. Therefore, it goes through a path before reaching a leaf node which indicates the predicted class.

An advantage of using a DAG is that some analysis of generalization can be established. There are still no similar theoretical results for one-against-all and one-against-one methods yet. In addition, its testing time is less than the one-against-one method.

III Experiments

The characteristic extraction: Select tentative data for hand-written Chinese character. In this paper we select the hand-written Chinese number characters. Preprocess these Chinese characters using the model match, binarization, thinning, denoising and so on. Use the appropriate characteristic extraction method to extract the Chinese character's vector characteristic as the support vector machine input sample. Divide each Chinese character image those were binarizated and normalizatied, again do wavelet transformation to these divided images, obtain the wavelet coefficients of the Chinese character as the vector characteristic.

Data format normalization: Make normalized processing of the extracted image characteristic, to adapt the request of SVM training software.

Overlapping confirmation: The overlapping confirmation basic idea is: Divide the sample collection to into two subsets, one group (training subset) trains the classifier, another group (examination subset) exam the training (which estimate the error of the trained classifier), then the basis the exam result to estimate the effect of classifier, then adjusts the related parameter of the classifier. Carries on the training and adjust again as before. When the error achieves the ideal value, then obtains classifier for goal classifier [13].

The software LibSVM is proposed for hand-written Chinese characters training. The multi-class method is 'one-against-one'. LibSVM is a SVM tool developed by Chih-Chung Chang and Chih-Jen Lin. LIBSVM is a library for support vector machines (SVM). Its goal is to help users to easily use SVM as a tool.
The experiment procedure is as follows:
Data format normalization, data value normalization, use RBF function $K(x,y)=e^{-\gamma\|x-y\|^2}$ as kernel function, Train the data to create a model with svmtrain by overlapping confirmation. Predict new input data with svmpredict and get the result.

In experimental system, the recognition of the Chinese character to have 11 kinds, every kind of Chinese character has collected 100 different written samples. Altogether has the sample number is 1100. Parts of training samples are shown in Figure 1.

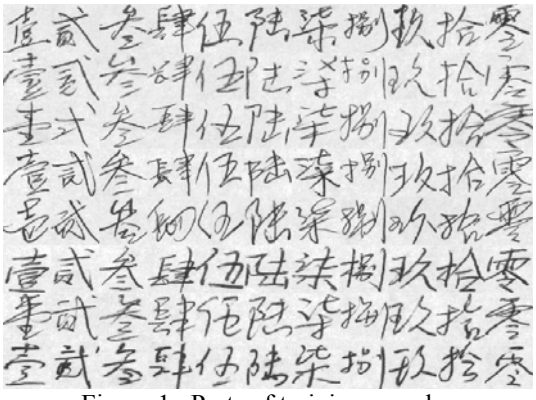


Figure 1 Parts of training samples

Table 1 has listed the SVM and Euclidean Distance classifier the result that obtains from the classified method comparison. As it shows, in the small sample situation, the SVM method has the high recognition rate compared with Euclidean Distance classifier.

Table 1 The recognition rate comparison of SVM and Euclidean Distance method

Method	SVM	Euclidean Distance
Accuracy	93.3%	84.6%

Ⅳ Conclusions and Future Works

In this paper, we introduce the basic theory of SVM theories and algorithms, then discuss the theories and algorithms related to multi-class classification of SVM. Though the experiment, we could see the SVM method has a high recognition rate in handwriting Chinese character classification. According to different case and application, use different multi-class methods and characteristic extraction methods, the effect would be better.

The SVM kernel function is due to the different case. The SVM theory didn't give a way to say which kernel function is the best kernel function. So there are many research in kernel function.

SVM is a new technology based on kernel to solve the question learning form samples. And the problem learning form the samples is an ill posed problem, it can be solved though regularization method transformation to a proper posed problem. Reproducing kernel and the reproducing kernel Hilbert space (RKHS) play a important role in function approximation and regularization theory. But, different function approximation question fits to different approximation function. So it is very important to format special kernel that is suit to special approximation function characteristic. SVMs based on different kernel would solve different questions and applications. Riesz kernel, especially wavelet kernel, is widely applicable. It is operation significance to format the reproducing kernel suited to this approximation function characteristic. In future works, we would research the kernel function of SVM, improve characteristic extraction method, and according to handwritten Chinese characters, study adapted SVM multi-class methods.

Ⅴ REFERENCES

1. VANPNIK V.N Statistical Learning Theory[M]. New York: John Wiley and Sons,
2. J.Weston and C.Watkins. Multi-class support vector machines, Proceedings of ESANN99 ,Brussels,.D.Facto Press
3. KCrammer and Y.Singer. On the learnability and design of output codes for multi-class problems. In Computational Learning Theory, pages35-46, 2000
4. Y.Guermeur. Combining discriminate models with new multi-class SVMs. Neuro COLT Technical Report NC-TR-00-086, LORIA Campus Scientifique,
5. E.J.Bredensteiner and K.P.Bennett. Multicategory classification by support vector machines. Computational Optimizations and Applications, pages53-79.1999
6. J.Kinder,and, E.Leopold, and,G.Paass. Multi-class classification with error correcting coeds., Treffen der GI-Fachgruppe 113,Maschinells Lernen
7. E.Mayoraz and E.Alpaydin . Support vector machines for multi-class classification. In IWANN(2),pages833-842.1999
8. K.K.Chin Support vector machines applied for speech pattern classification. Master's thesis, University of Cambridge
9. S.Knerr, L.Personnaz, and G.Dreyfus.Single-layer learning revisited: a stepwise procedure for building and training a neural network .Neurocomputing: Algorithms, Architectures and Applications. Springer-Verlag
10. J.Friedman Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University
11. U.Krebel. Pairwise classification and support vector machines. Advances in Kernel Methods – Support Vector Learing, pages 255-268,Cambridge,MA,MIT Press.
12. J.C.Platt, N.Cristianini, and J.Shawe-Taylor. Large margin DAGs foe multi-class classification. Advances in Neural Information Processing Systems, volume 12, pages 574-553, MIT Press, 2000
13. Bian Zhaoqi,Zhang Xuegong, Pattern Recognition tsinghua University Press

Application of SVM in the Pattern Recognition

Limin Cui

Abstract

A support vector machine (SVM) is a new, powerful classification machine and has been applied to many application fields, such as pattern recognition in the past few years .We give an overview of the basic idea underlying SVM and some new methods.

1. Introduction

Support Vector Machines (SVM) is created by Vapnik (1995). SVM is binary classifiers of objects represented by points in R^n . Now these pattern classifiers are popular because of a number of good features. The formulation embodies the Structural Risk Minimization (SRM) principle, and opposed to the Empirical Risk Minimization (ERM) approach. SVM is based on statistical learning. Because a lot of real-world objects can be represented as points in R^n , and multi-class classifiers can be built by employing arrays of SVMs, the technique can be applied to many classification problems

2. Theory of SVM

We'll introduce the theory of SVM in the section, including linearly separable case, linearly nonseparable case, and nonlinear case through a two-class classification problem [1-4]. Assume that a training set S is given as

$$S = \{x_i, y_i\}_{i=1}^n \quad (1)$$

Where $x_i \in R^n$, and $y_i \in \{-1 \mid +1\}$, such that

$$w^T x_i + b \geq 1 \quad for \quad y_i = +1$$

$$w^T x_i + b \leq 1 \quad for \quad y_i = -1 \quad (2)$$

$w \in R$ is the weight vector, and bias b is a scalar. If the inequalities in Eq. (2) hold for all training data, it is said to be a linearly separable case. SVMs maximize the margin of separation ρ between classes In the learning of the optimal hyperplane, where $\rho = 2/\|w\|$. We can solve the following constrained optimization problem for the linearly separable case

$$\text{Minimize } \Phi(w) = \frac{1}{2} w^T w \quad (3)$$

$$\text{Subject to } y_i(w^T x_i + b) \geq 1 \quad i = 1,2,\cdots,n \quad (4)$$

Quadratic programming (QP) can solve the above constrained optimization problem. However, if the inequalities in Eq. (2) do not hold for some data points in S , the SVM becomes linearly nonseparable. Then the margin of separation between classes is said to be soft since some data points violate the separation conditions in Eq. (2). To set the stage for a formal treatment of nonseparable data points, the SVMs introduce a set of nonnegative scalar variables, $\{\xi_i\}_{i=1}^n$ into the decision surface

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1,2,\cdots,n \quad (5)$$

To find an optimal hyperplane for a linearly nonseparable case, we can solve the following constrained optimization problem

$$\text{Minimize } \Phi(w,\xi) = \frac{1}{2} w^T w + C\sum_{i=1}^n \xi_i \quad (6)$$

$$\text{Subject to } y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1,2,\cdots,n \quad (7)$$

$$\xi_i \geq 0, i = 1,2,\cdots,n \quad (8)$$

where C is a positive parameter.

It is difficult to find the solution of Eq. (6)-(8) by QP when it becomes a large-scale problem. We can introduce a set of Lagrange multipliers α_i and β_i for constraints (7) and (8), the primal problem becomes to find the saddle point of the Lagrangian. Thus, the dual problem is

$$\text{Maximize } Q(\alpha) = \sum_{i=1}^n \alpha - \frac{1}{2}\sum_{i=1}^n \sum_{i=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (9)$$

$$\text{Subject to } \sum_{i=1}^n \alpha_i y_i = 0 \quad (10)$$

$$0 \leq \alpha_i \leq C, i = 1,2,\cdots,n \quad (11)$$

These Lagrange multipliers can be obtained by using the constrained nonlinear programming with the equality constraint in Eq. (10) and those inequality constraints in Eq.(11). The Kuhn-Tucker (KT) condition plays a key role in the optimization theory and is defined by

$$\alpha_i[y_i(w^T x_i + b) - 1 + \xi_i] = 0, i = 1,2,\cdots,n$$

$$\beta_i \xi_i = 0, i = 1,2,\cdots,n$$

Where $\beta_i = C - \alpha_i$.

SVM has a important advantage that it can map the input vector into a higher dimensional feature and thus can solve the nonlinear case. We can choose a nonlinear mapping function $\varphi(x) \in R^M$, where M > N, the SVM can construct an optimal hyperplane in this new feature space. K(x, xi) is the inner-product kernel performing the nonlinear mapping into feature space

$$K(x, x_i) = K(x_i, x) = \varphi(x)^T \varphi(x)$$

Then the dual optimization problem is
Maximize

$$Q(\alpha) = \sum_{i=1}^{n} \alpha - \frac{1}{2} \sum_{i=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x) \qquad (12)$$

Subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$

$0 \le \alpha_i \le C, i = 1,2,\cdots,n$

Using Kernel functions, we can classify as follows

$$x \in \begin{cases} positive \quad class, \quad if \quad g(x) > 0 \\ negative \quad class, \quad if \quad g(x) < 0 \end{cases}$$

where the decision function is

$$g(x) = sign\left(\sum_{SVs} \alpha_i y_i K(x, x_i) \right).$$

We have three common types of kernels in SVM,

$$(1 + x^T x_i)^p, \exp(-\frac{1}{2\sigma^2} \|x - x_i\|^2), \tanh(s_0 x^T x_i + s_1)$$

3. Applications of Pattern Recognition

3.1. Principles of Pattern Recognition

The pattern recognition is component of sensors, feature extraction system, classification system.

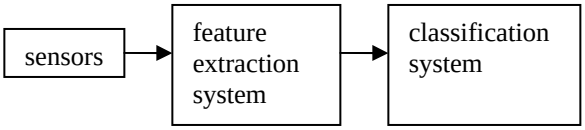


Figure 1 Main component of pattern recognition system

3.2. The New methods of SVM

3.2.1 ν -SVM

Scholkoph etc [7] have advanced ν -SVM method, where optimization problem as follows

$$\begin{cases} \min\limits_{w,b,\xi} \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum\limits_{i=1}^{n} \xi_i \\ s.t. \quad y_i(w \cdot \phi(x_i) + b) \le \rho - \xi_i \\ \xi_i \ge 0 \\ \rho > 0 \end{cases}$$

Then the dual optimization problem is

$$\begin{cases} \max\limits_{\alpha} Q(\alpha) = -\frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ s.t. \quad 0 \le \alpha_i \le 1/n \\ \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \\ \sum\limits_{i=1}^{n} \alpha_i \ge \nu \end{cases}$$

According to KKTtheory, optimization points satisfy

$$\sum_{i=1}^{n} \alpha_i = \nu$$

For Bound Support Vector, $\alpha_i = 1/n$. We have

$(N_{BSV} / n) \le \sum_{i=1}^{n} \alpha_i = \nu$ for N_{BSV} bound support vector. However for support vector, $\alpha_i \le 1/n$.

We have $\sum_{i=1}^{n} \alpha_i \le N_{SV} / n$ for N_{SV} support vector, so $(N_{BSV} / n) \le \nu \le (N_{SV} / n)$.

3.2.2 LS-SVM

Suykens etc [8] have advanced Least Squares Support Vector Machines (LS-SVM), where optimization problem as follows

$$\begin{cases} \min\limits_{w,b,\xi} \frac{1}{2}\|w\|^2 - v\rho + \frac{1}{2} r \sum\limits_{i=1}^{n} \xi_i^{\,2} \\ s.t. \quad y_i(w \cdot \phi(x_i) + b) = 1 - \xi_i \end{cases}$$

such that linear equation set

$$\begin{bmatrix} 0 & y^T \\ y & Q + r^{-1} I \end{bmatrix}_{(n+1)\times(n+1)} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ e \end{bmatrix}$$

Where $e \in R^n$, and e is a vector which elements are 1; $I \in R^{n\times n}$ is unit matrix. $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_n]^T \in R^n$; $y = [y_1, y_2, \cdots, y_n]^T \in R^n$; $Q = [q_{ij}]_{n\times n} \in R^n$; $q_{ij} = y_i y_j K(x_i, x_j)$. LS_SVM reduced complexity, but lost sparse-the advantage of SVM.

3.2.3 W-SVM

In real world, sometimes we need high standard of samples classification, sometimes low standard. So we can adopt different coefficient C, this method is said to weighted SVM (W-SVM) [9]. The optimization problem as follows

$$\begin{cases} \min\limits_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{n} s_i \xi_i \\ s.t. \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1,2,\cdots,n \end{cases}$$

where s_i is weight coefficient. Then the dual optimization problem is

$$\begin{cases} \max\limits_{\alpha} Q(\alpha) = \sum\limits_{i=1}^{n} \alpha_i - \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ s.t. \quad 0 \leq \alpha_i \leq C s_i \\ \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

3.2.4 DirectSVM

For above SVM, we must solve optimization problem by quadratic programmingor linear equation set. Roobaert [10] has advanced direct SVM. DirectSVM adopt heuristic search method that can search SVM in the all training set.

3.3. Application

Pattern recognition techniques are widely used in various domains. For example, medicine, security, military, etc. SVM is a valid tool for pattern recognition. Edgar Osuna et al have proved how to use a support vector machine for detecting vertically oriented and unconcluded frontal views of human faces in grey level images in literature [5]. Schölkopf at al [6] compared SVM with Gaussian kernel to Radial Basis Function Classifiers for handwritten digits.

4. Experiment

Now give a simple example of LS-SVM. Suppose X and Y are vectors, and

$$X = 2 \cdot rand(30,2) - 1.1,$$

where $rand(30,2)$ denote the matrix that have 30 rows and 2 column. Y is the column vector which element is corresponding value of symbolic function. Suppose

$$Y = sign(\sin(X(:,1) + X(:,2))$$

By LS-SVM method, we can get

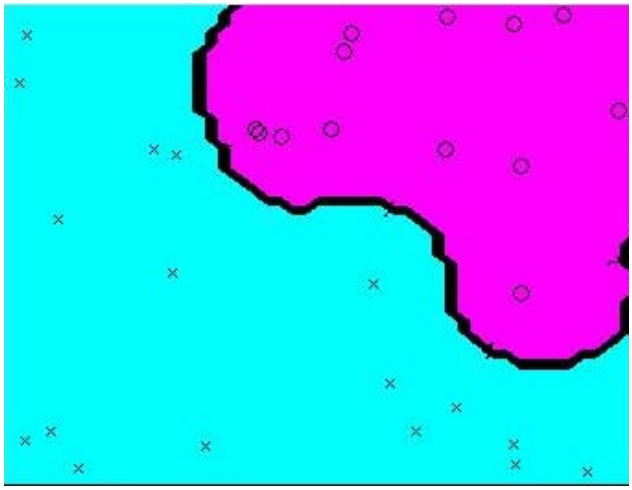


Figure 1 the LS-SVM result

In Figue 1, we classified two kind of point.

5. Future

Sometimes we imminently need accurate detection and classification, we always study new method. Now I study a novel method based on wavelet packet decomposition and support vector machines for detection and classification of power quality disturbances. Wavelet

packet decomposition is mainly used to extract features of power quality disturbances for classification; and support vector machines are mainly used to construct a multi-class classifier which can classify power quality disturbances according to the extracted features of power quality disturbances.

6. Conclusion

SVM can be used of a tool of problem of pattern recognition. Support Vector Machines exhibit an excellent generalization performance, and that they can be successfully applied to a wide range of pattern recognition problems.

References

[1] J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, Vol. 2, pp. 121-167, 1998.
[2] C. Corts and V. N. Vapnik, "Support Vector Networks," Machine Learning, Vol. 20, pp. 273-297, 1995.
[3] Simon Haykin, Neural Networks: A Comprehensive Foundation, Second Edition, New Jersey: Prentice-Hall, 1999.
[4] V. N. Vapnik, Statistical Learning Theory, New York: John Wiley & Sons, 1998.
[5] E. E. Osuna, R. Freund, F. Girosi, *Support Vector Machines: Training and Application*, C.B.C.L. Paper No. 144, MIT, 1997
[6] B. Schölkopf et al *Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers*, MIT 1996
[7] B Scholkoph, A J Smola, L Bartlettp, New support vector algorithms [J]. Neural Computation, 2000, 12(5): 1207-1245.
[8] J A K Suykens, J Vandewale, Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999, 9(3): 293-300.
[9] H C Chew, R E Bogner, C C LIM, Dual ν-support vector machine with error rate and training size beasing [A]. Proceedings of 2001 IEEE Int Conf on Acoustics, Speech, and Signal Processing [C]. Salt Lake City, USA:IEEE, 2001, 1269-1272.

A New Quantization Scheme for Multimedia Authentication

Hao-tian Wu

Abstract

As used in fragile watermarking to embed a tamper-proof watermark, quantization has become a promising way for multimedia authentication because of its simplicity and flexibility. However, in some cases, only part of illegal modifications made to the watermarked content can be detected by using the quantization method. In this paper, a new quantization scheme is proposed to detect them more efficiently. Given a set of embedding primitives representing the authenticity and integrity of media content, a watermark can be embedded by slightly modifying them in a sequential way. The proposed scheme does not affect the capacity, imperceptibility and security of fragile watermarking algorithms, but improves the fragility of the embedded watermark. Nevertheless, slight changes of the resulting primitives can still be allowed without altering the embedded data. The proposed scheme is applied to a high-capacity fragile watermarking algorithm, whereby a new one is generated for mesh authentication. Several aspects of the new algorithm are investigated and compared with those of the former one. Numerical results are given to show the efficacy of the proposed quantization scheme for multimedia authentication.

1 Introduction

With the rapid growth and dissemination of multimedia works, such as digital images, audio and video streams, and 3D models, it has been a real need to verify their authenticity and integrity [4]. Traditional data authentication is performed by appending a hash-based signature to the file, which is encrypted with a private key and decrypted with the corresponding public key. However, cryptographic algorithms are extremely sensitive to outer processing because a single bit error will lead the resulting value to vary. As for multimedia authentication, there are some additional requirements, such as tamper localization and semi-fragility. Therefore, new methods are desired to satisfy these requirements.

As a branch of information hiding [1], digital watermarking has been proposed for a variety of applications. In this paper, we concentrate on fragile watermarking that can be used to verify the integrity and authenticity of media content. As shown in Fig. 1, a watermarked object O' is generated by imperceptibly embedding a watermark m into an original object O. The embedding process is controlled by a key K and a parameter set α as represented in the embedding function $O' = E_{K,\alpha}(O, m)$. The watermarked object O' may be changed by some manipulations n, resulting in the modified object $\hat{O}$, from which the watermark information $\hat{m}$ is extracted by the extraction function $\hat{m} = D_{K,\alpha}(\hat{O})$ with the knowledge of K and α. Hence, generic watermarking can be modelled as a constrained communication problem with side information [2, 3], such as K and α.

Figure 1. A general model of digital watermarking is given, where m is a watermark to be embedded into an original object O with a key k and a parameter set α. O' is the watermarked object, which may be processed by some manipulations n to generate the modified object $\hat{O}$. $\hat{m}$ is the information extracted from $\hat{O}$ with the knowledge of K and α.

In fragile watermarking, the extracted watermark $\hat{m}$ is expected to be different from the original one m if the watermarked object O' has been tampered. Depending on the applications, the embedded watermark may be sensitive to all the outer processing, or robust against those manipulations that preserve the content of the watermarked object. In the latter case, the watermark is called semi-fragile because the value of $\hat{m}$ will be different from m only when O' has been processed by illegal manipulations. In addition, the capability of tampering localization may be achieved by exploiting inherent properties of media content, such as the block-wise algorithms in [5, 6] for digital images. Therefore, fragile watermarking capable of tampering localiza-

tion or allowing some content-preserving manipulations has become an efficient way to multimedia authentication.

In this paper, we take the authentication of polygonal meshes for instance, which are used for geometry representation, such as the cultural heritage recording like [7], CAD models, and structural data of biological macromolecules [8]. In the literature, a variety of watermarking algorithms (e.g.[9]-[22]) have been proposed to embed the watermarks into meshes. Among them, only a few algorithms (e.g.[9]-[14]) are presented for authentication purpose. In [9], a fragile watermarking algorithm based on lookup tables (LUTs) is addressed by Yeo and Yeung. However, the embedded watermark is sensitive to outer processing that preserves the mesh content, such as Rotation, uniformly Scaling, and Translation transformations (denoted as RST hereinafter). By adopting the work in [9], Lin et al. embed a watermark with resistance to vertex reordering in [10], but RST transformations are still not allowed. Moreover, several fragile watermarking algorithms based on quantization have been proposed to allow some content-preserving manipulations whereas detecting illegal ones. As shown in [11, 12, 13, 14], the embedded watermarks are resistant to RST transformations and mantissa truncation of vertex coordinates to a certain degree, but sensitive to illegal modifications. In the rest of this paper, we name the host signal that is modified to embed a watermark as the embedding primitive, and the embedding primitive that has been slightly modified to embed data as the resulting primitive.

In some cases, only part of illegal modifications to the resulting primitives can be detected by fragile watermarking based on quantization. In this paper, we try to improve the fragility of the embedded watermark by proposing a new quantization scheme. Given a set of embedding primitives representing the authenticity and integrity of media content, a watermark can be embedded by slightly modifying them in a sequential way to generate the watermarked content. We analyze the conditions in which the modifications to the resulting primitives can be detected, as well as slight changes of the resulting primitives that can be allowed without altering the embedded data. The proposed scheme is applied to our preliminary work in [14], whereby a new fragile watermarking algorithm is presented for mesh authentication. Since a fragile watermarking algorithm with high information rate is suitable for authentication applications [23], the distance from a vertex to the centroid of its traversed neighbors is therefore chosen as the embedding primitive so that high capacity (about 1 bit/vertex, higher than those in [12, 13]) is achieved. A new quantization method is utilized in the proposed scheme so that it is hard to estimate the quantization step from the resulting primitives. By numerically reserving a margin around the quantization grid, slight changes of the resulting primitives can be allowed after applying the proposed quantization scheme.

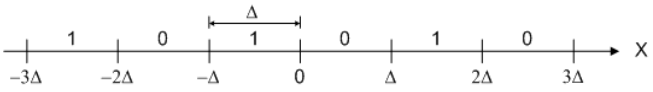


Figure 2. A binary number can be embedded by quantizing the embedding primitive X with the quantization step Δ.

As a result, the embedded watermark is robust against some content-preserving manipulations, but more sensitive to illegal manipulations.

The rest of this paper is organized as follows. In Section II, a new quantization scheme is proposed for authentication applications, after introducing a quantization method for fragile watermarking. Section III presents a new fragile watermarking algorithm for mesh authentication by using the proposed scheme. Experimental results are given and discussed in Section IV. Finally, we draw a conclusion in Section V.

2 A New Quantization Scheme for Multimedia Authentication

In this section, a quantization method will be introduced so that it is hard to estimate the quantization step from the resulting primitives, whereas slight changes of the resulting primitives can be allowed by numerically reserving a margin around the quantization grid. Then we address the case that only part of illegal modifications to the watermarked content can be detected in quantization-based fragile watermarking. After that, a new quantization scheme will be proposed to improve the fragility of the embedded watermark so that illegal modifications can be detected more efficiently.

2.1 A Quantization Method for Fragile Watermarking

Quantization has been used to embed the binary numbers [24, 25]. For simplicity, we only discuss one dimensional embedding primitive in this paper. As shown in Fig. 2, a binary number $m \in \{0, 1\}$ can be embedded by quantizing the embedding primitive X with the quantization step Δ. By assigning binary numbers to the quantization cells, X can be quantized to the nearest quantization cell corresponding to the number to be embedded so that the difference between the resulting primitive X' and X is minimized. If the value of X' is further modified to $\hat{X}$, the binary number corresponding to $\hat{X}$ may be different from m so that the modification can be detected. To make it hard to estimate the quantization step Δ from the resulting prim-

itive X', we adopt the following quantization method: To embed a binary number $m \in \{0, 1\}$ by quantizing the embedding primitive X, its corresponding integer quotient U and the remainder R should be calculated by

$$\begin{cases} U = \lfloor X/\Delta \rfloor \\ R = X\%\Delta \end{cases} . \tag{1}$$

And X is modified by

$$X' = \begin{cases} X & \text{if } U\%2 = m \\ X + 2 \times (\Delta - R) & \text{if } U\%2 \neq m \ \& \ R \geq \frac{\Delta}{2} \\ X - 2 \times R & \text{if } U\%2 \neq m \ \& \ R < \frac{\Delta}{2} \end{cases} \tag{2}$$

so that the embedded value can be extracted by $\lfloor X'/\Delta \rfloor \%2$. Under the circumstances, the error introduced by the modification in Eq.(2) will not exceed the quantization step Δ. Suppose that R is uniformly distributed within $[0, \Delta)$ and the chance of $m = 0$ is 0.5. Then $X'\%\Delta$ will also be uniformly distributed within $[0, \Delta)$ so that it is hard to estimate the value of Δ even a lot of resulting primitives are given.

Furthermore, to allow slight change of the resulting primitive X', a margin around the quantization grid is required. Hence, Eq.(2) is slightly deformed by adding a parameter $\epsilon \in (0, \frac{\Delta}{2})$ through

$$X' = \begin{cases} (U+1) \times \Delta - \epsilon & \text{if } U\%2 = m \ \& \ \Delta - \epsilon < R \\ X & \text{if } U\%2 = m \ \& \ \epsilon \leq R \leq \Delta - \epsilon \\ U \times \Delta + \epsilon & \text{if } U\%2 = m \ \& \ R < \epsilon \\ (U+1) \times \Delta + \epsilon & \text{if } U\%2 \neq m \ \& \ \Delta - \epsilon < R \\ X + 2 \times (\Delta - R) & \text{if } U\%2 \neq m \ \& \ \frac{\Delta}{2} \leq R \leq \Delta - \epsilon \\ X - 2 \times R & \text{if } U\%2 \neq m \ \& \ \epsilon \leq R < \frac{\Delta}{2} \\ U \times \Delta - \epsilon & \text{if } U\%2 \neq m \ \& \ R < \epsilon \end{cases} \tag{3}$$

so that the change of X' within $(-\epsilon, \epsilon)$ can be allowed without changing the embedded value. An appropriate value should be assigned to ϵ without disclosing the value of the quantization step Δ. If we choose the value of ϵ in proportional to the quantization step Δ, $\frac{\Delta}{6}$ for example, the allowed range can be adjusted by the value of Δ.

Obviously, the embedded watermark will be imperceptible and sensitive to outer processing if a small step is used in the quantization. However, only part of modification can be detected by quantizing the embedding primitives separately. Suppose that X' is changed by δx (so we can use $X' + \delta x$ to denote the modified primitive) and $\lfloor \frac{X'\%\Delta + \delta x}{\Delta} \rfloor \%2 = 1$. The retrieved value $\lfloor \frac{X'+\delta x}{\Delta} \rfloor \%2$ will be different from m, which is equal to $\lfloor \frac{X'}{\Delta} \rfloor \%2$. However, the retrieved value $\lfloor \frac{X'+\delta x}{\Delta} \rfloor \%2$ will be equal to m as long as $\lfloor \frac{X'\%\Delta + \delta x}{\Delta} \rfloor \%2 = 0$. As shown in the first line of Fig. 3, if the embedding primitive is modified to the cross, only those modifications that change the resulting primitive to the diagonal intervals can be detected by comparing $\lfloor \frac{X'+\delta x}{\Delta} \rfloor \%2$ with m.

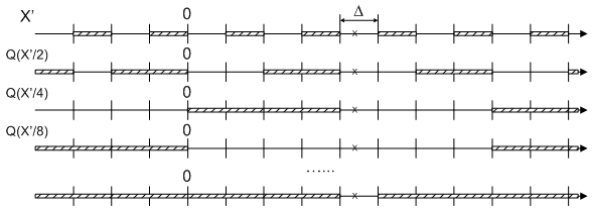


Figure 3. Suppose a binary number m is embedded by modifying the embedding primitive X to the cross with the quantization step Δ. Then only the modifications changing the resulting primitive X' to the diagonal intervals in the first line can be detected by comparing $\lfloor \frac{X'+\delta x}{\Delta} \rfloor \%2$ with m, where δx is the change of X'. When we quantize $X_1 + \lfloor \frac{X'}{2\Delta} \rfloor \Delta$, $X_2 + \lfloor \frac{X'}{4\Delta} \rfloor \Delta$, $X_3 + \lfloor \frac{X'}{8\Delta} \rfloor \Delta$ with Δ by modifying X_1, X_2 and X_3, the modifications changing X' to the diagonal intervals from the second to the fourth line can be detected, respectively, supposing the resulting primitives X'_1, X'_2 and X'_3 are unchanged. Furthermore, the modifications changing X' to the diagonal intervals in the last line can be detected by quantizing $X_i + \lfloor \frac{X'}{2^i \Delta} \rfloor \Delta$ to embed a binary number m_i by solely modifying X_i for $i = 1, 2, 3, \ldots$, if the set of resulting primitives $\{X'_1, X'_2, X'_3, \ldots, \}$ are unchanged.

In addition, one may define the embedding primitives correlated with each other to detect illegal modifications. In that case, several resulting primitives will be simultaneously changed when a single modification is made. As in [13], where the distance from the centroid of a surface polygon to the mesh centroid is defined as the embedding primitive, if one vertex position is modified, the distances from all the polygons containing the modified vertex to the mesh centroid will be affected. By making multiple embedding primitives correlated with each other, resistance to the class of substitution attacks, including cut-and-paste attack, VQ attack [26] and collage attack [27], can also be achieved. Normally, the inherent properties of media data should be exploited to define such an embedding primitive, such as the connectivity and positions of vertices in [12, 13, 14]. In the following, a new quantization scheme independent from the properties of media data will be proposed to improve the fragility of the embedded data. Therefore, it is applicable to any quantization-based fragile watermarking algorithm, regardless whether the embedding primitives are correlated or not.

2.2 A New Quantization Scheme

Besides making the embedding primitives correlated with each other, we want to improve the quantization scheme to detect illegal modification more efficiently. One way is to add a resulting primitive, or its representation, to the current embedding primitive to generate a new element, which can be quantized to embed a binary number by solely modifying the current embedding primitive. By this means, the changes of both the current and previous resulting primitives will affect the embedded value. Suppose that a binary number m is embedded by quantizing the embedding primitive X with the quantization step Δ, and the resulting primitive X' is further changed by δ_x. The change can be detected by comparing $\lfloor\frac{X'+\delta_x}{\Delta}\rfloor\%2$ with m only when $\lfloor\frac{X'+\delta_x}{\Delta}\rfloor\%2 \neq \lfloor\frac{X'}{\Delta}\rfloor\%2$. To represent the quantization result, we use $Q(x)$ to denote $\lfloor\frac{x}{\Delta}\rfloor \times \Delta$. If we add $Q(\frac{X'}{2})$ to X_1, and quantize $X_1 + Q(\frac{X'}{2})$ to embed a binary number m_1 by modifying X_1, illegal modifications to X' may be detected by comparing $\lfloor\frac{X'_1+Q(\frac{X'+\delta_x}{2})}{\Delta}\rfloor\%2$ with m_1. If the resulting primitive X'_1 is not changed, δ_x will be detected if $\lfloor\frac{X'+\delta_x}{2\Delta}\rfloor\%2 \neq \lfloor\frac{X'}{2\Delta}\rfloor\%2$. Furthermore, after we quantize $X_2 + Q(\frac{X'}{4})$ by modifying X_2 to embed m_2, δ_x will be detected by comparing $\lfloor\frac{X'_2+Q(\frac{X'+\delta_x}{4})}{\Delta}\rfloor\%2$ with m_2 if $\lfloor\frac{X'+\delta_x}{4\Delta}\rfloor\%2 \neq \lfloor\frac{X'}{4\Delta}\rfloor\%2$ and the resulting primitive X'_2 is not changed. The chance to detect δ_x will be further increased if we quantize $X_3 + Q(\frac{X'}{8})$ and $X_4 + Q(\frac{X'}{16})$ to embed m_3 and m_4, respectively. As shown in Fig. 3, if we quantize $X_1+Q(\frac{X'}{2})$, $X_2+Q(\frac{X'}{4})$, $X_3+Q(\frac{X'}{8})$ by solely modifying X_1, X_2 and X_3, those modifications that change X' to the diagonal intervals from the second to the fourth line can be respectively detected, supposing the resulting primitives X'_1, X'_2 and X'_3 are unchanged. Consequently, after we quantize $X_i+Q(\frac{X'}{2^i})$ to embed a binary number m_i by solely modifying X_i for $i = 1, 2, 3, \ldots$, the change of X' outside the quantization cell where it is located will be detected if the set of resulting primitives $\{X'_1, X'_2, X'_3, \ldots,\}$ are unchanged. As shown in the last line of Fig. 3, the modification made to X' will be detected if δ_x exceeds Δ.

To achieve the property in the last line of Fig. 3, a new quantization scheme is proposed as follows: Given a set of embedding primitives $\{X_1, X_2, \ldots, X_N\}$ that can represent authenticity and integrity of media content, a set of elements $\{Y_1, Y_2, \ldots, Y_N\}$ can be generated from them by

$$\begin{cases} Y_1 = X_1 \\ Y_2 = X_2 + Q(\frac{X'_1}{2}) \\ Y_3 = X_3 + Q(\frac{X'_2}{2}) + Q(\frac{X'_1}{4}) \\ \cdots \\ Y_N = X_N + Q(\frac{X'_{N-1}}{2}) + Q(\frac{X'_{N-2}}{4}) + \ldots + Q(\frac{X'_1}{2^{N-1}}) \end{cases}, \quad (4)$$

where X'_i with $i \in \{1, \ldots, N\}$ is the resulting value after quantizing Y_i with Eq.(1) and Eq.(3) by solely modifying X_i to embed a binary number m_i. Based on Eq.(4), we can obtain the set of quantized elements $\{Y'_1, Y'_2, \ldots, Y'_N\}$ by

$$\begin{cases} Y'_1 = X'_1 \\ Y'_2 = X'_2 + Q(\frac{X'_1}{2}) \\ Y'_3 = X'_3 + Q(\frac{X'_2}{2}) + Q(\frac{X'_1}{4}) \\ \cdots \\ Y'_N = X'_N + Q(\frac{X'_{N-1}}{2}) + Q(\frac{X'_{N-2}}{4}) + \ldots + Q(\frac{X'_1}{2^{N-1}}) \end{cases}, \quad (5)$$

whereas the embedded value m_i can be retrieved by $\lfloor\frac{Y'_i}{\Delta}\rfloor\%2$. Therefore, the binary number m_i embedded by quantizing Y_i is associated with the resulting primitives X'_j with $1 \leq j \leq i$. In this way, the change of X'_i will probably alter the quantized elements Y'_k with $i \leq k \leq N$. If we denote the modified value of Y'_k as $\hat{Y}_k$, the change of X'_i may be detected by comparing $\lfloor\frac{\hat{Y}_k}{\Delta}\rfloor\%2$ with m_k for $i \leq k \leq N$. One may argue that the change δ_N made to X'_N can only be detected when $\lfloor\frac{X'_N+\delta_N}{\Delta}\rfloor\%2 \neq \lfloor\frac{X'_N}{\Delta}\rfloor\%2$. Actually, we can use the ratio R_a between Δ and X'_Δ, which is defined as $X'_N + \frac{X'_{N-1}}{2} + \frac{X'_{N-2}}{4} + \ldots + \frac{X'_1}{2^{N-1}}$, instead of Δ itself as the input of authentication process. In the authentication process, the value of R_a will be used to calculate the quantization step Δ from X'_Δ to retrieve the embedded data. The change of X'_N will eventually change the value of X'_Δ, as well as the quantization step Δ so that the retrieved data will be different from the original one. Therefore, the modifications to the last ones among the set of resulting primitives $\{X'_1, X'_2, \ldots, X'_N\}$ can be easily detected. To analyze in which condition that the modifications to the resulting primitives can be detected, we have the following theorem.

Theorem 1 *Suppose that there are k resulting primitives among $\{X'_1, X'_2, \ldots, X'_N\}$ having been modified with the changes $\{\delta_1, \delta_2, \ldots, \delta_N\}$, respectively, and denoted as $\{X'_{t_1}, X'_{t_2}, \ldots, X'_{t_k}\}$ with the sequence numbers $\{t_1, t_2, \ldots, t_k\} \subseteq \{1, 2, \ldots, N\}$. The modifications will be detected by the proposed quantization scheme if there exists an integer $i \in \{1, 2, \ldots, k\}$ so that*

$$\begin{aligned} &(\lfloor\frac{X'_{t_i}+\delta_i}{2^n\Delta}\rfloor + \lfloor\frac{X'_{t_{(i-1)}}+\delta_{(i-1)}}{2^{n+t_i-t_{(i-1)}}\Delta}\rfloor + \ldots + \lfloor\frac{X'_{t_1}+\delta_1}{2^{n+t_i-t_1}\Delta}\rfloor)\%2 \neq \\ &(\lfloor\frac{X'_{t_i}}{2^n\Delta}\rfloor + \lfloor\frac{X'_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta}\rfloor + \ldots + \lfloor\frac{X'_{t_1}}{2^{n+t_i-t_1}\Delta}\rfloor)\%2 \end{aligned}, \quad (6)$$

for any an integer $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$, where $t_{i+1} = N + 1$ if $i = k$.

Proof: Suppose there exists an integer $i \in \{1, 2, \ldots, k\}$ so that Eq.(6) holds for an integer $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$, where $t_{k+1} = N + 1$. Without loss of generalization, we take the case of $n = 0$. If we denote the resulting primitive that has not been changed by the modifications as X'_{u_j} with $\{u_1, u_2, \ldots, u_{(t_i-i)}\} \cup \{t_1, t_2, \ldots, t_i\} =$

$\{1, 2, \ldots, t_i\}$, then the retrieved value $\lfloor \frac{\hat{Y_{t_i}}}{\Delta} \rfloor \% 2$ can be represented by $(\lfloor \frac{X'_{t_i}+\delta_i}{\Delta} \rfloor + \lfloor \frac{X'^{+\delta_{(i-1)}}_{t_{(i-1)}}}{2^{t_i-t_{(i-1)}}\Delta} \rfloor + \cdots + \lfloor \frac{X'_{t_1}+\delta_1}{2^{t_i-t_1}\Delta} \rfloor + \sum_{j=1}^{t_i-i} \lfloor \frac{X'_{u_j}}{2^{t_i-u_j}\Delta} \rfloor) \% 2$, which is different from the embedded value m_{t_i}, i.e. $(\lfloor \frac{X'_{t_i}}{\Delta} \rfloor + \lfloor \frac{X'_{t_{(i-1)}}}{2^{t_i-t_{(i-1)}}\Delta} \rfloor + \ldots + \lfloor \frac{X'_{t_1}}{2^{t_i-t_1}\Delta} \rfloor + \sum_{j=1}^{t_i-i} \lfloor \frac{X'_{u_j}}{2^{t_i-u_j}\Delta} \rfloor) \% 2$, so that the modifications can be detected.

If there is only one resulting primitive X'_i among $\{X'_1, X'_2, \ldots, X'_N\}$ having been modified by δ_i, the condition in Theorem 1 can be simplified to $\lfloor \frac{X'_i+\delta_i}{2^n\Delta} \rfloor \% 2 \neq \lfloor \frac{X'_i}{2^n\Delta} \rfloor \% 2$ for any an $n \in \{0, 1, \ldots, N-i\}$. If i is close to N, δ_i will lead X'_Δ, as well as the quantization step Δ generated in the authentication process, to vary so that the modification can be easily detected. In the case that δ_i has little effect on Δ, it can still be detected by comparing $\lfloor \frac{\hat{Y_k}}{\Delta} \rfloor \% 2$ with m_k for $k \in \{i, i+1, \ldots, N\}$, respectively. Therefore, only the change within the quantization cell where the modified primitive is located can be allowed without altering the embedded values, as shown in the last line of Fig. 3. Otherwise, one or more of the retrieved values will be different from the embedded ones so that the modification is detected.

In the case that there are multiple resulting primitives having been modified as in Theorem 1, the modifications are possibly undetectable if for all $i \in \{1, 2, \ldots, k\}$,

$$\begin{aligned}(\lfloor \frac{X'_{t_i}+\delta_i}{2^n\Delta} \rfloor + \lfloor \frac{X'^{+\delta_{(i-1)}}_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \rfloor + \ldots + \lfloor \frac{X'_{t_1}+\delta_1}{2^{n+t_i-t_1}\Delta} \rfloor) \% 2 = \\ (\lfloor \frac{X'_{t_i}}{2^n\Delta} \rfloor + \lfloor \frac{X'_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \rfloor + \ldots + \lfloor \frac{X'_{t_1}}{2^{n+t_i-t_1}\Delta} \rfloor) \% 2\end{aligned}, \tag{7}$$

for all $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$, where $t_{k+1} = N+1$. If the embedding primitives are quantized separately without applying the proposed scheme, the modifications are undetectable if for all $i \in \{1, 2, \ldots, k\}$,

$$\lfloor \frac{X'_{t_i} + \delta_i}{\Delta} \rfloor \% 2 = \lfloor \frac{X'_{t_i}}{\Delta} \rfloor \% 2. \tag{8}$$

Comparing Eq.(7) with Eq.(8), it can be seen that the chance to detect illegal modifications has been increased by applying the proposed scheme. In Eq.(8), the change δ_i in $[2b\Delta - X'_{t_i}\%\Delta, (2b+1)\Delta - X'_{t_i}\%\Delta)$ for any an integer b is undetectable. Whereas in Eq.(7), only the change δ_i satisfies that $(\lfloor \frac{X'_{t_i}+\delta_i}{2^n\Delta} \rfloor - \lfloor \frac{X'_{t_i}}{2^n\Delta} \rfloor) \% 2 = (\lfloor \frac{X'^{+\delta_{(i-1)}}_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \rfloor - \lfloor \frac{X'_{t_{(i-1)}}}{2^{n+t_i-t_{(i-1)}}\Delta} \rfloor + \ldots + \lfloor \frac{X'_{t_1}+\delta_1}{2^{n+t_i-t_1}\Delta} \rfloor - \lfloor \frac{X'_{t_1}}{2^{n+t_i-t_1}\Delta} \rfloor) \% 2$ for all $n \in \{0, 1, \ldots, t_{i+1} - t_i - 1\}$ is undetectable, where $t_{k+1} = N+1$. Obviously, the condition in Eq.(7) is stronger than that in Eq.(8) because each value within $\{0, 1, \ldots, t_{i+1} - t_i - 1\}$ ($t_{k+1} = N+1$) will be assigned to the parameter n in Eq.(7), respectively. To improve the security, the order of the embedding primitives $\{X_1, X_2, \ldots, X_N\}$ can be scrambled by using a secret key K as the seed of pseudo-random number generator. Without the correct order of the embedding primitives, it is even harder to make the illegal modifications to the resulting primitives undetectable as the strict requirement on the changes of resulting primitives is given in Eq.(7). To analyze if slight changes of the resulting primitives can be allowed without changing the embedded data, we have the following theorem.

Theorem 2 *Suppose that the change of each resulting primitive X'_i caused by the modification is within $(-S, S)$. Then the data embedded by the proposed quantization scheme will remain the same if*

$$S < \frac{\epsilon}{1 + \frac{2X'_M}{X'_\Delta}}, \tag{9}$$

where the parameter ϵ is given in Eq.(3), X'_M is the greatest one among the set of resulting primitives $\{X'_1, X'_2, \ldots, X'_N\}$ and $X'_\Delta = X'_N + \frac{X'_{N-1}}{2} + \frac{X'_{N-2}}{4} + \ldots + \frac{X'_1}{2^{N-1}}$.

Proof: If we denote the change of X'_i caused by the modification as δ_i, then $\delta_i \in (-S, S)$ and the change of X'_Δ, i.e. $X'_N + \frac{X'_{N-1}}{2} + \frac{X'_{N-2}}{4} + \ldots + \frac{X'_1}{2^{N-1}}$, will be within $(-2S, 2S)$. Suppose δ_d is the error introduced to the quantization step Δ, which is generated by $\frac{X'_\Delta}{R_a}$ in the authentication process. Since $R_a = \frac{X'_\Delta}{\Delta}$, it can be seen $\delta_d \in (-\frac{2S\Delta}{X'_\Delta}, \frac{2S\Delta}{X'_\Delta})$ if the value of R_a is correctly provided. After the modification, $\lfloor \frac{X'_i}{\Delta} \rfloor$ becomes $\lfloor \frac{X'_i+\delta_i}{\Delta+\delta_d} \rfloor$ and the two values will be the same if $X'_i\%\Delta + \delta_i - \lfloor \frac{X'_i}{\Delta} \rfloor \times \delta_d \in (0, \Delta)$, i.e. $\epsilon > S + \lfloor \frac{X'_i}{\Delta} \rfloor \times \frac{2S\Delta}{X'_\Delta}$, as $X'_i\%\Delta \in (\epsilon, \Delta - \epsilon)$, $\delta_i \in (-S, S)$ and $\delta_d \in (-\frac{2S\Delta}{X'_\Delta}, \frac{2S\Delta}{X'_\Delta})$. Because $X'_i \geq \lfloor \frac{X'_i}{\Delta} \rfloor \times \Delta$, $\lfloor \frac{X'_i}{\Delta} \rfloor$ will be identical to $\lfloor \frac{X'_i+\delta_i}{\Delta+\delta_d} \rfloor$ for $i \in \{1, 2, \ldots, N\}$ if $\epsilon > S + \frac{2SX'_M}{X'_\Delta} \geq S + \lfloor \frac{X'_i}{\Delta} \rfloor \times \frac{2S\Delta}{X'_\Delta}$, where X'_M is the greatest one among $\{X'_1, X'_2, \ldots, X'_N\}$ as in Eq.(9). Actually, the value of $\lfloor \frac{X'_i}{2^n\Delta} \rfloor$ will not either be changed by the modification given Eq.(9). Since $\lfloor \frac{X'_i}{2^n\Delta} \rfloor$ becomes $\lfloor \frac{X'_i+\delta_i}{2^n(\Delta+\delta_d)} \rfloor$ after the modification, the two values will be identical to each other if $X'_i\%\Delta + \delta_i - \lfloor \frac{X'_i}{2^n\Delta} \rfloor \times 2^n \times \delta_d \in (0, \Delta)$, i.e. $\epsilon > S + \lfloor \frac{X'_i}{2^n\Delta} \rfloor \times 2^n \times \frac{2S\Delta}{X'_\Delta}$. Given Eq.(9), it can be seen that $\epsilon > S + S \times \frac{2X'_M}{X'_\Delta} = S + \frac{X'_M}{2^n\Delta} \times 2^n \times \frac{2S\Delta}{X'_\Delta}$. Because $\frac{X'_M}{2^n\Delta} \geq \lfloor \frac{X'_i}{2^n\Delta} \rfloor$, we have $\epsilon > S + \lfloor \frac{X'_i}{2^n\Delta} \rfloor \times 2^n \times \frac{2S\Delta}{X'_\Delta}$ so that $\lfloor \frac{X'_i}{2^n\Delta} \rfloor = \lfloor \frac{X'_i+\delta_i}{2^n(\Delta+\delta_d)} \rfloor$. From Eq.(5), it can be seen that the change of Y'_i caused by the modification will be $\delta_i + \delta_d \times (\lfloor \frac{X'_{i-1}}{2\Delta} \rfloor + \ldots + \lfloor \frac{X'_1}{2^{i-1}\Delta} \rfloor)$ given Eq.(9). Because

$\lfloor \frac{X'_{i-1}}{2\Delta} \rfloor + \ldots + \lfloor \frac{X'_1}{2^{i-1}\Delta} \rfloor \leq \frac{X'_M}{2\Delta} + \ldots + \frac{X'_M}{2^{i-1}\Delta} < \frac{X'_M}{\Delta}$, the change of Y'_i will be distributed in $(-S - \frac{2S\Delta}{X'_\Delta} \times \frac{X'_M}{\Delta}, S + \frac{2S\Delta}{X'_\Delta} \times \frac{X'_M}{\Delta})$, i.e. $(-S - \frac{2SX'_M}{X'_\Delta}, S + \frac{2SX'_M}{X'_\Delta})$. Since $Y'_i\%\Delta = X'_i\%\Delta \in (\epsilon, \Delta - \epsilon)$, the embedded data will not be changed by the modification if $\epsilon > S + \frac{2SX'_M}{X'_\Delta}$, as indicated by Eq.(9).

If each of the embedding primitives $\{X_1, X_2, \ldots, X_N\}$ is quantized separately, the embedded value will remain the same after the modification if $S < \epsilon$. Compared with Eq.(9), it can be seen the allowed range of each resulting primitive X'_i has been reduced by applying the proposed quantization scheme. Nevertheless, the change of X'_i within $\{-\frac{\epsilon}{1+\frac{2X'_M}{X'_\Delta}}, \frac{\epsilon}{1+\frac{2X'_M}{X'_\Delta}}\}$ is still allowed without changing the embedded data, where the values of X'_M and X'_Δ are obtained from the set of resulting primitives $\{X'_1, X'_2, \ldots, X'_N\}$.

3 A New Watermarking Algorithm for Mesh Authentication

In the preceding section, a new quantization scheme that is applicable to any fragile watermarking algorithm based on quantization has been proposed. To test the efficacy of the proposed scheme, we will apply it to the algorithm in [14] so that a new fragile watermarking algorithm will be presented for mesh authentication. The distance from a vertex to the centroid of its traversed neighbors is chosen as the embedding primitive so that high capacity can be achieved. In the embedding process, a watermark is embedded by slightly modifying the defined distances in a sequential way as in Eq.(4). In the authentication process, the embedded watermark is retrieved from the watermarked mesh and compared with the original one to obtain the authentication result.

3.1 Watermark Embedding

There are two parts of information contained in polygonal meshes, i.e. geometrical and topological. The mesh geometry can be represented by the set of vertex positions $V = \{v_1, \ldots, v_l\}$, which defines the shape of the mesh in R^3 given l vertices in a mesh. The mesh topology, i.e. the connectivity between vertices, specifies the n vertices $\{v_k^1, \cdots, v_k^n\}$ in the k-th polygon, as described by IndexedFaceSet in VRML [28] format. Given a string of binary numbers $W = (w_i)_{i=1}^B$ of the length B, the task of watermark embedding is to hide the value of each w_i by slightly modifying the embedding primitive, i.e. the distance from a vertex to the centroid of its traversed neighbors. A secret key K is used as the seed of pseudo-random number generator to scramble the vertex indices I and face indices F, respectively, whereas the scrambled vertex indices I' and face indices F' are used to order the process of mesh traversal.

Let us suppose N_i neighboring vertices of a newly traversed vertex v_i have been traversed and denoted as $(v_i^j)_{j=1}^{N_i}$. Then the centroid of its traversed neighbors can be calculated by

$$v_{ic} = \frac{1}{N_i} \sum_{j=1}^{N_i} v_i^j. \tag{10}$$

The distance from v_{ic} to v_i is chosen as the embedding primitive X_i in Eq.(4), as defined by

$$X_i = \sqrt{(v_{icx} - v_{ix})^2 + (v_{icy} - v_{iy})^2 + (v_{icz} - v_{iz})^2}, \tag{11}$$

where $\{v_{icx}, v_{icy}, v_{icz}\}$ and $\{v_{ix}, v_{iy}, v_{iz}\}$ are the coordinates of v_{ic} and v_i in R^3, respectively. By choosing the distance from a vertex to the centroid of all its neighbors as the embedding primitive, high capacity can be achieved. If we choose the distance from a vertex to the centroid of all its neighbors as the embedding primitive and modify the distance to embed a binary number by adjusting the vertex position, the positions of the adjusted vertex and its neighboring vertices cannot be changed any more after the adjustment to preserve the embedded value. The capacity will drop because most of vertex positions cannot be used to embed binary numbers. Hence, only the traversed vertices among each vertex's neighbors are chosen to generate X_i so that all vertex positions except the first traversed one can be adjusted. Consequently, the element Y_i can be generated as in Eq.(4) and quantized by solely modifying X_i via adjusting the vertex position v_i.

The process of mesh traversal is ordered by the scrambled vertex indices I' and face indices F' as follows: Among those vertices in the polygon lastly indexed by F', the one first indexed by I' is traversed at first. Among the neighbors of the traversed vertices, the one first indexed by I' will always be subsequently traversed. Given l vertices in a manifold polygonal mesh, in which an edge belongs to exactly two polygons, there are $l - 1$ embedding primitives because only the first traversed vertex has no traversed neighbor. In [14], each of the embedding primitive $\{X_1, X_2, \ldots, X_{l-1}\}$ is quantized separately to embed a watermark. To implement the proposed quantization scheme, a set of the corresponding elements $\{Y_1, Y_2, \ldots, Y_{l-1}\}$ is generated as shown in Eq.(4). Each Y_i with $i \in \{1, 2, \ldots, l-1\}$ is quantized with the same quantization step Δ to embed a binary number w_i by solely modifying X_i. Consequently, the resulting primitive X'_i is used to adjust the vertex position v_i by

$$v'_i = v_{ic} + (v_i - v_{ic}) \times \frac{X'_i}{X_i}, \tag{12}$$

where v'_i is the adjusted vertex position. The watermarked mesh is generated after the position of the last traversed vertex is adjusted. After that, the ratio R_a between X'_Δ, which is defined as $X'_{l-1} + \frac{X'_{l-2}}{2} + \ldots + \frac{X'_1}{2^{l-2}}$, and Δ is calculated by

$$R_a = \frac{X'_{l-1} + \frac{X'_{l-2}}{2} + \ldots + \frac{X'_1}{2^{l-2}}}{\Delta}. \quad (13)$$

The obtained value of R_a will be used to calculate the quantization step Δ in the authentication process.

3.2 The Authentication Process

To retrieve the embedded watermark from the watermarked mesh, the quantization step Δ used in the embedding process is required. To obtain Δ, the values of the parameters R_a and X'_Δ, which can be generated from the set of resulting primitives $\{X'_1, X'_2, \ldots, X'_{l-1}\}$, should be provided. Since the mesh traversal is ordered by the scrambled vertex indices I' and face indices F', the secret key K is needed to generate them from the original vertex indices I and face indices F. Therefore, the secret key K and the parameter R_a are required in the authentication process.

The process of watermark retrieval is as follows: Initially, the vertex indices I and face indices F in the watermarked mesh are scrambled by using K as the seed of pseudo-random number generator to generate I' and F', respectively. During the process of mesh traversal, the distance from a vertex to the centroid of its traversed neighbors can be calculated by using Eq.(10) and Eq.(11), as in watermark embedding process. If the watermarked mesh is intact, the obtained distances are the resulting primitives generated in the embedding process, i.e. $\{X'_1, X'_2, \ldots, X'_{l-1}\}$. With the parameter R_a, the quantization step Δ is calculated by

$$\Delta = \frac{X'_{l-1} + \frac{X'_{l-2}}{2} + \ldots + \frac{X'_1}{2^{l-2}}}{R_a}. \quad (14)$$

The set of quantized elements, $\{Y'_1, Y'_2, \ldots, Y'_{l-1}\}$, can also be generated from $\{X'_1, X'_2, \ldots, X'_{l-1}\}$ as in Eq.(5). With the obtained quantization step Δ, the bit value w'_i is retrieved by

$$w'_i = \lfloor Y'_i/\Delta \rfloor \% 2. \quad (15)$$

The whole message string $W' = (w'_i)_{i=1}^{l-1}$ will be retrieved after the last bit is extracted from Y'_{l-1}.

To detect the tampering made to the watermarked mesh, and estimate its strength if any, the retrieved bit string $W' = (w'_i)_{i=1}^{l-1}$ is compared with the original one $W = (w_i)_{i=1}^{l-1}$ by defining a numerical value NC as follows:

$$NC = \frac{1}{l-1} \sum_{i=1}^{l-1} I(w'_i, w_i), \quad (16)$$

with

$$I(w'_i, w_i) = \begin{cases} 1 & \text{if } w'_i = w_i \\ 0 & \text{otherwise} \end{cases}. \quad (17)$$

The value NC is expected to be less than 1 if the mesh content has been processed by illegal modifications, as discussed in the following subsection. So the mesh content is considered as intact if $NC = 1$.

3.3 The Properties of The Embedded Watermark

Since the ratio between any two resulting primitives X'_i and X'_j in the watermarked mesh remains the same after the mesh is rotated, uniformly scaled or translated, the ratio between X'_i and X'_Δ, the linear combination of $\{X'_1, X'_2, \ldots, X'_{l-1}\}$, is invariant to RST transformations. As the quantization step Δ used in the authentication process is proportional to X'_Δ, the ratio between X'_i and Δ is also invariant to RST transformations. Hence, the embedded value w_i, which is identical to $\lfloor \frac{Y'_i}{\Delta} \rfloor \% 2 = (\lfloor \frac{X'_{l-1}}{\Delta} \rfloor + \lfloor \frac{X'_{l-2}}{2\Delta} \rfloor + \ldots + \lfloor \frac{X'_1}{2^{l-2}\Delta} \rfloor) \% 2$, is robust against RST transformations. In the case of topological modifications, the neighboring information of a vertex will be changed so that the mesh traversal will be different from that in the embedding process. As a result, the resulting primitives cannot be correctly obtained in the authentication process. Therefore, the embedded watermark is sensitive to the topological modifications.

As for mantissa truncation of vertex coordinate, which is stored as a single-precision floating-point number, let us suppose the truncation error is distributed within $(-T, T)$. Then the error introduced to the centroid of a vertex's traversed neighbors as in Eq.(10) is also distributed within $(-T, T)$. The error introduced to the resulting primitive X'_i in Eq.(11) will be distributed within $(-2\sqrt{3}T, 2\sqrt{3}T)$ so that the parameter S in Theorem 2 will be $2\sqrt{3}T$. Therefore, mantissa truncation of vertex coordinates will be allowed without changing the embedded watermark if

$$T < \frac{\epsilon}{2\sqrt{3}(1 + \frac{2X'_M}{X'_\Delta})}, \quad (18)$$

where X'_M is the greatest one among the set of resulting primitives $\{X'_1, X'_2, \ldots, X'_{l-1}\}$ and $X'_\Delta = X'_{l-1} + \frac{X'_{l-2}}{2} + \ldots + \frac{X'_1}{2^{l-2}}$.

For the geometrical modifications that take place on part of vertices, we take for instance the case that one vertex has been modified. The distance X'_i from the modified vertex to its traversed neighbors may be changed, denoted as $X'_i + \delta_v$. If the modification has little affect on the quantization step Δ obtained from Eq.(14) and $|\delta_v| < \epsilon$, the embedded value will remain the same because $Y'_i \% \Delta = X'_i \% \Delta \geq \epsilon$, whereas the change will also be detected if $|\delta_v| > \Delta$. For an

Table 1. THE MESH MODELS IN THE EXPERIMENTS

Model	Meshes	Vertices	Polygons	Capacity(bits)
fish	1	742	1408	741
teapot	5	1631	3080	1626
dog	48	7616	13176	7568
wolf	90	8176	13992	8086
horse	31	10316	18359	10285

untraversed neighboring vertex of the modified one, it will be traversed later than the modified vertex so that the distance from it to the centroid of its traversed neighbors will also be changed. As a result, the chance to detect the modification is increased. In summary, if a vertex is modified outside the allowed range, the data embedded by adjusting the positions of itself and its untraversed neighbors will be altered by the modification.

4 Experimental Results

The fragile watermarking algorithm in the preceding section was performed on the mesh models listed in Table I, where the capacity is also given. Since the precision of mesh data is limited, we can denote the precision interval of vertex coordinate as $\{-T, T\}$. Then the quantization step Δ should be chosen satisfying Eq.(18), where the parameter ϵ was assigned $\frac{\Delta}{6}$ to allow slight change of the resulting primitives. The implementation of the proposed quantization scheme increased the complexity of the fragile watermarking algorithm, as the runtime of embedding 1626 bits into the mesh model "teapot" using the new algorithm was 1.406 seconds in a 2.66G Pentium 4 PC with 512MB RAM, greater than 0.750 seconds of the former algorithm in [14].

4.1 Distortion of the Original Mesh

In the experiments, an appropriate quantization step Δ should be chosen to achieve the imperceptibility of the embedded watermark. It can be seen from Eq.(3) that the change of each embedding primitive X_i is within $(-\Delta, \Delta)$. As shown in Eq.(4), the impact of watermark embedding on mesh content is not aggravated by the proposed quantization scheme. From Eq.(12), we can further conclude that the adjustment of each vertex is bounded in the sphere with its original position as the centroid and Δ as the radius. Upon the fact that the mesh topology has not been changed, the distance from the adjusted vertex to its former position is used to represent the distortion of the mesh content. The impact of watermark embedding could be tuned by the quantization step Δ used in the experiments. If 0.01

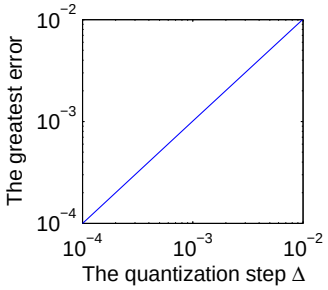


Figure 4. The greatest error increases with the quantization step.

was assigned to Δ, the greatest error (i.e. the greatest distance among all the adjusted vertices) never exceeded 0.01, whereas the greatest error was below 0.001 if 0.001 had been assigned to Δ instead, as shown in Fig. 4. The pictures rendered from the mesh models "teapot" and "horse" before and after the embedding process are shown in Fig. 5.

4.2 Capacity

The capacity of the new fragile watermarking algorithm is identical to that of the former algorithm in [14], as one bit value is embedded by modifying each embedding primitive X_i in Eq.(4). So it can be seen the proposed quantization scheme does not affect the capacity of fragile watermarking algorithms. As in [14], the new watermarking algorithm is applicable for any manifold polygonal mesh, in which an edge belongs to exactly two polygons. Given l vertices in the original mesh, the capacity will be $l - 1$ bits, which is tending to 1 bit/vertex and higher than the previously reported works [12, 13]. If a mesh model consists of s separate meshes as in Table I, the capacity will be $l - s$ bits since the position of the first traversed vertex in each mesh has not been adjusted.

4.3 Security

As the former algorithm in [14], the security of the new fragile watermarking algorithm relies on the secrecies of the key K and the parameter R_a. The secret key K is used to scramble the vertex indices I and face indices F to generate the scrambled indices I' and F', which determine the order of the embedding primitives $\{X_1, X_2, \ldots, X_{l-1}\}$, whereas the parameter R_a is used to calculate the quantization step Δ in the authentication process. Given there are l vertices and p polygons in a mesh model, the permutation of the vertex indices is $l!$. Without the secret key K, the mesh traversal must be performed at least $pl!$ times to guarantee the embedded watermark can be correctly retrieved, given the accurate quantization step Δ. To make it hard to estimate

Table 2. In the experiments, $1/10,000$ of the greatest distance from a vertex to the mesh centroid is chosen as the quantization step Δ, and $\frac{\Delta}{6}$ is assigned to the parameter ϵ. The NC values are calculated by comparing the retrieved watermarks and the original watermark after the watermarked meshes have been processed by the following manipulations, respectively.

Meshes	RST transformations	Modifying one vertex position	Moving two vertices oppositely	Reducing one face	Truncating seven LSBs	Truncating eight LSBs	Extracting without the correct key
fish	1.0000	0.9959	0.9892	0.7584	1.0000	0.9946	0.8704
teapot	1.0000	0.9969	0.9963	0.6014	1.0000	0.9766	0.6875
dog	1.0000	0.9996	0.9993	0.7009	1.0000	1.0000	0.7448
wolf	1.0000	0.9993	0.9992	0.5186	1.0000	1.0000	0.5179
horse	1.0000	0.9998	0.9995	0.9120	1.0000	0.9993	0.5437

the quantization step Δ from the set of resulting primitives $\{X'_1, X'_2, \ldots, X'_{l-1}\}$, the parameter ϵ used in Eq.(3) should be assigned with a relatively small value, $\frac{\Delta}{6}$ for instance. Moreover, the embedding primitive X_i is defined over the neighborhood of a vertex so that resistance to substitution attacks is achieved, which makes it even harder to construct a counterfeit mesh with the same watermark.

4.4 Fragility

What we really improved in the new algorithm is the fragility of the embedded watermark, which is critical to detect illegal modifications to the watermarked mesh. In the experiments, 1/10,000 of the greatest distance from a vertex to the mesh centroid, as denoted by D_m in the following, was chosen as the quantization step Δ, whereas $\frac{\Delta}{6}$ was assigned to the parameter ϵ. To test the fragility of the embedded watermark, the watermarked mesh models went through RST transformations, modifying one vertex position by adding the vector $\{3\Delta, 3\Delta, 3\Delta\}$, changing the positions of two vertices oppositely (respectively by adding the vectors $\{2\Delta, 2\Delta, 2\Delta\}$ and $\{-2\Delta, -2\Delta, -2\Delta\}$), reducing one face from the mesh, and truncating the least significant bits (LSB) of each vertex coordinate, respectively. The watermarks retrieved from the processed mesh models were compared with the original one by using Eq.(16). The obtained NC values are listed in Table II.

From the experimental results, it can be seen that those content-preserving manipulations, such as a certain degree of coordinate truncation and RST transformations, could be allowed although the proposed quantization scheme was applied. Compared with the NC values obtained in the former algorithm in [14], it can be seen the embedded watermark was more sensitive to illegal manipulations. It is worth to be noted that the allowed range of coordinate truncation could be adjusted with the quantization step Δ. If $1/10,000$ of D_m was assigned to Δ, truncating of 7 least significant bits (LSB) of vertex coordinate was allowed for the "teapot" model. Whereas $1/100,000$ of D_m was assigned instead, only 4 LSBs of vertex coordinate could be truncated without changing the embedded watermark.

4.5 Tamper Localization

As for block-wise fragile algorithms based on quantization, the property of tamper localization is not affected since the proposed scheme can be applied to each block, respectively. If the resulting primitives are tampered, the blocks containing the modified primitives can be found out. Otherwise, if the fragile watermarking algorithm is not block-wise, such as the algorithm in Section III, the property of tamper localization may be sacrificed to improve the fragility of the embedded watermark. By using the former algorithm in [14], those geometrical modifications that have little impact on the quantization step can be localized by comparing the retrieved bit values with the original ones. For a vertex where the two values do not match with each other, its position or those of its previously traversed neighboring vertices might have been changed. After applying the proposed scheme, for a vertex where the two values do not match, the positions of itself and all the previously traversed vertices might have been modified so that the modification cannot be localized. In our experiments, the watermarked mesh model "teapot" in Fig. 5(b) was tampered by modifying only one vertex on its handle and the tampered mesh model is shown in Fig. 6(a). The tampering is detected by comparing the extracted watermark with the original one. However, those vertices where the original and retrieved bit values did not match with each other were more than the modified vertex and its previously traversed neighbors, as shown in Fig. 6(b).

(a) The original mesh model "teapot"

(b) The "teapot" model with 1626 bits embedded

(c) The original mesh model "horse"

(d) The "horse" model with 10285 bits embedded

Figure 5. In total, 1626 and 10285 bits are embedded within the mesh model "teapot" and "horse", respectively, by choosing 0.001 as the quantization step.

5 Concluding Remarks

In this paper, a new quantization scheme has been proposed to improve the fragility of the watermark embedded in fragile watermarking based on quantization. Given a set of embedding primitives representing authenticity and integrity of media content, a watermark can be embedded by slightly modifying them in a sequential way. If only one resulting primitive in the watermarked content is modified, the change will be detected if it exceeds one quantization step. If there are multiple resulting primitives having been modified together, the chance to detect illegal modifications can be increased by applying the proposed scheme. By utilizing a new quantization method, it is hard to estimate the quantization step from the resulting primitives, whereas slight change of the resulting primitives can be allowed.

The proposed scheme has been applied to a high-capacity fragile watermarking algorithm, whereby a new algorithm has been presented for mesh authentication. Experimental results have shown that the proposed scheme does not affect the imperceptibility, capacity and security of the fragile watermarking based on quantization, but improves the fragility of the embedded watermark. Nevertheless, the content-preserving manipulations can still be allowed to a certain degree without altering the embedded watermark. Therefore, the proposed scheme can be applied for multimedia authentication to allow some content-preserving manipulations whereas detecting illegal ones more efficiently.

(a)

(b)

Figure 6. On the left, the position of only one vertex on the handle of the watermarked mesh model "teapot" in Fig. 5(b) has been modified; on the right side, those vertices where the original and retrieved bit values do not match with each other have been eliminated from the tampered mesh model.

Acknowledgment

The mesh models used in the experiments are from http://www.cs.unc.edu/ isenburg/ac/models/ and http://pascal.leyaud.free.fr/3ds/, respectively.

References

[1] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding-A Survey," *Proceedings of the IEEE*, vol. 87(7), pp. 1062-1078, July, 1999.

[2] I. J. Cox, M. L. Miller and A. McKellips, "Watermarking as Communications With Side Information," *Proceedings of the IEEE*, Vol. 87, pp. 1127-1141, 1999.

[3] B. Chen and G. W. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding," *IEEE Transactions on Information Theory*, vol. 47, pp. 1423-1443, May 2001.

[4] C. G. Boncelet Jr., "The NTMAC for Authentication of Noisy Messages," *IEEE Transactions on Information Forensics and Security*, Vol. 1, pp. 35- 42, 2006.

[5] P. W. Wong and N. Memon, "Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification," *IEEE Transactions on Image Processing*, Vol. 10, pp. 1593-1601, 2001.

[6] M. U. Celik, G. Sharma, E. Saber and A. M. Tekalp, "Hierarchical Watermarking for Secure Image Authentication with Localization," *IEEE Transactions on Image Processing*, Vol. 11, pp. 585-595, 2002.

[7] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. ACM SIGGRAPH*, pp. 131-144, 2000.

[8] The protein data bank, http://www.rcsb.org/pdb/.

[9] B. L. Yeo and M. M. Yeung, "Watermarking 3-D Objects for Verification," *IEEE Computer Graphics and Application*, pp. 36-45, Jan./Feb. 1999.

[10] H. Y. S. Lin, H. Y. M. Liao, C. S. Lu and J. C. Lin, "Fragile Watermarking for Authenticating 3-D Polygonal Meshes," *IEEE Transactions on Multimedia*, Vol. 7, No. 6, pp. 997-1006, 2005.

[11] O. Benedens and C. Busch, "Toward Blind Detection of Robust Watermarks in Polygonal Models," *Proceedings of the EUROGRAPHICS'2000, Computer Graphics Forum*, vol. 19, pp. C199-C208, 2000.

[12] F. Cayre and B. Macq, "Data Hiding on 3-D Triangle Meshes," *IEEE Transactions on Signal Processing*, vol. 51, pp. 939-949 (4), 2003.

[13] H. T. Wu and Y. M. Cheung, "A Reversible Data Hiding Approach to Mesh Authentication," *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'2005)*, pp. 774-777, 2005.

[14] H. T. Wu and Y. M. Cheung, "A High-Capacity Data Hiding Method for Polygonal Meshes," *to appear in the Proceedings of the 8th Information Hiding International Conference (IH'2006)*, Virginia, USA, July, 2006.

[15] O. Benedens, "Geometry-Based Watermarking of 3-D Models," *IEEE Computer Graphics and Application, Special Issue on Image Security*, pp. 46-55, 1999.

[16] E. Praun, H. Hoppe and A. Finkelstein, "Robust Mesh Watermarking," *Proceedings of ACM SIGGRAPH*, pp. 69-76, 1999.

[17] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications," *IEEE Journal of Selected Areas in Communications*, vol. 16, pp. 551-560, Apr. 1998.

[18] R. Ohbuchi, S. Takahashi, T. Miyasawa and A. Mukaiyama, "Watermarking 3-D Polygonal Meshes in the Mesh Spectral Domain," *Proceedings of Graphics Interface 2001*, pp. 9-17, Ottawa, 2001.

[19] S. Zafeiriou, A. Tefas and I. Pitas, "Blind Robust Watermarking Schemes for Copyright Protection of 3D Mesh Objects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11(5), pp. 596-607, 2005.

[20] H. Date, S. Kanai and T. Kishinami, "Digital Watermarking for 3-D Polygonal Model Based on Wavelet Transform," *Proceedings of ASME Design Engineering Technical Conference*, Sept. 12-15, 1999.

[21] F. Uccheddu, M. Corsini and M. Barni, "Wavelet-Based Blind Watermarking of 3d Models," *Proceedings of ACM Multimedia & Security Workshop*, pp. 143-154, Magdeburg, Germany, 2004.

[22] Y. Maret and T. Ebrahimi, "Data Hiding on 3D Polygonal Meshes," *Proceedings of the 2004 workshop on Multimedia and security* , pp. 68-74, Magdeburg, Germany, 2004.

[23] S. Bounkong, B. Toch, D. Saad and D. Lowe, "ICA for Watermarking Digital Images," *Journal of Machine Learning Research*, Vol. 4, pp. 1471-1498, 2003.

[24] B. Chen and G. W. Wornell, "Dither Modulation: A New Approach to Digital Watermarking and Information Embedding," *Proc. SPIE: Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 342-353, San Jose, January 1999.

[25] D. Kundur and D. Hatzinakos, "Digital Watermarking for Telltale Tamper Proofing and Authentication," *Proceedings of the IEEE*, vol.87, no. 7, pp. 1167-1180, July 1999.

[26] M. Holliman and N. Memon, "Counterfeiting Attacks for Block-wise Independent Watermarking Techniques," *IEEE Transactions on Image Processing*, Vol. 9(3), pp. 432-441, 2000.

[27] J. Fridrich, M. Goljan and N. Memon, "Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme," *Proc. SPIE, Security and Watermarking of Multimedia Contents II*, Vol. 3971, pp. 428-437, 2000.

[28] The Web3D Consortium, http://www.vrml.org/

Value Directed Compression for POMDP Planning With Belief State Analysis

LI Xin

Abstract

Partially observable Markov decision process (POMDP) is commonly adopted for modelling planning problems under uncertainty. Obtaining the optimal policy of POMDP for large-scale problems is known to be intractable, where the high dimension of its belief state is one of the major causes. The use of the compression approach has recently been shown to be promising tackling the curse of dimensionality problem. In this paper, a novel value-directed belief compression technique is proposed together clustering of belief states to further reduce the underlying computational complexity. We first cluster a set of sampled belief states into disjoint partitions and then apply a non-negative matrix factorization (NMF) based projection to each belief state cluster for dimension reduction. The optimal policy is then computed using a pointed-based value iteration algorithm defined in the low-dimensional projected belief state space. The proposed algorithm has been evaluated using the synthesized navigation problem, where solutions of comparable quality were obtained at a significantly lower in computational cost.

1 INTRODUCTION

Building intelligent agents which can make optimal decisions in a stochastic environment is known to be important for some applications like robot navigation [13], machine vision [2], service composition [4], etc. One crucial issue is how to compute the optimal policy for an agent to decide its next action based on some feedback observed from the environment so as to maximize its long-term reward and at the same time complete a given mission. The stochastic nature of the problem includes the uncertainty of the environment and that of the agent's performed actions. The agent policy can in general be defined as a mapping of its observed information to the recommended actions to be performed. The ideal scenario of the problem is that the stochastic environment can be accurately modelled by a set of abstracted fully observable true states, (denoted as S) with stochastic state transitions. However, in many real situations, we cannot expect the agent to have full but only partial observation reflecting the current true state. This partially observable situation makes the optimal decision making challenging as the agent needs ways to efficiently abstract its inflatable observation history for its decision on next action.

Partially observable Markov decision process (POMDP) is a model that extends Markov decision process (MDP) to partially observable situations, where the belief state, defined as the probability distribution over the unobservable true states, is estimated to summarize all the past observation history via Bayesian updating. The belief states form a continuous state space of $|S|-1$ dimensions (later on called belief space). The optimal policy becomes a mapping between actions and belief states.

To compute the optimal policy, iterations of a value function — defined as the expected reward of the agent are typical adopted. The best complexity bound for obtaining the optimal policy of POMDP with t step ahead considered is $O(\gamma_{t-1}^{|Z|})$[1] where γ_i is the space complexity of the value function of the i^{th} iteration. Even though there exist some computational shortcuts which make use of the piecewise linear and convex (PWLC) property of the value function (e.g., the witness algorithm [1]). Large-scale POMDP problems are still widely considered to be computationally infeasible In the literature, there exist a number of approximation methods proposed to solve large-scale POMDP problems efficiently. The point-based value iteration (PBVI) [5, 6] algorithm optimizes the value function based on a finite set of beliefs instead of over the entire belief space. The value directed compression (VDC) [12] engender a deflated POMDP by maintaining the policy's value. The belief compression[10] tends to reduce the POMDP problem by exploring the sparsity of belief space. An orthogonal alternative is to take the divide-and-conquer approach [11].

In this paper, we propose a way to combine the ideas of the recently proposed belief compression and value directed compression and then argue that clustering the belief state for problem decomposition can further reduce the complexity in a problem-specific manner. In particular, we apply first, a k-means algorithm which takes care of the belief states' temporal relationship for the clustering, with the conjecture that temporally close belief states are more likely to form clusters with further reduced intrinsic dimensions. we then non-negative matrix factorization (NMF) to

each cluster for belief compression. The use of NMF can guarantee the elements of the low-dimensional representation of belief states to be non-negative, which is important as the representations are by themselves probability distribution. Following the value directed approach as in [12], we apply NMF the second time so that the low-dimensional POMDP approximate for each cluster (including the low-dimensional reward and state transition functions) can be quickly computed, resulting in a set of sub-POMDPs. For computing the optimal policies for each sub-POMDP, we adopt the point-based value iteration technique. Here, we report the detailed analysis of proposed method using a typical synthesized navigation problem — hallway2 problem and show that proposed method can result in policies of superior long-term rewards with less time cost.

This paper is organized as follows. Section 1 reviews the background of POMDP. Section 2 analyzes the inspiring existing compression methods and illustrates the intuition of our proposed method. Section 3 and Section 4 describe the details of a novel value directed compression integrated with belief clustering. Section 5 shows the policy strategies and the experimental results. Concluding remarks and future research directions are included in Section 6.

2 LITERATURE REVIEW ON COMPRESSION OVER POMDP

2.1 FORMULATION

A standard POMDP model is characterized by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, O, R >$, which contains a finite set of real states $\mathcal{S}$, a finite set of actions $\mathcal{A}$, the state transition probabilities $T : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$, a reward function which depends on the action and the state $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$, a finite set of observations $\mathcal{Z}$ and a set of corresponding observation probabilities $O : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{Z})$. Solving POMDP problems typically makes use of the belief state concept. A belief state is defined as a probability mass function over the current state, given as $b = (b(s_1), b(s_2), ...b(s_{|S|}))$, where $s_i \in \mathcal{S}, b(s_i) \geq 0$, and $\sum_{s_i \in \mathcal{S}} b(s_i) = 1$. $b_j = SE(b_i, a, z)$ is defined using Eq.(1) and (2).

$$\begin{aligned} b_{t+1}(s_j) &= P(s_j|z, a, b_t) \\ &= \frac{O(s_j, a, z) \sum_{s_i \in \mathcal{S}} T(s_i, a, s_j) b_t(s_i)}{P(z|a, b_t)} \end{aligned} \quad (1)$$

$$P(z|a, b_t) = \sum_{s_j \in \mathcal{S}} O(s_j, a, z) \sum_{s_i \in \mathcal{S}} T(s_i, a, s_j) b_t(s_i) \quad (2)$$

The reward function if the j^{th} belief state b_j can then be computed as $\rho(b_j, a) = \sum_{s_i \in \mathcal{S}} b_j(s_i) R(s_i, a)$. Also, the transition function over the belief states becomes $\tau(b_i, a, b_j) = p(b_j|b_i, a)$ (see [1] for more details). To compute the optimal policy $\pi : \mathcal{R}^{|\mathcal{S}|} \rightarrow \mathcal{A}$ iteratively, a value function is typically involved, as

$$V(b_i) = max_a[\rho(b_i, a) + \gamma \sum_{b_j} \tau(b_i, a, b_j) V(b_j)] \quad (3)$$

$$\pi^*(b_i) = argmax_a[\rho(b_i, a) + \gamma \sum_{b_j} \tau(b_i, a, b_j) V(b_j)]. \quad (4)$$

Where γ is the discounting factor for the past history. In practical, it is common to have the optimal policy represented by a set of linear functions (so called α vectors) over the belief space, with the maximum "envelop" of the intersections forming the overall value function. The computation complexity for identifying the hyperplane's intersections is proportional of the hyperplane's dimension which is the infamous "curse of dimensionality" of POMDP.

2.2 BELIEF COMPRESSION

Belief compression is a recently proposed approach [10] to dispel the curse of dimensionality, by reducing the sparse high-dimensional belief space to a low-dimensional one via a projection. The idea behind is to explore the redundancy in computing the optimal policy for the entire belief space which is typically sparse. Using a sample of belief states obtained based on observations of a specific problem as the training set, data analysis techniques, e.g., exponential principal component analysis (EPCA), can be adopted in to characterize the originally high-dimensional belief space using a compact set of belief state bases. This approach has been found to be effective in making some POMDP problems much more tractable. However, as the transformation used is non-linear the value function of the projected belief states is no longer piecewise linear. The consequence is that many existing algorithms taking the advantage of the piecewise-linear and convex property of value function become not applicable in the context of belief compression. In [10], the sampled belief states in the projected belief space were used as the states of an associated MDP that can approximate the original POMDP, and one can compute the policy using the steps of MDP. The limitation is that the quality of the resulting policy now depends not only on that of the belief compression, but also that of the grid-like approximation for policy computation. The latter one is problem dependent, making the applicability of this non-linear belief compression approach restricted. But of course, non-linear transformation is more effective in digging out the structure of the high-dimensional data than the linear ones given the same compression ration. Also, this belief analysis approach can further be extended to belief clustering so

that an even more compact set of belief state bases can be resulted [7].

2.3 VALUE DIRECTED COMPRESSION

Another interesting approach to address the dimensionality issue is value directed compression (VDC) [12] where a linear projection is used instead. VDC computes the minimal *Krylov* subspaces and thus the corresponding reward and state transition functions so that the value governing the agents' decision policy remain before and after the compression (thus called value directed). To contrast with the belief compression approach, value-directed compression does not performing any data analysis on belief space but to compute a sub-space which is invariant to the compression projection matrix. As the projection is linear, the value function after the projection remains to be PWLC and thus most of the existing efficient algorithms for the policy solving can be adopted. Computing the *Krylov* sub-space is however time-consuming as a large number of linear programming problems are to be solved and yet a high compression ratio cannot be guaranteed. While a truncated *Krylov* iteration algorithm and an alternating optimization algorithm have been introduced in [12] for obtaining a forcibly compressed POMDP, these approximations were achieved by either quickly stopping the *Krylov* iterations with loss due to the incomplete set of belief bases and by minimizing the errors between high dimensional policy value and low dimensional one respectively. There exist no mechanism for exploring the characteristics of each specific belief space as what the belief compression approach can provide. Thus, the compression quality could be rather limited. Also, the *Krylov* space analysis in VDC is applied to the whole belief space and it is not straight forward to see how it can be combined with the belief analysis approach and how the notion of sub-space computation can support the problem decomposition as suggested.

3 VALUE DIRECTED COMPRESSION WITH BELIEF SPACE ANALYSIS

In this section, we propose a novel value directed compression method that has belief state analysis incorporated as well. Recall that the goal of the value directed compression is to keep value remain unchanged between the original problem and its reduced version. Given the new reward function to be $\tilde{R}$ and the new value function to be $\tilde{V}$, it has been proved In [12] that as long as we can find the proper reward function and transition functions with the base case $V_0^\pi(b) = R(b) = \tilde{R}(\tilde{b}) = \tilde{V}_0^\pi(\tilde{b})$, we could keep $V_{t+1}^\pi(b) = \tilde{V}_{t+1}^\pi(\tilde{b})$ hold through the whole horizon whatever it is finite or infinite. Therefore, the key point is how to find the proper reward function and transition functions in the low-dimensional belief space.

3.1 BELIEF COMPRESSION BY NON-NEGATIVE MATRIX FACTORIZATION (NMF)

To explore the belief space's sparsity for compression and at the same time preserve the PWLC property of the POMDP's value function, we consider only linear data projection techniques, instead of the non-linear EPCA's extension as described in [7]. Moreover, the projection is expected to be able to guarantee all the elements of the reduced and the reconstructed belief state to be positive as belief states, no matter in which space, are probability distributions by themselves. Thus, some standard dimension reduction techniques like PCA are not suitable. Instead of using the *Krylov* iteration to compress the problem, we adopt the non-negative matrix factorization (NMF)[3] and apply it to the belief state sample to get the reduced dimension belief space. NMF is a technique to compute a *linear* and *non-negative* representation for approximating a given set of data. Given V to be an $M \times N$ matrix with each of its columns being an observation with possible elements, one can approximate V using NMF so that $V \approx WH$, where W is an $M \times P$ matrix with its column forming a set of P (normally $< M$) non-negative basis components and the matrix H are the coefficients of the corresponding basis components. Intuitively, V is approximately represented by a weighted sum (H) of the basis components (W). W and H are derived using the updating rules in Eq.(5) and Eq.(6), given as

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \quad (5)$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu W_{a\nu}} \quad (6)$$

Eq.(5) and Eq.(6) alternates until the values of W and H converge. It has been shown that the updating rules are in effect minimizing a generalized Kullback-Leibler (KL) divergence between V and WH.

3.2 INCORPORATING NMF INTO VDC

Let B denote a $n \times |S|$ matrix defined as $[b_1|b_2|...|b_n]^T$ where n is the number of belief states in the training sample, and $b_i(S_j) \geq 0$ is the j^{th} element of b_i. Also, denote F to be the $|S| \times l$ transformation matrix which factors B into the matrices F and $\widetilde{B}$ such that

$$B^T \approx F^T \widetilde{B}^T \quad (7)$$

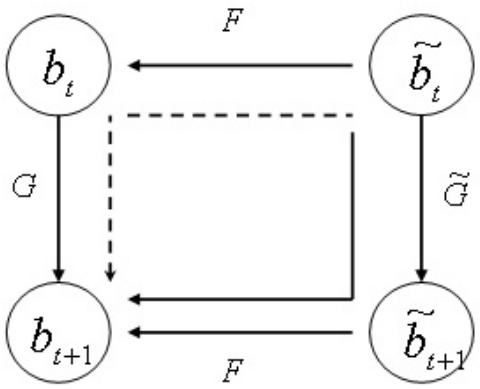


Figure 1. Transformations between beliefs

where each row of B equals $b \approx b^r = \widetilde{b}F$ and the dimension of $\widetilde{B}$ is $n \times l$[1]. As the main objective of deriving F is for dimension reduction, it is typical that $l << |S|$. To derive the reward function for the low-dimensional belief space after compression, it is first observed that in the original belief space,

$$\begin{aligned} V(b) &= \sum_{s_i} b(s_i) \cdot R(s_i, a) \\ &= bR_{\cdot a} \\ &= \widetilde{b}FR_{\cdot a}. \end{aligned} \tag{8}$$

In the compressed belief space,

$$V(\widetilde{b}) = \widetilde{b}\widetilde{R}_{\cdot a}. \tag{9}$$

Combining Eq.(8) and Eq.(9) by allowing $V(b) = V(\tilde{b})$ gives

$$\widetilde{R} = FR. \tag{10}$$

To derive the low-dimensional state-transition function, let's consider the two different paths for computing the next belief state in the high-dimensional space (shown as solid and dotted path in Figure 1). One can then obtain Eq.(11) and Eq.(12), given as

$$\begin{aligned} \widetilde{b}^T_{t+1} &= \widetilde{SE}(\widetilde{b}_t, a, z) \\ &= \widetilde{G}^{<a,z>}\widetilde{b}^T_t \end{aligned} \tag{11}$$

$$\begin{aligned} b^T_{t+1} &= F^T\widetilde{b}^T_{t+1} \\ &= SE(b_t, a, z) \\ &= G^{<a,z>}b^T_t \\ &= G^{<a,z>}F^T\widetilde{b}^T_t. \end{aligned} \tag{12}$$

[1]Note that $V \approx WH$ and $V^T \approx H^T W^T$.

Combining Eq.(11) and Eq.(12) by Eq.(7) gives

$$\begin{aligned} F^T\widetilde{G}^{<a,z>}\widetilde{b}^T_t &= G^{<a,z>}F^T\widetilde{b}^T_t \\ F^T\widetilde{G}^{<a,z>} &= G^{<a,z>}F^T. \end{aligned} \tag{13}$$

It is noted that Eq.(10) and Eq.(13) are essentially equivalent to the corresponding criteria obtained in [12], with the exception the F defined here is the inverse of the equivalence in [12]. One advantage of our formulation is that this is no need to obtain $\tilde{R}$ using LP technique as in [12]. To obtain $\widetilde{G}^{<a,z>}$, one intuitive solution is to take $\widetilde{G}^{<a,z>} = pinv(F^T)G^{<a,z>}F^T$, where $pinv(X)$ is a pseudo inverse of X. However pseudo inverse has no non-negative guarantee for each entry of the matrix and thus can result in $\widetilde{G}^{<a,z>}$ with non-negative values. Another alternative to solve Eq.(13) is to use constrained linear programming which is however computationally demanding. Instead, we adopt NMF again by taking the known $G^{<a,z>}F^T$ as V and the F^T as the W. Then, we update only H by the H-updating rule Eq.(5)with W kept unchanged.

It is obvious to prove the convergence of this H-only updating. Also, due to the property of NMF, all the elements of $\tilde{G}$ can be guaranteed to be non-negative. To contrast with VDC proposed in [12] which requires solving a large number of linear programs, what we propose is more efficient in computing the projection, reward function and state transition functions.

4 POMDP DECOMPOSING VIA BELIEF CLUSTERING

4.1 DIMENSION REDUCTION ORIENTED CLUSTERING

For value directed compression (and in fact other compression-based methods as well), the policy computation efficiency depends very much on the effectiveness of belief compression. As expected, the efficiency gained due to te compression is at the expense of the accuracy. As shown in Figure 4 and 5, we can see that the policy quality greatly increases as the KL-divergence between the original and the reconstructed belief-state decreases. To further exploit the dimension reduction paradigm, we propose to decompose the belief space by applying clustering techniques to the sampled belief states for clustering. We anticipate that in most of the cases, there should exist some clusterings which could result in more substantial per-cluster dimension reduction when compared with that of the overall belief states.

4.2 A DISTANCE FUNCTION WITH TEMPORAL RELATIONSHIP CONSIDERED

As in [7], we propose to cluster the belief states using a distance based on both their Euclidean distance in the belief space as well as temporal difference of the belief state sample, with the conjecture that regularities should be easier to identify for spatially (in belief space) and temporally close belief states. In particular, we define a distance function between two belief states, given as

$$dist(b_i, b_j) \quad = \quad \min(\|b_i - b_j\|, \|\frac{i-j}{\lambda}\|) \qquad (14)$$

where λ is a scaling parameter to align the scale for the distances in the two different spaces. The k-means algorithm [9] is used for the clustering due to its simplicity.

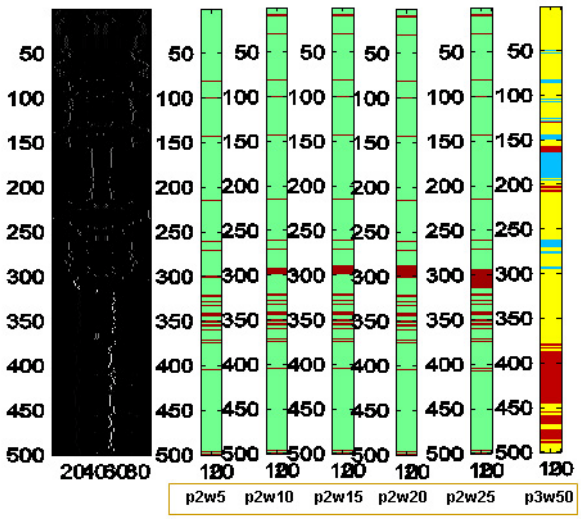


Figure 2. An illustration of the cluster cases in different parameter settings.

As each belief state is a probability distribution, the Kullback-Leibler (KL) divergence is used again for evaluating the discrepancy between the original and the reconstructed belief states, given as

$$\overline{KL}(B) = \frac{\sum_{i=1}^{n} KL(b_i \| b_i^r)}{n} \qquad (15)$$

$$KL(b_i \| b_i^r) = \sum_{j=1}^{|S|} b_i(s_j) \ln \left(\frac{b_i(s_j)}{b_i^r(s_j)} \right). \qquad (16)$$

Based on the clustering results, the belief state sample can be partitioned into K clusters $\{C_1, C_2, ..., C_K\}$ and thus there are K NMF transformation matrix, with each F_k corresponding to the cluster C_k.
Figure 2 shows the clustering cases in different parameter settings in our experiment(See Section 6) .

5 POLICY COMPUTATION AND APPLICATION

5.1 POINT BASED VALUE ITERATION

As mentioned before, NMF involves linear transformations only and thus can preserve the PWLC properties, making the POMDP of reduced dimension suitable to be solved by any existed POMDP algorithms. Here, we choose Perseus [14] an efficient randomized point-based approximate value iteration algorithm for computing the policy. The combination is straightforward. But it is to be noted Perseus requires a backup belief set to be used for limiting the number of stored α vectors. As, Perseus creates the set using the trajectory-based approach which is common for belief state generation. Therefore, we take the same sampled belief states for both the belief clustering (Section 4.1) and policy computation. Intuitively, we believe that this makes the belief space analysis somehow more consistent to the backup beliefs which affect the value iteration.

5.2 AGGREGATING THE SUB-POMDP'S POLICIES

The policy computation will operate independently by the sub-POMDPs defined for each the clustered belief state clusters. One cluster will be associated with one transformation function F_k with the corresponding policy computed using Perseus space. Note that Perseus computes a set of α vectors (thus the optimal value function) over the belief space.

For policy application in un-partitioned POMDPs, the current belief state can just be compressed directly with the one and only one compression matrix F. The α vector that gives the highest utility value can then be identified and the best action can be selected. For partitioned cases, we need to determine which cluster the current belief state belongs to and compute the policy based on the set of α vectors in the corresponding sub-POMDP. We considered three different strategies for aggregating the sub-POMDP's policies:

- Strategy A - Nearest neighbor with α vectors in original belief space.

  This strategy identifies that the current belief state and all the belief state sample points. Once identified, the best α vector corrected by $\alpha = \tilde{\alpha}F$ can be computed for selecting the best action accordingly.

- Strategy B - Nearest neighbor with α vectors in compressed belief space.

  Similar to strategy A, this strategy identifies the cluster that the current belief state belongs to by comparing in the original belief space the current belief state and all the belief state sample points. Once identified, the current belief state is compressed by the corresponding F. The best α vector in the low-dimensional space can be computed for selecting the best action.

- Strategy C - Winner-Take-All

  This strategy computes, for each cluster, the best α vector and the corresponding expected reward. The cluster withe best reward is chosen and the corresponding action is selected.

6 EXPERIMENT

6.1 THE HALLWAY2 PROBLEM

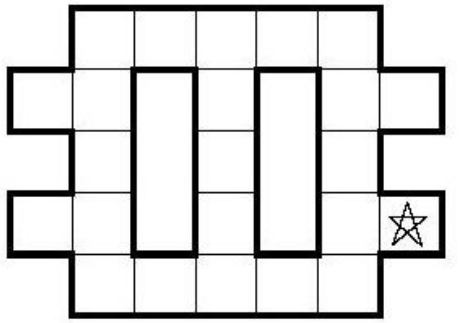


Figure 3. Navigation environment with 92 states

The Hallway2 Problem which is constructed by [8] to model a robot navigation domain with a specific maze is commonly used to test the scalability of algorithms for solving POMDP problems (see Figure 3). The problem is to find the goals in the maze with 92 states (4 possible orientations in each of 22 rooms, and 4 being the goal states which are 4 possible orientations in the star room), and contains 5 actions (stay in place, move forward, turn right, turn left, turn around) and 17 types of observations (all combinations of walls, plus "reaching goal"). Reaching one of the goal states will yield a $+1$ reward and then the next state will be set to a random non-goal state. In addition, it is assumed that all the non-goal states of the problem are equally likely to be the initial state location and thus the starting belief state is $b_1 = (\frac{1}{88}, ..., \frac{1}{88}, 0.0, 0.0, 0.0, 0.0, \frac{1}{88}, ..., \frac{1}{88})^T$. The

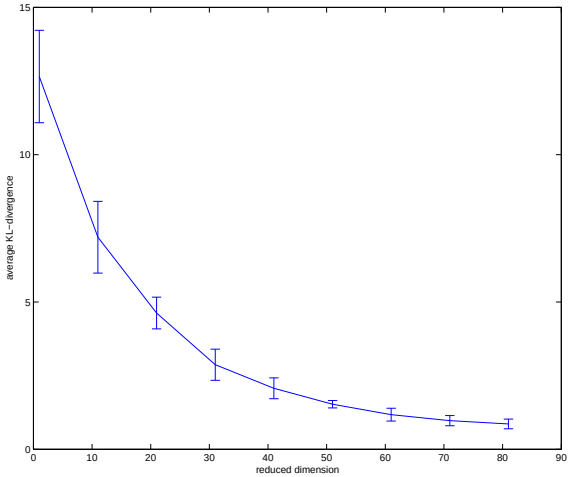


Figure 4. KL-divergence between the original and reconstructed belief states over different reduced dimensions.

zeros are corresponding to four goal states. Also, the discount factor used is 0.95. In this paper, all the experimental results reported are based on this problem setting.

6.2 BELIEF STATE SAMPLING

The process of belief compression is operated on a belief state sample generated via simulation. During the simulation for sample generation, two levels of random numbers are used to select an action (one level is used to determine when to call MDP program, another level is used to determine the random action), and the Bayes rules are used to evolve the belief states. When one random number is found to be less than the threshold defined as 0.5, another random number will be generated to decide the next action. Otherwise, it will sum up all the beliefs generated so far and take the state with the maximal sum of probabilities as the current state. Then, an MDP solver will be called to get the corresponding policy table to choose the next action for its 'current state'.

6.3 RESULT INTERPRETATION

The experiments show that the proposed method owns the promising results in both enhancing the effectiveness of policy and efficiency of the policy computing.

Experientially, we use $\lambda = 25$ as a proper scaling parameter (See Figure 6). Table 1 gives a snapshot about the

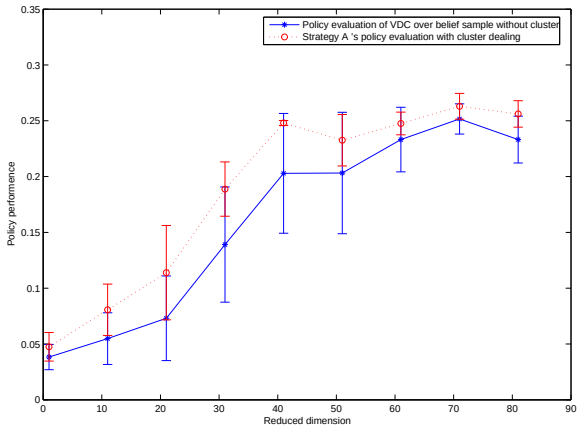


Figure 5. Policy performance over different reduced dimensions.

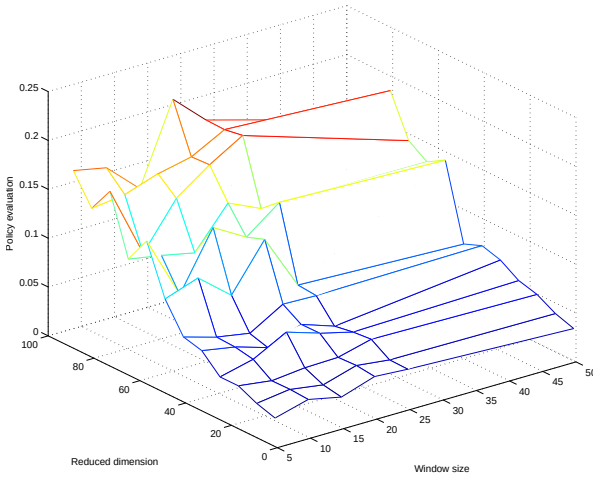


Figure 6. An illustration for an experiential scaling parameter selection.

same policy evaluated under 3 different strategies. our experiments show In general, $StrategyA \succ StrategyB \succ StrategyC$. Figure 4 and the blue line of Figure 5 show a reasonable policy tendency that the policy performance is decreasing along with the number of dimension's reducing since the accuracy of belief set's factorization and reconstruction using NMF is decreasing. Figure 5 shows a comparison between the evaluated policy got from our proposed method over the belief sample without clustering and a clustering one evaluated by strategy A, and it is obvious to see the latter one is better.

Spending time (sec.)	# reduced dim	Strategy A	Strategy B	Strategy C
100	33	0.1843	0.1615	0.2608
100	35	0.0694	0.1219	0.0514
100	37	0.2462	0.1185	0.0670
100	39	0.2478	0.2429	0.0640
100	41	0.2414	0.1036	0.0449

Table 1. Performance comparison among the different strategies, where the number of clusters is 2, the scaling parameter setting $\lambda = 25$.

While Table 2 shows that the aggregated policy derived from the clusters via belief space analysis has much more superiority than the policy derived from the original belief space directly under limited policy solving time.

	# Spending Time (sec.)	Case dim39	Case dim37
	80	0.2589	0.2058
	70	0.2297	0.2058
	60	0.2297	0.2058
Decomposing	50	0.2532	0.2016
Cases	40	0.2532	0.2031
	30	0.2626	0.2031
	20	0.2503	0.1934
NODecomposition	100	0.1903	0.1457

Table 2. A comparison between the case using belief space decomposition and without decomposition ones under limited solving spending time. Where the case dim37, case dim39 are the cases with reduced dimension on 37 and 39 respectively whose specific parameter settings include: the number clusters is 2, the scaling parameter setting $\lambda = 25$.

7 CONCLUSION

While the experimental results obtained so far are positive, there still exist a number of possible extensions to further improve the proposed methods. While the distance function used in this paper has shown to be effective empirically, it is by no means an optimal choice. In addition, we still lack automatic mechanisms for setting the parameters to govern the clustering. We believe that this is an immediate and important extension of this work to be pursued in the future. In addition, Hierarchical POMDP (HPOMDP) [11] Decomposition has recently been proposed for decomposing a POMDP problem into a hierarchy of related POMDP models of reduced size. The limitation, however is that the decomposition has to ben done manually, requiring knowledge of domain experts. It would be interested to see if the notion of hierarchical decomposition can be incorporated in the proposed decomposition-by clustering framework with further performance boosting. Furthermore, the decomposition based on the proposed belief clustering may not result in a set of sub-POMDP problems which are equivalent to the original POMDP problems, interaction between those agents for achieving the overall optimal policy is an another interesting direction to look at.

Lastly, what being described so far assumes that the whole model of the decision process is known. That is, we have the perfect knowledge about the reward function, the state transition function and the observation function. Solving the corresponding POMDP problems is an off-line process. It is also interested to see how the proposed method can be extended to support online learning (e.g., Q-learning [15]) of POMDP under the partial observation scenario.

In all, this paper illustrates the motivation and the details of a novel value-directed compression method which explores also the belief space sparsity and clustering for dimension reduction, and thus computational complexity reduction.The proposed method has been evaluated using a navigation related problem with positive results. The clustering model was demonstrated to be able to further save the policy solving time at the same time remain the policy performance.

References

[1] A.Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. U.Brown, 1998.

[2] Bandera, Cesar, Vico, Francisco J., Bravo, Jose M., Harmon, Mance E., and Baird III, Leemon C. Residual Q-Learning Applied to Visual Attention. In *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*, pages 20–27. Morgan Kaufmann, 1996.

[3] D. D.Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 1999.

[4] P. Doshi, R. Goodwin, R. Akkiraju, and K. Verma. Dynamic workflow composition using markov decision processes. *JWSR*, 2:1–17, 2005.

[5] M. Hauskrecht. Incremental methods for computing bounds in partially observable markov decision processes. In *Proceedings of the 14th National Conference on Artificial Intelligence, AAAI'97, Providence, Rhode Island, USA*, pages 734–739. AAAI Press, 27–31 1997.

[6] M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of AI Research*, 13:33–94, 2000.

[7] X. Li, W. K. Cheung, and J. Liu. Decomposing large scale pomdp via belief state analysis. In *Proceedings of 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)*, Compiegne, France, 2005.

[8] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. pages 495–503. 1997. (Reprinted from *Proceedings of the 12th International Conference on Machine Learning, 1995*).

[9] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkley Symposium on Mathematics and Probability*, pages 281–297, 1967.

[10] N. Roy, G. Gordon and S. Thrun. Finding approximate POMDP solutions through belief compressions. *Journal of Artificial Intelligence Research*, 23:1–40, 2005.

[11] J. Pineau and S. Thrun. An integrated approach to hierarchy and abstraction for POMDPs. Technical Report CMU-RI-TR-02-21, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2002.

[12] P. Poupart and C. Boutilier. Value-directed compression of POMDPs. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, 2003.

[13] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. *Artificial Intelligence Journal*, 1997.

[14] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[15] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge Univ., Cambridge England, 1989.

A Semi-supervised SVM for Manifold Learning

Zhili Wu, Chunhung Li
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
vincent@comp.hkbu.edu.hk, chli@comp.hkbu.edu.hk

Abstract

Many classification tasks benefit from integrating manifold learning with semi-supervised learning. By formulating the learning task in a semi-supervised manner, we propose a novel objective function that combines the manifold consistency of whole dataset with the hinge loss of class label prediction. This formulation results in a SVM-alike task operating on the kernel derived from the graph Laplacian, and is capable of capturing the intrinsic manifold structure of the whole dataset and maximizing the margin separating labelled examples. Results on face and handwritten digit recognition tasks show significant performance gain. The performance gain is particularly impressive when only a small training set is available, which is often the true scenario of many real-world problems.

1 Introduction

The learning of real world data often benefits from manifold learning, such as the classification tasks arising from face and handwritten recognition and text categorization [9, 10]. The learning of intrinsic manifold structure of data by manifold learning methods can often helps exploit intrinsic information of the dataset [14]. The manifold structure of the data is often captured by a weighted graph and then graph operators, e.g. Laplacian, can be used for manifold learning [1].

Many classification tasks can also take advantage of semi-supervised learning (or transductive inference [11]). In this setting, some unlabelled examples are also available for training the classifier. Rather than solely training on the data with known labels in an inductive way, semi-supervised learning can further consider the distribution or interrelations of both the labelled and unlabelled data to improve the prediction on the unlabelled data.

Many learning tasks suitable for manifold learning can be formulated with the semi-supervised setting too [2, 4]. Their combination is particularly suitable for tasks of scarce labelled points but numerous unlabelled data, and the data have some underlying manifold coherence. The presence of unlabelled data can be used to better recover the manifold structure and consequently improve the prediction. This learning scenario is relevant to tasks where the labelling process is very costly, e.g. in face recognition and protein classification; or large-scale tasks with only a small proportion of data being labelled, e.g. Web page analysis. The combination of manifold learning with semi-supervised learning thus has shown many successes [12, 2].

We propose a new formulation of semi-supervised classification based on the criteria of manifold consistency and the hinge loss of class prediction. It follows the framework of *consistency method* proposed in [12, 13], but adopts a different loss measure for class prediction and consequently results in a novel SVM learning strategy over manifold structures.

This paper is organized as follows. In Section 2, we explain our formulation for semi-supervised manifold learning. In Section 3, the implementation and some further justification are provided, accompanied by the discussion of related methods. In Section 4, some experiments are used to show the superiority of our semi-supervised manifold learning approach. At the end, it is the conclusion section.

2 Formulation of Semi-supervised Manifold Learning

In this section, we explain in detail a new formulation of semi-supervised classification based on the criteria of manifold consistency and the hinge loss of class prediction. The underlying motivation is that manifold learning can take into account of the intrinsic structure of the whole dataset, while the hinge loss can give robust generalization capability for classification.

Given a set of n points $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, each $\mathbf{x}_i$ is associated with a label y_i, $y_i \in \{\pm 1, 0\}$. If $y_i \in \{\pm 1\}$ then the

point is labelled, otherwise unlabelled. Here the labelled points are limited to be the classes of ± 1, but in later experiment part it can be seen that the multiclass case is handled too. For clarity, we assume that the first q ($q << n$) points are labelled. And we further construct a diagonal matrix $\mathbf{Y}$ with $\mathbf{Y}_{ii} = y_i$.

And then, a weighted graph is used to capture the manifold structure of the dataset. This graph can be represented by a symmetric matrix $\mathbf{W}$ with size $n \times n$, with W_{ij} measuring the pairwise affinity between the i-th and j-th points. And disconnected pairs usually are set with a zero value. Further let $\mathbf{D}$ be a matrix with its main diagonal holding the row summation of $\mathbf{W}$. Each D_{ii} thus measures the total weight of edges connected to the point $\mathbf{x}_i$.

Our task is to get a set of value $\mathbf{F} = \{f_i\}_{i=1}^n = \{f(\mathbf{x}_i)\}_{i=1}^n$, such that: first, the values of f_i and f_j are similar if they are connected by an edge with a large weight in the graph; second, $y_i \geq \lambda$, if $y_i = 1$, $y_i \leq -\lambda$, if $y_i = -1$, where $\lambda > 0$. The second condition implies that (f_i, f_j) should be separated by the margin of at least 2λ, if they have different class labels. Hereby we can try to minimize the following objective function:

$$\mathbf{Q}(\mathbf{F}) = \frac{1}{4}\sum_{i,j=1}^{n} W_{ij}(\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}})^2 + \sum_{i=1}^{n}(\lambda - y_i f_i)_+,$$

where $(\lambda - y_i f_i)_+ = \max\{\lambda - y_i f_i, 0\}$ is adapted from hinge loss, which is typical in support vector machines [11].

By introducing slack variables $\xi_i \geq 0$, the objective function becomes,

$$\mathbf{Q}(\mathbf{F}) = \min \frac{1}{4}\sum_{i,j=1}^{n} W_{ij}(\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}})^2 + \sum_{i=1}^{n}\xi_i$$

$$\text{subject to: } y_i f_i - \lambda + \xi_i \geq 0, \forall i, i = 1, \ldots, n$$

Note the left part can be written into a vectorial form:

$$\frac{1}{4}\sum_{i,j=1}^{n} W_{ij}(\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}})^2 = \frac{1}{2}\mathbf{F}'(\mathbf{I} - \mathbf{S})\mathbf{F},$$

where $\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

The Lagrange (primal) of the objective function thus is

$$\begin{aligned} L_p &= \frac{1}{2}\mathbf{F}'(\mathbf{I} - \mathbf{S})\mathbf{F} + \sum_{i=1}^{n}\xi_i \\ &- \sum_{i=1}^{n}\alpha_i y_i f_i + \lambda\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n}\tau_i\xi_i, \\ &\text{subject to: } \alpha_i, \tau_i \geq 0, \forall i, i = 1, \ldots, n. \end{aligned}$$

Let $\boldsymbol{\alpha}$ be the vector containing α_i, take the derivative of L_p with respect to ξ_i and $\mathbf{F}$, and set them to zero, we get

$$\alpha_i = 1 - \tau_i,$$
$$(\mathbf{I} - \mathbf{S})\mathbf{F} = \mathbf{Y}\boldsymbol{\alpha}.$$

Since $(\mathbf{I} - \mathbf{S})$ is shown to be positive semi-definite [5] where the smallest eigenvalue is always zero, we can only obtain its pseudo-inverse. In practice, we can also adopt the regularized form $(\mathbf{I} - \rho\mathbf{S})^{-1} = \mathbf{K}$ to guarantee the invertibility and the positive definite of $\mathbf{K}$, where $\rho < 1$ but close to 1 (e.g. 0.999999). The objective function in dual form thus becomes

$$L_D = \max\{\lambda\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{Y}\mathbf{K}\mathbf{Y})\boldsymbol{\alpha}\}.$$

where $0 \leq \alpha_i \leq 1$. This problem can be solved by standard quadratic programming. It can be obtained that the α_i for an unlabelled point with $y_i = 0$ is 1 always, hereby the quadratic programming is actually of size q, where q is number of labelled points with nonzero label values. The calculation of $\mathbf{K}$ involves all labelled and unlabelled data. After obtaining all α_i, the predicted value for the whole set $\mathbf{X}$ is given by

$$\mathbf{F}(\mathbf{X}) = \mathbf{K}\mathbf{Y}\boldsymbol{\alpha}.$$

Note that the dual-form objective function L_D is equivalent to a SVM-alike form:

$$\begin{aligned} \tilde{L}_D &= \max\{\textstyle\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{Y}\mathbf{K}\mathbf{Y})\boldsymbol{\alpha}\} \\ &\text{subject to: } 0 \leq \alpha_i \leq C \end{aligned} \quad (1)$$

It can be noticed that $L_D = \lambda^2\tilde{L}_D, C = 1/\lambda$. The setup here only differs from a typical SVM in the absence of the constraint $\sum_{i=1}^{n}\alpha_i y_i = 0$ because our formulation directly aims at the prediction value f_i rather than the functional form in a typical SVM, which has an intercept b usually. In our informal testing, adding such b to our formulation, e.g. $f_i + b$, does not bring performance improvement though it can result in an exact SVM setup which can be solved by fast SVM solvers.

Our formulation here even allows each α_i to be associated with a λ_i, which means that for different training points, their f_i can be larger than a unique λ_i, that is, $y_i f_i \geq \lambda_i$, which is useful to add different penalties to points with different confidence.

3 Implementation and Further Justification

3.1 Algorithm

The algorithm for the manifold semi-supervised SVM is as follows:

- Construct a graph of all data points by connecting each point with its k nearest neighbors (KNN-graph) or its neighboring points ϵ-distance away (ϵ-graph), or connecting a portion of smallest distances among all pairs of points till a connected graph is formed, or simply connecting all pairs. There is not an edge connecting from a point to itself.

- For each connected pair, set $W_{ij} = e^{-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2}$; for pair without connection, set $W_{ij} = 0$. The main diagonal entries of $\mathbf{W}$ are all zero too.
- Calculate $\mathbf{D}, \mathbf{S}$ and $\mathbf{K}$. Obtain $\boldsymbol{\alpha}$ by solving (1).
- Get $\mathbf{F} = \mathbf{KY}\boldsymbol{\alpha}$ and take $sign(\mathbf{F})$ as the class outputs.

3.2 Further Justification and Related Methods

Our formulation is directly inspired by the consistency method in [12, 13]. Their framework, however, is limited to the use of quadratic loss term $C\sum_{i=1}^{n}(f_i - y_i)^2$ for close-form solutions. Recently, the hinge-loss formulation has been intensively used for classification tasks and shown great success, e.g. SVM. Here our formulation in the use of hinge loss function is also as competitive as the consistency method.

It is later found that a manifold regularization framework has been proposed [3]. It requires a function form of $\mathbf{F} = \mathbf{K}_1\boldsymbol{\alpha}'$ where $\mathbf{K}_1$ is an additional kernel calculated for all data points, and then the following objective function is minimized:

$$C_1\frac{1}{2}\mathbf{F}'(\mathbf{I}-\mathbf{S})\mathbf{F}+C_2\sum_{i=1}^{n}(1-y_i(f_i+b))_+ + C_3\boldsymbol{\alpha}'\mathbf{K}_1\boldsymbol{\alpha}. \quad (2)$$

This formulation in fact also leads to a SVM-alike QP problem, but the kernel matrix in the QP problem is a more complex combination of $\mathbf{K}_1$ and $\mathbf{I} - \mathbf{S}$. Our formulation can be transformed into a special case of their general setting, by not considering the $\mathbf{K}_1$ and additional parameters. However, as the small number of training data in semi-supervised learning is often insufficient for tuning a complicated model involving lots of parameters, our approach thus is practical to use. We also adopt a slightly different hinge loss term by relating it to λ rather than C_2, which leads to a natural and clear interpretation.

Another similar setup with our work is the measure based regularization [8], it does not use the term associated with C_3 as emerged in Eq. 2 too, but requires the function form of $\mathbf{F} = \mathbf{K}_1\boldsymbol{\alpha}'$ and the intercept b. And also its main aim is not on semi-supervised learning.

4 Experiments

We conduct experiments on face and handwritten recognition tasks. As mentioned for the original framework [12, 13], the model selection problem is still open for the cases of only a few training points, which implies that a validation set usually is not available. In our experiments, we follow the way of reporting the best average testing results chosen from a fairly large parameter range.

4.1 UMIST Face Recognition

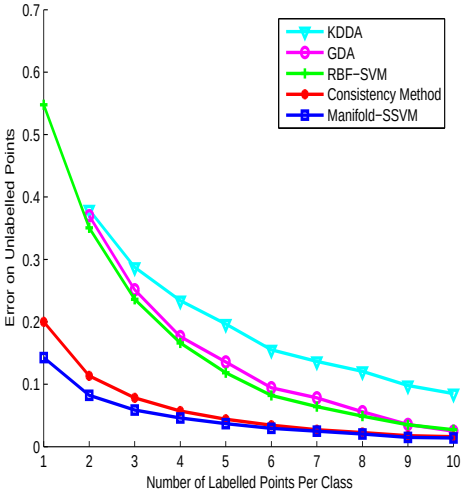


Figure 1. UMIST Face Recognition Results

The UMIST Face Recognition database [6] is tested [1]. It has 575 grayscale face images of 20 different people. Each person has $19 \sim 48$ face images with different views. Each image is processed to be 112×92 in size. For this multiclass task, it is decomposed into multiple binary tasks and then the outputs are combined through one-vs-one mechanism.

In our experiments, the pixel intensity are further divided by 10^3, no other preprocessing is used. Each person is taken $1 \sim 10$ face images for training, with the remaining for testing. Tested methods include SVM with radial basis function kernel (RBF-SVM), generalized discriminant analysis (GDA), Kernel Direct Discriminant Analysis (KDDA), the consistency method and our Manifold-SSVM.

It is expected that enough training images (e.g. $8 \sim 10$ images per person) will result in very comparable performance by these state-of-the-art classifiers. Much attention is thus paid to the setting of very scarce training images (e.g. less than 8), which complies with the practical situation too. For γ in affinity/kernel construction, as well as the C in RBF-SVM and Manifold-SSVM, it is searched from $[10^{-2} \sim 10]$; for the parameter in consistency method, it is from $[0.9 \sim 0.99]$.

In each 100-run experiment, the optimal parameters for Manifold-SSVM, consistency method and RBF-SVM are $(\gamma = 5, C = 0.1), (\gamma = 5, \alpha = 0.99), (\gamma = 0.01, C = 10)$ respectively. The γ in the RBF kernel used by KDDA and GDA is 0.005. The error rates of unlabelled points are shown in Fig. 1. Both the consistency method and Manifold-SSVM clearly outperform RBF-SVM, GDA and

[1] http://www.cs.toronto.edu/∼roweis/data.html

KDDA when the training size is small, though are later approached by them when the training size per person is large enough. Manifold-SSVM performs marginally but consistently better than the consistency method.

4.2 USPS Handwritten Digit Recognition

We test the subset of USPS Handwritten data as tested in [12]. It contains 3874 digit images for digit $1 \sim 4$. Each is represented by a 256-dimensional vector, with values in $[-1, 1]$. For both the Manifold-SSVM and consistency method, the $\sigma = 1/\sqrt{2\gamma}$ in calculating the affinity matrix is 1.25, while for RBF-SVM, σ is 5, and the C in Manifold-SVM and RBF-SVM are set to the optimal value 1. The parameter in the consistency method is 0.99.

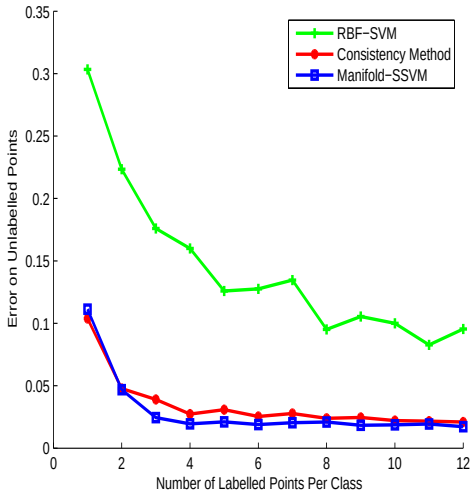


Figure 2. USPS Handwritten

The Manifold-SVM approach can also achieve slight improvement over the consistency method, as shown in Fig. 2.

5 Conclusion and Future Work

This paper extends the semi-supervised manifold learning with the use of robust hinge-loss criteria in class prediction. Our new formulation not only ensures the consistency of the classification with the manifold structure of the whole dataset, but also maximizes the separation margin through introducing the hinge loss. This formulation leads to a SVM-alike task over the kernel derived from the graph of the whole dataset. Experiments show its superior performance over RBF-SVM, GDA and KDDA methods when training points are very scarce. Furthermore, our approach achieves marginal but consistent improvement over the consistency method, though their performance is very close in general. The marginal improvement can arise from the robustness in handling the points near the classifying margin. Further investigation could be carried out to confirm this property.

Other future research issues include computational complexity and the multi-class setup. The calculation of $\mathbf{K}$ involves the inverse of a large matrix, which can cost much computation for large-scale tasks. The possible remedial ways are matrix inversion approximation methods, as well as the approach on reducing the size of unlabelled points. In addition, since many real tasks are multi-class in nature, the study on some systematic and integrated multi-class SVM formulation [7] can be conducted and compared with the one-vs-one decomposition approach.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning (Special Issue on Clustering)*, (56):209–239, 2004.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from examples. In *University of Chicago Computer Science Technical Report TR-2004-06*, 2004.

[4] B. S. D. Zhou and T. Hofmann. Semi-supervised learning on directed graphs. In *NIPS 17*, pages 1633–1640, 2005.

[5] C. Fan. *Spectral Graph Theory*. American Mathematical Society, 1997.

[6] D. B. Graham and N. M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. In *Face Recognition: From Theory to Applications*, pages 446–456, 1998.

[7] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, (99):659–672, 2004.

[8] O. B. Olivier. Measure based regularization. In *NIPS16*, 2004.

[9] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[10] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290(5500):2268–2269, 2000.

[11] V. Vapnik. *Statistical Learning Theory*. New York: Wiley, 1998.

[12] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS 16*, pages 321–328, 2004.

[13] D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Proceedings of the 27th German Association for Pattern Recognition(DAGM) Symposium*, pages 361–368. Springer, 2005.

[14] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS16*, 2004.

Personalized Spam Filtering with Classifier Ensemble

Victor Cheng
Department of Computer Science
Hong Kong Baptist University
victor@comp.hkbu.edu.hk

Abstract

Over the past few years, the proliferation of unsolicited emails, known as spam or junk emails clutter up emailboxes of millions of people worldwide. In order to guard email users against the risks of these emails, spam filters are now available to users. Spam filters generally use the combination of rule-based and learning-based techniques to discriminate normal emails from spam emails. They are trained with publicly available examples and tuned with emails from user's inbox. The tuning process however is usually costly and time consuming because the emails in user's inbox are required to be labeled manually. In this paper, a learning-based classifier ensemble is proposed which frees the user from the tedious labeling process. A small amount of labeled tuning examples are first generated by exploiting the distribution differences between spam and legitimate emails and then a semi-supervised learning algorithm is employed to label the user's inbox. Another classifier also trained with publicly available emails is used to generate another set of classification results. Finally, by exploiting the preferences of each classifier in labeling emails, the emails in user's inbox can be classified in high AUC value (Area Under the ROC curve).

1. Introduction

Over the past few years, the proliferation of unsolicited emails, known as spam or junk mails, clutter up emailboxes of millions of people worldwide. Being incredibly cheap and easy, spam causes a lot of trouble to the Internet community. Large amounts of spam emails delay internet traffic and degrade many on-line services. Sorting out the unwanted messages to get legitimate emails takes time and sometime there are cases that normal emails are deleted by mistake. In addition, virus, phishing fraud and pornographic spam emails are harmful to people and cause a lot of losses.

Fighting against spam has being started for years. For example, there is an anti-spam law introduced in the USA [1]. People are also instructed to keep their email addresses carefully and don't response to any spam. There are also resources in the Internet to help people to configure their email servers against spam. Black-lists of spammer are also available in websites [2],[3]. Although these efforts are effective, there are still considerable amounts of spam reach our e-mailboxes.

For ordinary email users, email filtering seems to be an effective way to block spam. Traditional spam filters use rule-based techniques to discriminate spam from normal emails. This approach uses a combination of spammers' email addresses, IP addresses, header information, keywords of the subject line, and even the keywords in email contents to formulate rules identifying spam emails. It works well in initial stage and most spam is blocked. Spammers however find many tricks (e.g. BUUY NOOW, get r*i*c*h) to get through the filter later and the battle between spammers and filters goes forever, just similar to that between virus and virus scanners. Spam filters using this approach as a result are required to update constantly. Machine learning is another approach to filter spam. Instead of specifying a set of rules explicitly, this approach uses a set of classified documents (including both spam and normal emails) to learn the rules implicitly [4]. Combination of both approaches is becoming popular in recent years because it takes the advantages of them, e.g. SpamAssassin [5] and DSPAM filters [6].

With modern machine learning techniques, high accuracy spam filtering is not difficult to achieve. For example, Tretyakov [7] found that SVM [8] and multi-layer perceptron [9] have false positive and false negative values below 5% on PU1 corpus [10] of email messages, Michelakis et al. [11] also reported that their spam filter, a SVM assisted with client rules, has over 90% precision on a seven month live testing. It seems that the problem is not too difficult. Right, it is not difficult if you have "suitable" training data. To train a personalized spam filter, both labeled spam and legitimate emails are required for training. For supervised training, the number of training examples is usually in thousands in order to obtain an accurate classifier. It is not difficult to get spam emails. Every corporate and individuals and even the public domain have a large amount of them. However, it is difficult to get legitimate emails because of the privacy reasons. Without labeled legitimate emails from user's inbox, the

performance of a spam filter will be downgraded drastically. This adaptation is known as the tuning process of spam filter. A simple test running on a dataset [12] (emails are represented by bag-of-words vectors) shows that the accuracy drops from over 90% to about 65% in classification of an user's inbox when training are based on public domain email dataset only and no tuning is performed. It is because there is a discrepancy between the distributions of emails from public domain and individual users' inbox. Even a user provides his emails; the tuning is still a costly and time consuming process because a large amount of human effort is required to label the emails manually.

In this paper, a classifier ensemble is proposed to label a user's emails so that the tuning process can be done in an efficient way. A small amount of labeled tuning examples are first generated by referring to the distribution differences between spam and legitimate emails. Then a semi-supervised learning algorithm [13] is employed to produce one set of labeling results for the emails of the user. There is another set of labeling results coming from classification of the same set of emails by another classifier, a SVM classifier assisted with a Naïve Bayes classifier. By exploiting the preferences of each classifier in classifying emails, the accuracy of email classification can be improved. Through out this paper, ranking is used rather than simple binary labeling, e.g. {-1,1}, unless specified explicitly. An email with higher ranking means that it is more probable to be a spam email. The performance is measured with AUC metric [14] (Area under the ROC curve). A view of AUC value under the spam filtering is that the probability of spam email having higher ranking than a legitimate email if they are drawn randomly from the dataset.

The remainder of this paper is organized as follows. Section 2 discusses the ranking of user's email inbox using support vector machine assisted with Naïve Bayes classifier. Section 3 presents another ranking method using the semi-supervised learning algorithm and the combination of ranking results is also discussed. Section 4 summarizes the testing results on dataset obtained from public domain. Finally, a conclusion and discussion is given in Section 5.

2. Classification of emails using SVM and Naïve Bayes Classifier

.

Support Vector Machine (SVM) is a very popular classifying tool in recent years. The main idea is to map the input data into some much higher dimensional feature space in which data becomes linear separable. The linear decision boundary is drawn in a manner that the margin (minimum distance between training examples to the boundary) is maximized. In case that the mapped data points are in-separable linearly, a cost is included to account for the wrongly classified examples and the margin is maximized together with minimizing the cost. One of the advantages of SVM is the efficient computation of similarity by using the kernel functions and hence classification can be done fast.

As stated in Section 1 that direct application of SVM as personal spam filter may not give accurate classification/ranking because of the distribution differences between public training emails and emails of the user's inbox. Some processing works however is helpful in improving the situation. It is found that turning the bag-of-words vectors representing emails to 0/1 vectors and then normalized the length can improve the SVM performance. To further enhance its performance, a Naïve classifier is proposed to correct the ranking produced by SVM. Construction of the Naïve classifier based on the training dataset is found not helpful because the probability distribution of words is different from that of user's emails. We propose using SVM to classify the user's emails and then use the results to compute the probability of appearance of a word given the class. (i.e. compute $P(w_{ij}|spam_j)$ and $P(w_{ij}|legitimate_j)$ for word i in email j). As there are hundred thousands of words, it is impractical in term of computation and storage to use all the words as attributes. An attribute selector is employed to select best attributes for the Naïve Bayes clssifier. This selector uses the information gain [15] to select attributes that have high frequency of appearance among emails. High frequency of appearance is important because it can reflect the statistical distribution of the words more accurately even the SVM cannot produce accurate classification. The Naïve Bayes clsssifier with this configuration gives very accurate precision value in identifying spam emails, however the recall value is only moderate, about 70% in our testing with the dataset [12]. It is because some spam emails are very distinctive that they contains words are rarely used in legitimate emails. As a result any emails classified as spam by the Naïve Bayes classifier is almost confirmed to be a spam email. These revised results can then combine with another set of results obtained by semi-supervised learning to give higher AUC value.

3. Semi-Supervised Learning with Label Propagation

Let $\{(x_1,y_1) \ldots (x_l,y_l)\}$ be the labeled data, $y \in \{-1,1\}$, and $\{x_{l+1} \ldots x_{l+u}\}$ the unlabeled data. The problem is to label or assign a probability to the unlabeled data such that a cost function is minimized. Let w_{ij} represents the similarity between x_i and x_j. Then a graph where nodes represent the data points, x_i, and edges represent similarity w_{ij} between x_i and x_j can be created. In this graph, the label in x_i can propagate through edges to another node x_j according to a transition probability

$$P_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \quad (1)$$

and the transition of labels of the whole graph is represented by the $(l+u \times l+u)$ dimension matrix P. Define a label matrix Y with dimension $(l+u \times 2)$, whose i'th row has two elements with values between 0 and 1. The first element indicates the probability of the i'th data point is a legitimate email and the second element indicates probability of that point is a spam email, i.e. $y_{i,1} + y_{i,2} = 1.0$. With this configuration, the labels of unlabeled data can be computed, in sense of probability, by using the label propagation algorithm [13] given as follows.

1. Initialize the label matrix Y:
   - If x_i is labeled spam, $y_{i,1}=0$, $y_{i,2}=1$
   - If x_i is labeled legitimate, $y_{i,1}=1$, $y_{i,2}=0$
   - If x_i is unlabeled, randomize $y_{i,1}$, $y_{i,2}$ to small values.
2. Update Y by Computing $Y_{n+1}=PY_n$.
3. Clamp the labels of labeled node to its original values.
4. Repeat 2, and 3 until Y_n converge.

This algorithm propagates the values of labeled nodes to class boundaries according to the distribution of the unlabeled data points. Hence, the unlabeled data can be exploited in learning even there are just a few useful labeled examples. The above configuration in fact minimize the cost function

$$E(y) = \frac{1}{2}\sum_{i,j} w_{ij}(y_i - y_j)^2 \quad . \quad (2)$$

Since it is found that spam emails usually have words and phases that are relatively less used in legitimate emails, appearance of many such words can be an indication of spam emails. Firstly, a dictionary, storing the words or indexes of words, is created with the legitimate emails coming from public training dataset, emails of the user's inbox are check against it and number of words not appearing in the dictionary can be counted. Figure 1a and 1b give an example of the distribution of number of distinct words not appearing in this dictionary for legitimate emails and spam emails respectively (from the dataset [12]). In the figures, the horizontal axis is number of words not appearing in dictionary and the vertical axis is the number of emails. Referring to them, emails with number of distinct words larger than a threshold, say K=60, can be safely classified as spam emails. In practice, since the ground true labels of user's email inbox are not available, the distributions like Fig. 1a and 1b cannot be found. However, they can be approximated with the revised SVM classification results described in Section 2 and hence the threshold value can be determined.

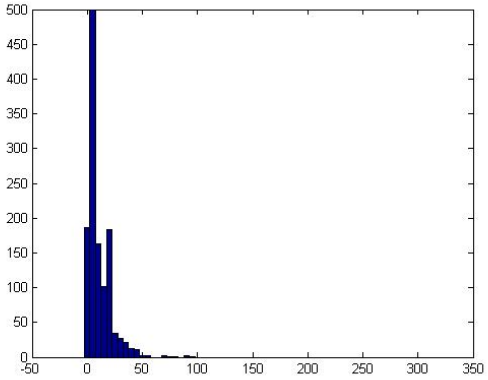


Figure 1a.

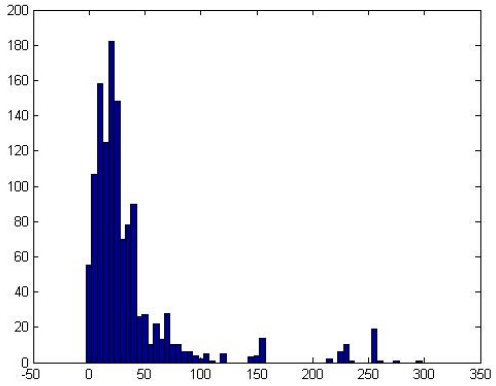


Figure 1b.

In Section 2, it has been discussed that the Naïve Bayes classifier has high precision in identifying spam emails although the recall rate is not high. This feature improves the SVM's performance in identifying spam emails. Combining the results of semi-supervised learning with SVM follow the same thinking, revising the SVM ranking if the classifier is confident on classifying some data points. Studies on the testing dataset show that semi-supervised learning is also good in identifying spam emails, however the added value is subtle to the revised ranking given by SVM and Naïve Bayes classifier. Interestingly, the semi-supervised classification is not good in classifying legitimate emails making some legitimate emails can be identified relatively easily. Fig 2a and 2b show the distribution of the labeling values y_{i1} and y_{i2} where $i > l$ for a dataset from [12]. From the figures, the distribution of legitimate emails is more diverse than spam emails. It is because legitimate emails are more diverse in topics. As a result, a small number of legitimate can be labeled with higher confidence if they spread apart from the center values and hence the SVM ranking can be improved.

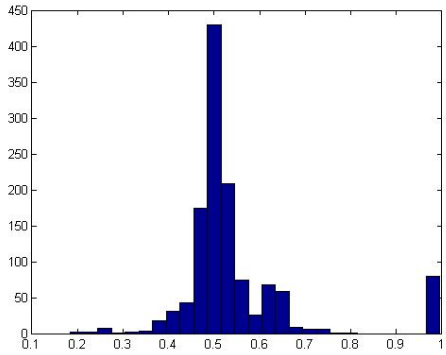


Fig 2a. Distribution of y_{i1}

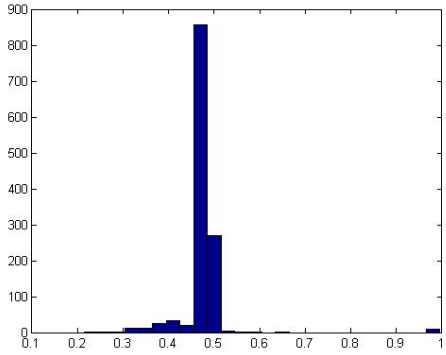


Fig 2b. Distribution of y_{i2}

Similarly to the case of Naïve classifier described in Section 2, the real distributions of y_{i1} and y_{i2} cannot be obtained because emails are not labeled. Nevertheless, they can be approximated by referring to the revised classification results of SVM.

4. Testing Results

The testing dataset in [12] are used in evaluation of the classifier ensemble. In this dataset, there are 4,000 labeled emails coming from public domain and they can be used as training data. Three sets of unlabeled evaluation data, each containing 2,000 emails from three respective different users' email inboxes, are also provided for testing. Ground true of the emails are given for evaluation. All emails in this dataset set are represented by bag-of-words vectors with all the email headers, sender, receiver, and subject lines removed, for the sake of privacy. Words with appearance frequency less than four in the whole dataset are removed as well. It should be noted that the email distributions of public domain is different from that coming from individual users' inboxes. Direct usage of training data to train a classifier to classify unlabeled emails will give unsatisfactory performance. The goal is to rank the emails for each user such that spam emails should have higher ranking than legitimate emails. The correctness of ranking is measured with AUC value. This value can be regarded as the probability that a spam email having higher ranking than a legitimate email if they are drawn randomly from a testing dataset. It has maximum value 1.0 representing that all spam emails are ranked higher than legitimate emails. In this dataset, there are also two additional sets of data, "E" and "F", which contain labeled emails for classifier design purpose. They are not used in our tests. Table 1 summarizes the properties of the dataset, and the evaluation process is described in Table 2.

Dataset		No. of emails (50% spam and 50% legitimate emails)	Labels (+1: spam) or (–1: Legitimate)
A	Training Dataset obtained from public domain	4,000	Labeled
B	Data from User 00	2,000	Unlabeled
C	Data from User 01	2,000	Unlabeled
D	Data from User 02	2,000	Unlabeled
E	Tuning Data obtained from public domain	4,000	Labeled
F	Tuning Data from User 00	2,000	Labeled
G	Ground True Values Of Data "B", "C", and "D"	n.a.	n.a.

Table 1. Data in the testing dataset.

Evaluation Process	
a.	Train and tune the classifier ensemble described in Section 2 and 3 with labeled "A" and unlabeled "B", "C", and "D" .
b.	Classify "B", "C", and "D".
c.	Compute the AUC for the classification by using the ground true values.

Table 2. The evaluation process for the classifier ensemble.

In the computation of the similarity matrix for the semi-supervised algorithm (SL) described in Section 3, cosine similarity is employed. Exponential similarity may give a bit better results sometimes; however it is hard to determine the bandwidth parameter. The results of the evaluation are summarized in Table 3.

Algorithms		AUC values		
		User 00	User 01	User 02
SVM	Use SVM only (no preprocessing)	0.73	0.78	0.89
SVM1	Use SVM only (data preprocessed to 0/1 vectors and normalized)	0.84	0.87	0.94
SVMN	Use SVM + Naïve classifier (data preprocessed to 0/1 vectors and normalized)	0.87	0.91	0.96
SL	Semi-supervised Learning + distinct word count	0.74	0.80	0.86
SVMNSL	**SVMN + SL**	**0.93**	**0.92**	**0.95**
Use majority voting for (SVM1 + Naïve classifier + SL)		0.85	0.9	0.94

Table 3: AUC values for classifying emails of User 00, User 01 and User 02.

From the results shown in Table 3, the emails in three users' inbox can be classified with high accuracy even our training data has distribution different from that of the users. This is because additional information, the properties of spam and legitimate emails and the properties of different classifiers, is exploited. It is worth mentioning that simple majority voting strategy has some improvement in SVM1 but it is still inferior to the SVMNSL. In our study, classification/ ranking of emails of three users' are done independently. This means the relationship and dependence between inboxes is not taken into consideration. Exploiting their relationship is believed having further improvement.

5. Conclusion

Unsolicited bulk emails, known as spam or junk emails, are very annoying to email users. Sometimes virus infected, phishing fraud, and pornographic messages are even harmful, particularly to children. There are many ways to fight against spam emails, one of them is spam filters. Today's spam filter usually makes use of a combination of rule-based approach and machine learning approach to discriminate spam emails, however tuning of spam filter is a costly and time consuming process and sometimes privacy issues are involved. This paper proposes a classifier ensemble to help labeling/ranking of user's emails so that they can be used to tune spam filter.

As the distribution of public domain emails is different from that of emails of individual users, simple supervised learning with common classifiers such as SVM or Naïve Bayes classifier cannot give satisfactory results. The proposed classifier ensemble exploits i) the different properties between spam and legitimate emails and ii) different preferences between classifiers in emails classification. It is found that Naïve Bayes classifier has very high precision in classifying spam emails although the recall rate is only moderate. On the other hand, semi-supervised learning algorithm assisted with distinct words counting is helpful in identifying some legitimate emails. Interestingly, these complementary properties can help SVM classifying user's emails with a high accuracy, in AUC metric. Finally, we conjecture further works can be done such as the effect of using more different kinds of classifiers, the ways of combining their results, and if there are some relationships between individual users, how they can be used properly.

Reference

[1] A.H.M Kamal, "Spammers! Beware", http://www.thedailystar.net/law/2005/02/01/index.htm.

[2] http://www.mail-abuse.org.

[3] http://www.spamcop.net.

[4] Mozilla Spam Filtering, http://www.mozilla.org/mailnews/spam.html.

[5] The Apache SpamAssassin Project, http://spamassassin.apache.org/

[6] The DSPAM Project, http://www.nuclearelephant.com/projects/dspam/ .

[7] K. Tretyakov, "Machine Learning Techniques in Spam Filtering", Data Mining Problem-oriented Seminar, MTAT.03.177, May 2004, pp. 60-79.

[8] Schölkopf B.: Statistical Learning and Kernel Method. MSR-TR 2000-23, Microsoft Research (2000)

[9] Omidvar, O., Dayhoff, J.: *Neural Networks and Pattern Recognition*, Academic Press, 1998.

[10] http://iit.demokritos.gr/skel/i-config/downloads/

[11] E. Michelakis, I. Androutsopoulos, G. Paliouras, G. Sakkis, P. Stamatopoulos, "Filtron: A Learning-Based Anti-Spam Filter", *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[12] Discovery Challenge, ECMLPKDD2006, http://www.ecmlpkdd2006.org/challenge.html .

[13] X. Zhu, *Semi-Supervised Learning with Graphs, Doctoral thesis, CMU-LTI-05-192*, May 2005

[14] A.P. Bradley, "The Use of the Area Under the ROC curve in the Evaluation of Machine Learning Algorithms, *Pattern Recognition*, 30:1145-1159,1997.

[15] T.M. Mictchell, *Machine Learning*, McGraw-Hill Companies, Inc., 1997.

Agent based testbed for relax criteria negotiation

Ng ka-fung

Abstract

Nowadays it is common to see tremendous number of computers are being interconnected to form a grid in order to provide enormous and virtually unlimited computational and storage capacity with arbitrary services available. Efficient negotiation mechanism and strategies would thus provide mutual benefits to both consumer and service providers in the grid, and finally boosting utilization of the grid as a whole. This paper reports experiment results of applying fuzzy logic can be applied in on both side of negotiation agents.

1 Introduction

Computers are often interconnected in large-scale to form a grid to provide enormous and virtually unlimited computational and storage capacity to users. The idea of applying microeconomic in grid then emerges as grid is essentially a resources market which perfectly suit into the traditional microeconomic demand and supply model, and negotiation is central to the idea of economic grid, which in return facilitates automated allocation of resources, a traditionally NP-hard matching problem in combinatorial optimization.

This paper reports the simulation results of incorporating fuzzy logic controller into negotiation kernel of market agents representing and negotiating on behalf of either service providers or service consumers, where the negotiation kernel implementation is based on the Rubinstein's alternating offers protocol [1], and in each offer apart from the initial proposal, they make certain amount of concession by considering the spread between its proposal and the counteroffer based on opportunity, time and competition factors described in [2]. The difficulty lies on the fact that resources negotiation in grid is large-scale multilateral instead of bilateral. The simulation focuses on how fuzzy logic controller helps relaxing stringent negotiation criteria, boosting negotiation success rate without loosing much utility and in addition increasing negotiation speed (reducing number of negotiation rounds required).

This paper is based on [3].

2 Fuzzy Decision Controller

Grid negotiation agents have to deal with wide variety of dynamic market situations to give optimal concession, and the aforementioned opportunity, time and competition (OCT) factors described in [2] are implemented and facilitate agents determining optimal concession to give in each negotiation round. Fuzzy Decision Controller (FDC) is then incorporated into the negotiation kernel to relax stringent concession by OCT. FDC is a prominent choice since agents are facing different levels of negotiation pressures and different market situations, and FDC is a rule-based system to handle all predefined situations.

FDC is deployed on both service consumer and provider agents. It generates a relaxing interval [0, k]. Agents with FDC will consider the counter proposal spread as sufficiently small and acceptable, if the utility difference of the agent proposal and the counter proposal is less then k.

2.1 Consumer FDC

The consumer agent FDC takes two inputs, the failure to success ratio fs_t and demand factor df_t to determine the output, the relaxation degree η, with fs_t [0, ∞] and $df_t, \eta \in$ [0, 1]. As agents are not assumed the knowledge of global grid utilization level, the failure to success ratio serves as a possible indicator of recent grid resources competition and utilization – high failure to success ratio may probably due to high grid utilization or strong competition.

Table I. Consumer Fuzzy Decision Rules

No	If fs_t	And df_t	Then η	No	If fs_t	And df_t	Then η
1	N	N	N	9	M	N	N
2	N	L	N	10	M	L	L
3	N	M	L	11	M	M	M
4	N	H	L	12	M	H	H
5	L	N	N	13	H	N	N
6	L	L	L	14	H	L	M
7	L	M	L	15	H	M	H
8	L	H	M	16	H	H	H

N—Negligible L—Low M—Moderate H—High

As different utilization and competition will put different pressure on negotiating agents to reach agreement with trading partners, the FDC guides agents to relax their trading position according to recent relative demand and negotiation results.

Each consumer agent calculates its own failure to success ratio fs_t at current round t, by taking previous n rounds trading history given as follows:

$$fs_t = \sum_{i=t-n}^{t-1} f_i \Big/ \sum_{i=t-n}^{t-1} s_i$$

where fs_t is the number of request the agent failed to reach a deal with any provider agents before the request deadline is reached, at round i. Similarly s_i is the number of request successfully reached an agreement with provider agent at round i, and n = x / P_m where P_m is the agent's mean event rate, which is the probability of the arrival of a task to a consumer agent in each round (one of the experiment input parameter), and x be the number of negotiation results to consider based on experimental tuning, a typical value ranging from 10 to 20 is used. Hence n controls the number of rounds taken into consideration for recent statistics, such that fs_t reflects only recent failure history.

Each consumer agent computes its demand factor df_t at round t is defined as:

$$df_t = d_t / \max(d_{t-n}, d_{t-n+1}, ..., d_t)$$

where d_i is the total capacity demand at round i, and n in df_t is identically defined as in fs_t , controlling the number of rounds taken into consideration for recent demand statistics, making df_t reflects only the recent demand information.

2.2 Provider FDC

The provider agent FDC also takes two inputs, the current resource utilization level u_t and request factor rf_t to determine the output, the relaxation degree η , with $u_t, rf_t, \eta \in$ [0, 1].

Table II. Provider Fuzzy Decision Rules

No	If u_t	And rf_t	Then η	No	If u_t	And rf_t	Then η
1	N	N	H	9	M	N	M
2	N	L	M	10	M	L	L
3	N	M	M	11	M	M	L
4	N	H	L	12	M	H	N
5	L	N	H	13	H	N	N
6	L	L	M	14	H	L	N
7	L	M	L	15	H	M	N
8	L	H	L	16	H	H	N

N—Negligible L—Low M—Moderate H—High

Utilization level is defined as:

$$u_t = u / p$$

where u is the utilized capacity and p is the total possessed capacity. For the simulated grid environment, agents trade computational resources, thus capacity is measured in million instruction per second (MIPS). In this design of the FDC, provider agents tend to relax more in lower utilization level, in order to obtain agreement utility and avoid resources being idled. On the contrary it relaxes less in low to moderate utilization level, and no relaxation for high utilization, for maximizing utility from users. Agent considers only its own resource utilization level since it has no information of other agents, nor aggregated grid utilization.

Request factor is defined as

$$rf_t = r_t / \max(r_{t-n}, r_{t-n+1}, ..., r_t)$$

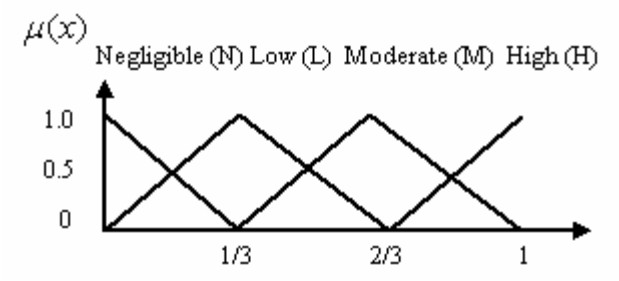
where r_t is the total capacity request at round t. It considers over the previous n rounds, where n = xd . Let $d = \frac{l+u}{2}$ be the average number of rounds in one negotiation session, where l is the lower bound of negotiation deadline, and u is the upper bound of negotiation deadline, so d is provider's mean negotiation deadline for incoming requests. Once a request arrives to provider agent, the capacity demand will persist until reaching a deal or the negotiation deadline, thus mean negotiation deadline is taken as a factor of d as an approximation of number of rounds that a request will persist in provider agent. x is an experimental value typically ranging from 10 to 20, such that rf_t covers recent x negotiation sessions, and reflects the current request level relative to the peak capacity request in previous *n* rounds. rf_t shows the demand relative to the recent peak, and is an indicator of possible future loads in the agent and the grid.

2.3 FDC Internals

The FDC comprises of three components: a fuzzification interface, a fuzzy rule base, and a defuzzification interface.

Fuzzification: Fuzzification interface converts crisp input value into fuzzy representation. Demand factor, utilization level and request factor are evaluated using the percentage membership function defined as below

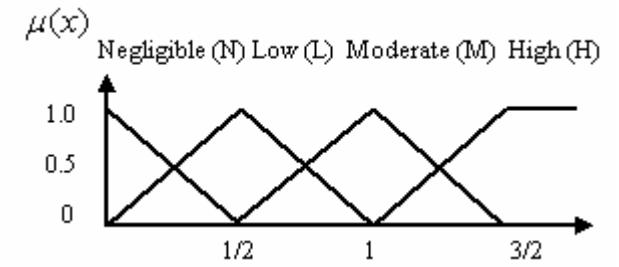
a) Percentage membership function



$$\mu(x) = \begin{cases} -3x + 1 & x \in [0, 1/3] \\ p_1(3x) + (1 - p_1)(-3x + 2), & x \in [0, 2/3] \\ p_2(3x - 1) + (1 - p_2)(-3x + 3), & x \in [1/3, 1] \\ 3x - 2 & x \in [2/3, 1] \end{cases}$$

where $p_1 = 1$ when $x \in [0, 1/3]$, $p_1 = 0$ when $x \in [1/3, 2/3]$
$p_2 = 1$ when $x \in [1/3, 2/3]$, $p_2 = 0$ when $x \in [2/3, 1]$

b) Failure To Success Ratio fs_t membership function



Failure Rate

$$\mu(x) = \begin{cases} -2x + 1, & x \in [0, 1/2] \\ p_1(2x) + (1 - p_1)(-2x + 2), & x \in [0, 1] \\ p_2(2x - 1) + (1 - p_2)(-2x + 3), & x \in [1/2, 3/2] \\ \min(1, 2x - 2) & x \in [1, \infty] \end{cases}$$

where $p_1 = 1$ when $x \in [0, 1/2]$, $p_1 = 0$ when $x \in [1/2, 1]$
$p_2 = 1$ when $x \in [1/2, 1]$, $p_2 = 0$ when $x \in [1, 3/2]$

For example, the fuzzification interface would converts df_t = 0.60 to 20% low and 80% moderate according to the membership function defined.

Fuzzy Rule Base: The rule base is a decision controller which aggregates input into output by proportion fs_t * df_t. For example if fs_t is 10% negligible and 90% low, with df_t is 40% low and 60% moderate, then by rule 2, it is 0.1 * 0.4 = 4% negligible, by rule 7 it is 0.9 * 0.6 = 54% low, and so on. Relaxation degree η is calculated by $\Sigma\ fs_t$ * df_t. In this example, η is determined as 4% negligible and 96% low.

Defuzzification: The defuzzification interface converts the linguistic value of η into crisp value, using *weighted average method* [4], employing also the percentage membership function for defuzzifying relaxation output η.

Whether agent will agree with the proposals utility differences can ultimately be determined from η.

3 Testbed

The grid negotiation testbed consists of five major components, (i) a market generator which generates tasks, resources and represented by respective agents. It assigns attributes to generated objects such as negotiation price set of tasks and agents lifespan, (ii) an agent registry provides and manages agent directory service to active agents in the grid space. It execute instruction from market generator like deploy new agent or terminate active agents on their deadline in the grid space, and delisting those expired from the directory, (iii) transaction repository which records succeeded deals and failed negotiation for statistics purposes, such as recent success to failure ratio, (iv) a messaging gateway buffers agent proposals and route to its trading partners in the next round. Agents avoid direct contact with other agents by placing their offers through the messaging gateway, and (v) the grid agent space.

Resource Provider and Consumer Agents: Consumer agents negotiate with provider agents to lease resources, with the objective of reaching a deal with any of its trading partner before the negotiation deadline is reached. Negotiation starts once a task arrives to the agent, and it assigns the initial offer and send to provider agents. Provider agents negotiate by sending counter-offers to consumer agent proposals. Negotiation continues until either an agreement is reached or the negotiation deadline on either side is reached.

Grid Agent Space: The grid agent space is responsible for simulating the entrance of resource consumer agents and provider agents to the market, with assistant from the agent registry. It provides the negotiation framework to agents such as managing rounds to support alternative offers.

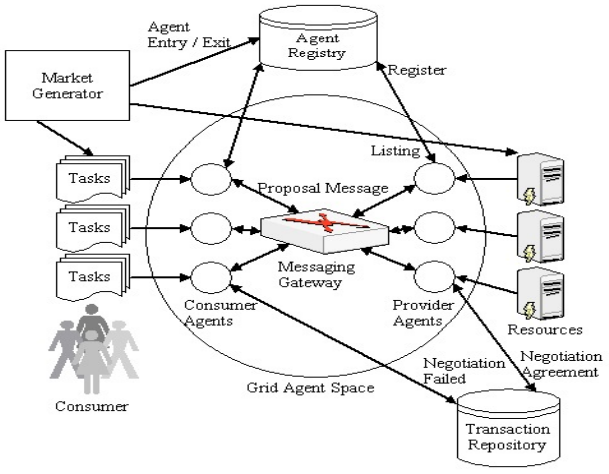


Figure 1. The architecture of the testbed

3.1 Experiment Parameters

The grid controller simulate the negotiation with four input parameters following uniform distribution, (i) market density, the probability an agent will enter the market in all negotiation round, (ii) consumer to provider ratio which determines the probability that determines whether a producer or consumer agent will be generated, (iii) negotiation deadline, the number of rounds a task negotiation cannot exceed deadline, and (iv) the service provider capacity. An additional input, (v) mean event rate, the probability of a task arrival to a consumer agent in each round, following the Poisson distribution. Input (iv) and (v) eventually control the grid utilization.

3.2 Performance Measure

Performance measures are success rate, average utility, expected utility and speed of acquiring resources.

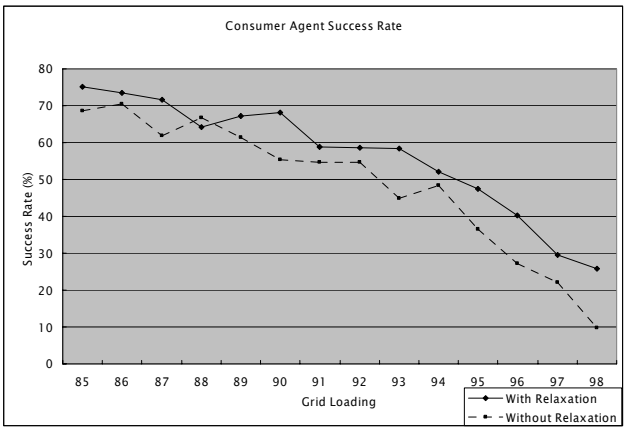
Success Rate	$R_{success} = N_{success} / N_{total}$
Expected Utiltiy	$U_{expected} = (\sum U_{success} + \sum U_{fail}) / N_{total}$ $= \sum U_{success} / N_{total}$
Average Utility	$U_{average} = \sum U_{success}) / N_{success}$
$N_{success}$	Number of tasks reached consensus
N_{total}	Total number of tasks negotiated
$U_{success}$	Average utility of a task that reached consensus
$U_{fail} = 0$	Average utility of a task that no consensus reached

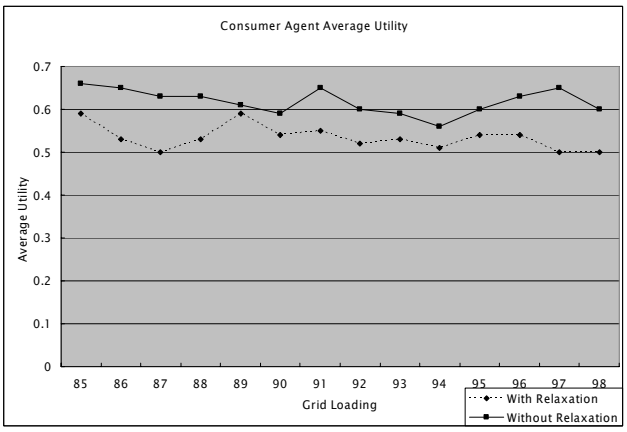
Utility function: Let l_{min} and l_{max} be the initial price and reserve price respectively for tasks in consumer agent, (the reverse for provider agent), and l be the price that consensus is reached by both side, then

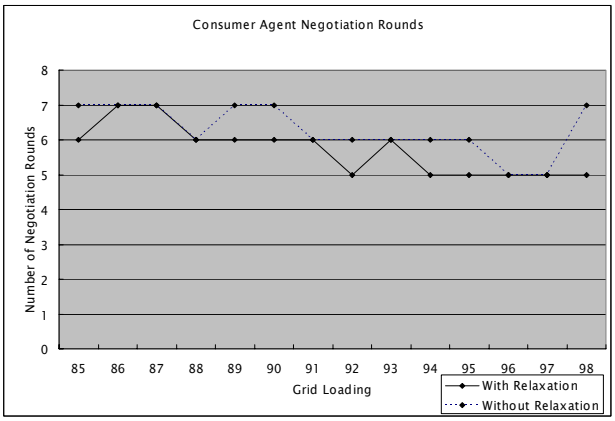
Consumer: $U_{success} = v_{min} + (1 - v_{min}) (l_{max} - l) / (l_{max} - l_{min})$
Provider: $U_{success} = v_{min} + (1 - v_{min}) (l - l_{min}) / (l_{max} - l_{min})$

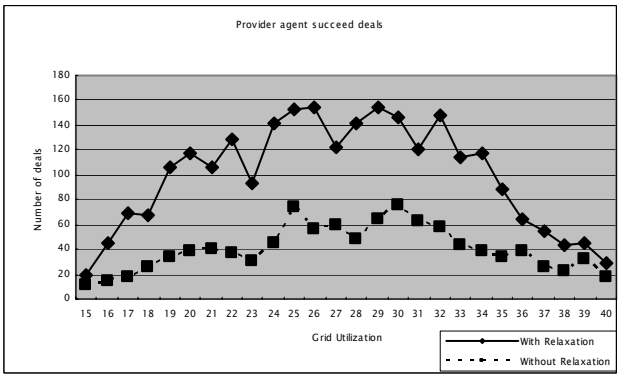
where v_{min} is the minimum utility that agents will get for reaching a deal at reserve price. In this experiment the value of v_{min} is defined to be 0.1. A value that is too close to zero would imply there is little differences with not reaching a deal, where a value too high would make agents making concession easily to aim for success rate.

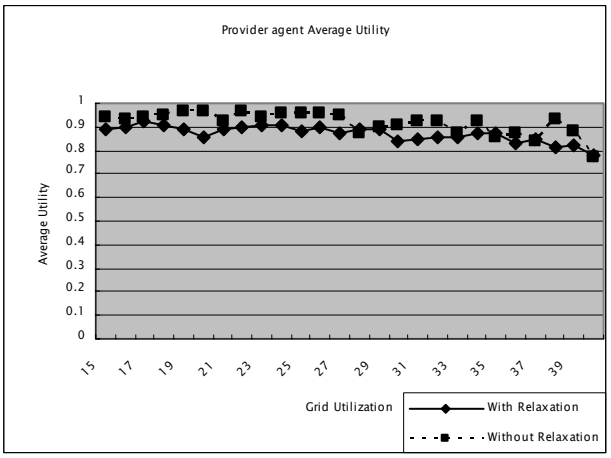
3.3 Simulation Results











4 Conclusion

Although experiment results are still partial in terms of utilization range, preliminary results shows agents with FDC incorporated into negotiation kernel perform better in success rate, utility and speed – the performance measure indicators. Future works will be on optimizing FDC.

References

1. M. Osborne and A. Rubinstein, Bargaining and Markets, The Academic Press, 1990.
2. Kwang Mong Sim and Shi Yu Wang, Flexible Negotiation Agent with Relaxed Decision Rules, IEEE Trans. On SMC Part B, Vol. 34 No. 3, June 2004.
3. K. M.Sim and K. F. Ng. Relaxed-criteria Negotiation for G-commerce. In Proceedings of the Workshop on Business Agents and the Semantic Web (BASeWEB'06), held in conjunction with the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, Hakodate, Japan, pp, 53-61.
4. T. J. Ross, Fuzzy Logic with Engineering Applications. NewYork: Mc-Graw-Hill, 1995

Lightweight Piggybacking for Packet Loss Recovery in Internet Telephony

Wing Yan Chow, Yiu Wing Leung
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
Email: {wychow, ywleung}@comp.hkbu.edu.hk

Abstract

Internet Telephony has been growing in popularity due to its enormous potential. A challenge to Internet Telephony is packet loss, which degrades voice quality. In this study, we propose a packet loss recovery scheme called lightweight piggybacking for Internet telephony. It integrates piggybacking with shared packet loss recovery. Using this scheme, the source telephone gateway applies erasure coding on the low-bit-rate versions of original voice streams, such that the resulting redundant packets (called lightweight redundant pieces) are very small and can be shared by all the original voice streams. To ensure better and more robust performance, we enhance lightweight piggybacking in multipath communication environment. This further minimizes correlated losses of original and redundant packets. Compared with the existing packet loss recovery schemes, the proposed scheme needs a smaller redundancy but achieves a smaller packet loss probability because the small redundancy can be fully utilized via sharing among the original voice streams. We will conduct simulation experiments for performance evaluation.

1. Introduction

Internet telephony has shown a substantial growth in recent years because of its huge potential [1]. Generally, Internet telephony can be classified into three types, i.e. computers to computers, computers to telephones, and telephones to telephones (see Figure 1). In particular, the third type is useful to the general public. It makes use of telephone gateways to bridge local telephone network and the Internet for voice transmission [1-2]. Compared with traditional telephone services, Internet telephony is particularly promising for long-distance calls because of its lower service charge.

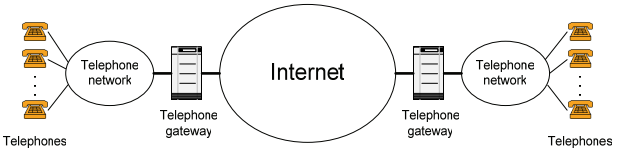


Figure 1. An Internet telephony system based on telephones to telephones

When real-time voice packets are transmitted through the Internet, some packets may be lost (if a voice packet is received after its playout time because of delay, it is equivalent to being lost). As a result, the voice quality is affected. To tackle the packet loss problem, a number of packet loss recovery and concealment methods are proposed in the literature [5-6]. For instance, the most common scheme to combat packet loss is forward error correction (FEC) [5-9]. One simple method is called XOR packet loss recovery [5], which produces redundancies by bitwise XOR operations. In [10], erasure codes are introduced to produce $n - k$ redundant packets from the original k packets at the sender. If the receiver gets any k out of n packets, the lost packets can be recovered. Using this feature, the scheme proposed in [2] enables multiple active voice streams to share redundant packets for packet loss recovery. In this way, any lost packets can be recovered from erasure decoding using original and redundant packets.

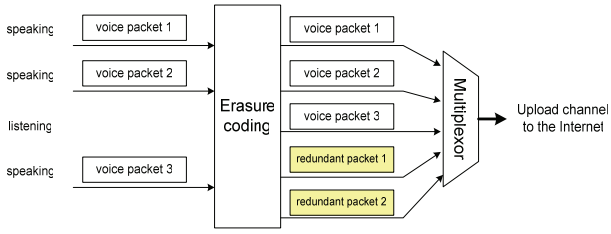


Figure 2. Shared packet loss recovery for multiple active voice streams in Internet Telephony [2]

Piggybacking [11] of redundant data to source packets is a commonly used technique. The source produces low-bit-rate audio packets using low bit-rate redundancy (LBR) [9]. Then they are attached to source packets. When a packet is lost during transmission, it can be replaced by the redundant version.

As an alternative to packet loss recovery, error concealment methods such as interpolation [5-6] and interleaving [5-6] can be used. With interpolation, lost packets can be estimated by neighboring packets. While in the interleaving method, packets are re-sequenced before transmission to minimize consecutive loss during transmission.

Packet loss recovery and error concealment methods can tolerate certain amount of lost packets. But burst loss of packets often occurs when networks get congested and router buffers are full. This significantly degrades the performance of FEC. To minimize correlated loss, path diversity (multipath transport) [7-8, 12-15] has been widely adopted in wired [7-8, 12-13] and wireless [14-15] multimedia communications. Path diversity is a technique that transports data simultaneously through two or more independent network paths in a packet-based network [13]. It takes advantage of uncorrelated loss and delay characteristics of network. By transporting packets over multiple paths, burst losses can be converted into isolated losses. Consequently, FEC can effectively recover the lost packets.

In this paper, we propose a packet loss recovery scheme for Internet telephony called lightweight piggybacking. The key feature of this scheme is to integrate piggybacking with shared packet loss recovery, such that the piggybacked redundant packets can be shared by all the original voice streams for packet loss recovery. We produce lower bit-rate voice packets and uses two stages of erasure coding to produce very small redundant packets. They are then piggybacked to original voice streams during transmission (Figure 3). To further minimize correlated losses of packets and redundancy, lightweight piggybacking is enhanced for multipath communication environment. The proposed scheme not only reduces the total size of redundant packets, but also significantly reduces packet loss probability.

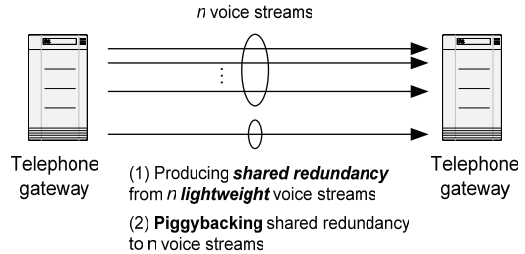


Figure 3. Piggybacking shared redundancy to voice streams

The rest of the paper is organized as follows. In section 2, we focus on the key steps of lightweight piggybacking. The packet loss recovery procedure is discussed in Section 3. Section 4 extends the lightweight piggybacking scheme to multipath communication environment. Section 5 provides a discussion on the effectiveness of the proposed scheme. Finally, section 6 ends the paper with conclusion and future work.

2. Lightweight piggybacking

In this section, we describe the detailed steps involved in lightweight piggybacking. These include forming compressed voice packets, computing redundant voice packets, computing lightweight redundant voice packets and piggybacking.

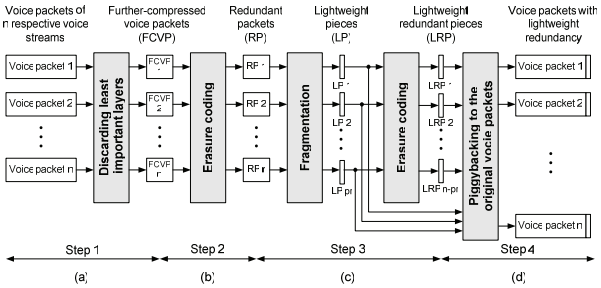


Figure 4. Steps of lightweight piggybacking

Step 1: Discarding least important layers

To make more efficient use of bandwidth, we need to produce compressed voice packets by discarding the least significant bits of voice streams.

In Internet telephony, a telephone session is active [6] if there is voice stream transmission. We denote the number of active sessions as *n*. Hence, there are *n* voice packets in each packetization period [2].

Multilayer coding [17-18] is extensively used in multimedia communications. In the scheme, the base layer only includes the most important bits of a stream. The more enhancement layers the destination receives, the better the voice quality. But the required bandwidth is therefore larger. Based on the idea of multilayer coding, we produce further compressed voice packets (FCVP) by discarding the least important bits of *n* original voice packets (see Figure 4 (a)). Note that the voice packets may have been compressed before. But further compression is required to produce lower bit-rate packets.

Since the compression method only requires simple calculations, it will not significantly delay packet transmission. The source telephone gateway can adjust the compression ratio of voice packets according to bandwidth requirement. It can discard more bits to save

bandwidth, provided that the voice quality is acceptable after compression.

Step 2: Erasure coding

Having produced FCVP, we use erasure coding to compute redundant voice packets.

Erasure coding [10] is adopted in producing redundant packets (RP) from FCVP. The key idea of this coding is to encode k blocks of source data into n blocks of encoded data where $k < n$. After encoding, redundant information is contained in the $n - k$ blocks of encoded data. The source then transmits the encoded blocks to the destination. If the destination can receive any k out of n packets, the lost packets can be recovered by erasure decoding.

As shown in Figure 4 (b), r RP is produced from n FCVP through erasure coding where $r < n$. These RP contain redundant information of FCVP so that they can act as substitutes for original voice packets when there is packet loss. Definitely, when the number of redundant packets increases, the chance of packet loss recovery at the receiver side becomes higher. However, increasing r will expand the required bandwidth for transmission. This is a tradeoff between packet loss recovery and bandwidth.

Step 3: Fragmentation and erasure coding

In this step, we further optimize the size of redundancies by fragmentation.

At the source telephone gateway, the r redundant packets produced in the previous step are fragmented into small pieces. This can greatly reduce the size of redundant packets. As illustrated in Figure 4 (c), each RP is fragmented into p pieces, forming pr lightweight pieces (LP). To achieve better results in packet loss recovery, the total number of pieces pr should not exceed the number of outgoing voice packets n, i.e. $pr < n$.

Now the source needs to produce $n - pr$ lightweight redundant pieces (LRP) by second erasure coding. This will produce a total of n pieces for piggybacking for the next step. Using this method, the destination can recover all LP if it can receive any pr pieces (including LP and LRP). Hence, all the r RP can be reconstructed.

Note that if a RP is fragmented into more pieces, the size of each piece and thus the redundancy is smaller. But this may cause adverse effect in packet loss recovery. Because the receiver needs to receive more pieces correctly in order to recover all LP by erasure decoding.

Step 4: Piggybacking

To attain the goal of packet sharing, we need to piggyback the redundant packets to original voice packets.

Together with the original k LP, the $n - k$ LRP from second erasure coding are piggybacked to the n original voice packets as shown in Figure 4 (d). The lightweight (redundant) pieces, i.e. LP and LRP, are piggybacked to voice packets in sequential order. For instance, the i th piece is piggybacked to the i th packet. Each voice packet now contains a lightweight (redundant) piece. They are shared between packets for packet loss recovery. After piggybacking, the voice packets are then ready for transmission.

3. Packet loss recovery

At the destination, lost packets can be recovered if one of the following requirements is satisfied. The destination receives

1) at least $n - r$ voice packets AND pr LP
2) at least $n - r'$ voice packets AND reassembling pr' LP to form r' RP where $r' < r < n$

When the destination telephone gateway receives voice packets from the source, it can start packet loss recovery according to the following steps:

Step 1: Reproducing redundant packets by erasure decoding and reassembly

When the destination receives any k out of n packets, it extracts the lightweight (redundant) pieces from the received packets, then performs erasure decoding on these pieces to obtain the lightweight pieces, and then reassembles these lightweight pieces to form the redundant packets.

Step 2: Producing further-compressed voice packets

The destination extracts k voice packets from the k received packets and discards their least important layers to produce the further compressed voice packets.

Step 3: Recovering lost packets by erasure decoding

The destination applies erasure decoding on the redundant packets obtained in Step 1 and the further-compressed voice packets obtained in Step 2, so that it recovers the lost packets (further-compressed version).

Detailed steps are illustrated in Figure 5. As shown in the figure, the source produces 6 voice packets. During transmission, voice packets 3 and 4 are lost. Voice packets and lightweight (redundant) pieces are extracted from received packets in the destination. After erasure decoding and reassembly, FCVP 3 and 4 are used to

substitute the lost packets. Although the voice quality produced by FCVP is relatively lower than original voice packets, it is better than inserting silence or noise to replace the lost packets. Moreover, the total size of redundant packets is significantly reduced compared with the existing packet loss recovery schemes.

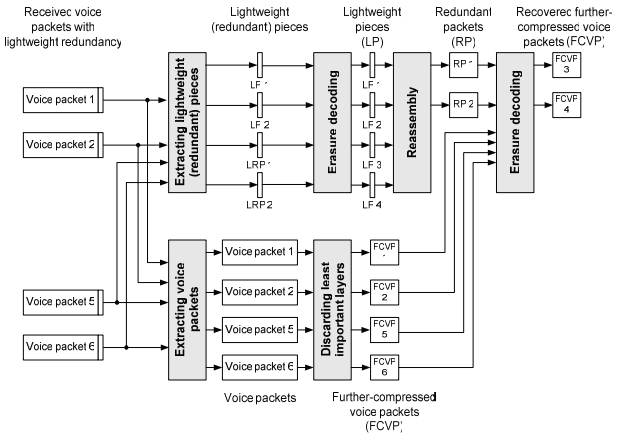


Figure 5. Packet loss recovery of lightweight piggybacking

4. Lightweight piggybacking over multipath

Multipath streaming is a recent hot research topic [13]. Using this approach, a media stream is coded into multiple bitstreams of roughly importance via multiple description coding, and these bitstreams are transmitted over multiple diverse paths in the Internet. This can reduce the adverse effect caused by the variability of individual paths, thereby achieving more stable transmission of the media. In multipath streaming, one of the research issues is how to enforce to transmit the bitstreams over multiple paths [13].

For the Internet telephony system which provides service to multiple cities for good service coverage, we can easily enforce to send the voice streams over multiple paths. Consider the example shown in Figure 6. A source gateway sends a part of the voice streams to an intermediate gateway and then the destination gateway, and ends the other part of the voice streams to another intermediate gateway and then the destination gateway. This enforces to send the voice streams over two diverse paths, thereby achieving more stable transmission of the voice stream.

In the following, we further enhance the lightweight piggybacking scheme such that the resulting scheme has better and more robust performance in multipath communication environment.

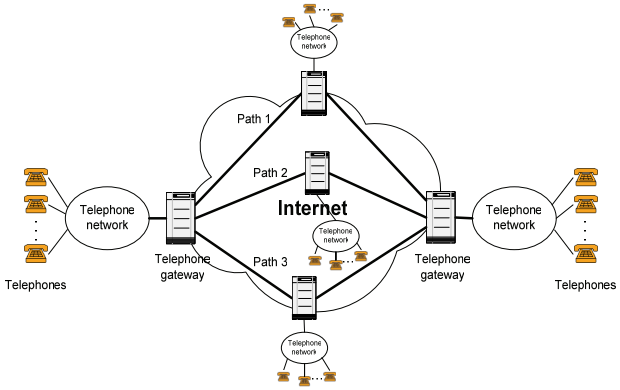


Figure 6. Telephone gateway configuration in Internet Telephony

4.1 Source telephone gateway operation

Suppose there are N_s voice streams at the source telephone gateway. In every packetization period, each stream produces one voice packet. So there are totally N_s voice packets per packetization period. Assume there are N_p disjoint paths from source telephone gateway to destination telephone gateway. The N_s voice packets are divided into N_p groups and distributed evenly on N_p paths. When N_s is not divisible by N_p, i.e. $N_s \bmod N_p \neq 0$, some groups or paths will have one more packet than others. The number of packets (m_x) transmitted on the x th path is calculated as follows [7]:

$$m_x = \left\lfloor \frac{N_s}{N_p} \right\rfloor + c_x, \quad 0 \le x < N_p,$$

$$\text{where} \quad c_x = \begin{cases} 1, & \text{if } x < N_s \bmod N_p \\ 0, & \text{otherwise} \end{cases}$$

Erasure coding is based on matrix operations. Computations will become slow when the number of matrix elements increases. Similarly, if the destination gateway decodes a large number of packets at the same time, it will cause significant delay to users. Therefore, we need to divide the incoming voice streams into groups so that each group can carry out erasure coding and decoding independently. This can make the whole process faster.

Let n_i be the number of packets in the i th group where $0 \le i < N_p$. Generally, the n_i packets of each group are distributed from the first to last path in a round robin manner. As illustrated in Figure 7, the first packet of each group is distributed on each path, followed by the second and later packets. To minimize consecutive loss of packets in the same group, groups of packets are interleaved with each other. This makes neighboring

packets on the same path belonging to different groups and having different sequence numbers. After packet distribution, the number of voice packets for each group on each path will differ by at most 1. Details of the packet transmission algorithm are shown in Figure 8.

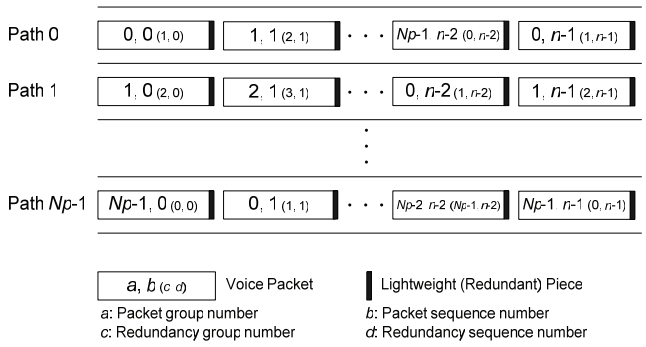


Figure 7. Packet transmission in multiple disjoint paths

Each group produces its redundancies independently. For the i th group, r redundant packets are produced by erasure coding where $r \le n_i$. Following the steps of lightweight piggybacking mentioned in section 2, each redundant packet is fragmented into p pieces where $pr \le n_i$. With second erasure coding, a total of l_i lightweight pieces (LP) can be produced from pr pieces. LP are not piggybacked to the packets of same group. Instead they are piggybacked to packets of previous group. For instance, the j th LP of the i th group is piggybacked to the j th packet of the i-1 th group where $0 \le j < n_{i-1}$. Therefore, the source should produce n_{i-1} LP so that all LP of the i th group can be piggybacked to original voice packets of the i-1 th group.

Using this approach, voice packets and LP of each group can be distributed evenly on each path. This can minimize burst loss of consecutive packets and LP of the same group. The number of lightweight (redundant) pieces to be produced for each group (l_i) is determined by:

$$l_i = n_k, \quad 0 \le i < N_p,$$

$$\text{where} \quad k = \begin{cases} i-1, & \text{if } i-1 \ge 0 \\ N_p - 1, & \text{otherwise} \end{cases}$$

After piggybacking, the packets are transmitted into different paths simultaneously.

/* Determine packets per group */
for $i = 0$ to N_p - 1 **do**
    $n_i \leftarrow N_s / N_p$
end for

remain_packets = N_s mod N_p
if remain_packets > 0 **then**
    $i \leftarrow (N_s - 1) / N_p \bmod N_p$
    while remain_packets > 0 **do**
        $n_i \leftarrow n_i + 1$
        remain_packets ← remain_packets – 1
        $i \leftarrow i + 1$
        if $i = N_p$ **then**
            $i \leftarrow 0$
        end if
    end while
end if

/* Lightweight piggybacking and packet transmission */
for $j = 0$ to $\max(n_i) - 1$ **do**
    $k = j \bmod N_p$
    for $x = 0$ to $N_p - 1$ **do**
        if $j * N_p + x + 1 > N_s$ **then**
            BREAK
        end if
        $i \leftarrow k$
        $k \leftarrow k + 1$
        if $k = N_p$ or $N_s < N_p$ and $k = N_s$ **then**
            $k \leftarrow 0$
        end if
        Piggyback LP(k, j) to Packet(i, j)
        Transport Packet(i, j) into the x th path
        $i \leftarrow k$
    end for
end for

Figure 8. Multipath packet transmission algorithm

4.2 Destination telephone gateway operation

At the destination telephone gateway, packets are received from all disjoint paths. Voice packets and LP are then extracted. Each group carries out packet loss recovery independently. The recovery conditions can be divided into two cases: For each group, the destination should receive:

1) at least $n_i - r$ packets AND pr LP from all disjoint paths

2) at least $n_i - r'$ packets AND reassembling pr' LP to form r' RP where $r' < r < n$

Each group recovers lost packets following the recovery steps mentioned in Section 3.

5. Discussion

In this section, we discuss the effectiveness of lightweight piggybacking over multipath in two cases:

loss of multiple voice packets and consecutive packet loss for the same voice stream.

5.1 Loss of multiple voice packets

For the existing piggybacking scheme, the lost packet from a particular voice stream cannot be recovered when the redundant packet is also lost in the next voice stream. In contrast, lightweight piggybacking can solve this problem by sharing redundancy. Given that the loss of voice packets is less than the FEC protection level, i.e. the amount of redundant packets, lightweight piggybacking can recover any number of lost packets through erasure decoding.

Burst loss, however, will seriously affect the performance of FEC schemes. In view of this, multipath transport is adopted in lightweight piggybacking. This ensures original voice packets and their corresponding redundancy to be transported on different paths, such that they are not lost concurrently when one path is congested. Moreover, as packets are divided into groups and distributed evenly on each independent path, the burst loss probability of packets in the same group is significantly lower. Thus lightweight piggybacking becomes more effective in recovering lost packets. If the destination can get any *pr* out of *n* lightweight (redundant) pieces, all the *r* redundant packets can be reconstructed. Otherwise, the source can still reassemble as many pieces as possible to reproduce redundant packets. Combining *r* redundant packets with *n* - *r* further compressed voice packets from received streams, all the low-bit-rate packets can be recovered by second erasure decoding. They are then used to substitute multiple lost packets.

5.2 Consecutive packet loss for the same voice stream

In the existing piggybacking method, the redundant packet of a voice stream (e.g. stream A) is attached to a particular voice stream (e.g. stream B) in every packetization period. If consecutive packet loss occurs in both stream A and B, the lost packets in stream A cannot be recovered. In contrast, lightweight piggybacking tackles this problem by sharing redundancy.

For each group of voice packets, *r* redundant packets are produced. They are not specific to any of the original packets. Instead, they can be shared by all the voice streams for the same group. Using the above example, even though both stream A and B are lost consecutively, the destination can still recover both streams given that any *pr* out of *n* lightweight (redundant) pieces are received. Therefore, lightweight piggybacking facilitates packet loss recovery for consecutive packet loss of the same voice stream.

6. Conclusion and future work

In this paper, we propose a packet loss recovery scheme for Internet telephony, called lightweight piggybacking. Compared with existing packet loss recovery schemes, it significantly reduces the size of redundant packets and packet loss probability by sharing of redundancy. To ensure better and more robust performance, we enhance lightweight piggybacking in multipath communication environment.

In the near future, we will conduct simulation experiments to evaluate the performance and effectiveness of the proposed scheme.

7. References

[1] "Special Issue on Internet Telephony", *IEEE Communications Magazine*, vol. 38, no. 4, April 2000.

[2] Y. W. Leung, "Shared Packet Loss Recovery for Internet Telephony", *IEEE Communications Letters*, vol. 9, no. 1, January 2005, pp. 84-86.

[3] C. Shim, L. Xie, B. Zhang, and C.J. Sloane, "How Delay and Packet Loss Impact Voice Quality in VoIP", *Qovia, Inc. White Papers*, December 2003, pp. 1-10.

[4] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failures on VoIP performance", *Proceedings of ACM NOSSDAV*, 2002, pp. 63-71.

[5] C. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio", *IEEE Network*, September/October 1998, pp. 40-48.

[6] M. J. Lipka, "An analysis of error handling techniques in voice over IP", *Proceedings of 21st Computer Science Seminar*, 2005.

[7] X. Yu, J. W. Modestino, and I. V. Bajic, "Modeling and analysis of multipath video transport over lossy networks using packet-level FEC", *Proceedings of The Eleventh International Conference on Distributed Multimedia Systems*, 2005.

[8] T. Nguyen and A.Zakhor, "Path diversity with forward error correction (PDF) system for packet switched networks", *Proceedings of INFOCOM*, 2003.

[9] W. Jiang and H. Schulzrinne, "Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss", *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Miami Beach, Florida, May 2002.

[10] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", *ACM Computer Communication Review*, vol. 27, April 1997, pp. 24-36.

[11] J. F. Kurose and K.W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, Chapter 6, 2nd ed., Addison Wesley, 2003.

[12] S. Tao, K. Xu, A. Estepa, and T. Fei, "Improving VoIP quality through path switching", *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[13] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming", *IEEE Communications Magazine*, August 2004, pp. 80-87.

[14] S. Mao et al, “Multipath video transport over ad hoc networks”, *IEEE Wireless Communications*, August 2005, pp. 42-49.
[15] S. C. Draper and M. D.Trott, “Costs and benefits of fading for streaming media over wireless”, *IEEE Network*, March/April 2006, pp. 28-33.
[16] G. Scheets, M. Parperis, and R. Singh, “Voice over the Internet: A Tutorial Discussing Problems and Solutions Associated with Alternative Transport”, *IEEE Communications Surveys*, vol. 6, no. 2, April 2004, pp. 22-31.
[17] S. McCanne, M. Vetterli, and V. Jacobson, “Low-Complexity Video Coding for Receiver-Driven Layered Multicast,” *IEEE JSAC*, vol. 16, no. 6, August 1997, pp. 983-1001.
[18] B. Girod, K. W. Stuhlmüller, M. Link and U. Horn, “Packet Loss Resilient Internet Video Streaming”, *Proceedings of SPIE Visual Communications and Image Processing '99*, vol. 3653, January 1999, pp. 833-844.

The Client-based Framework For Privacy-Preserving Location-based Data Access

Jing DU, Jianliang XU

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

Abstract

Recent years along with blooming of mobile and wireless networks and positioning technologies, LBS(location-based services) have been gradually becoming more and more valuable and important applications. At the same time the privacy issues raised by such applications have also grasped more and more attention and researchers have put efforts and made great progress to address them. In this paper, inspired by previous excellent achievements, we propose a novel client-based framework to facilitate privacy-preserving location-based data access in the context of LBS and currently the handling of NN(nearest neighbor) queries is focused on. We present the main idea of such a framework first. Then we describe details about algorithms and techniques geared in the most crucial components of the framework, i.e., mobile client devices. Furthermore we manage to improve classic RNN(range nearest neighbor) query handling algorithm and finally conclude our paper with directions for future work.

1 Introduction

We consider a client-server architecture, where clients are mobile and equipped with wireless interface to communicate with the server. We assume clients are location-aware - they can position their own locations (e.g., using GPS or WLAN based positioning) or obtain their locations from a trusted location server. Clients are interested in querying information related to their current locations with location privacy retained. General speaking, existing middleware-based solutions[8, 9, 10, 11] realize this through privacy policy checking or de-personalization which is executed at a middleware server. But these approaches heavily rely on the trust in middleware service provider. Furthermore, the middleware must seek for the client's consent before use of location data in accordance with laws, which complicates the system administration. Therefore it's meaningful to explore the feasibility of client-based privacy-preserving approaches and we propose a novel client-based framework in which the needs of trust in service provider and consent from client are relieved. In this paper, we will focus on handling kNN queries, an important kind of location-based queries.

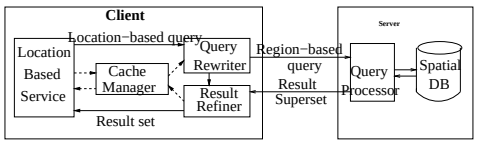


Figure 1. Proposed client-based framework for privacy-preserving location-based data access

Illustrated by **Figure 1**, the proposed framework employs a query rewriter in the client to transform a location-based query to a region-based query such that the resolution of current location is reduced before it leaves the client, thereby protecting location privacy. Based on received region-based query, the server evaluates a result superset, which is returned to the client. Finally, the actual result set is computed by a result refiner in the client. Meanwhile, a cache manager is optional to improve the system performance.

Figure 2a gives such an example scenario. Instead of providing current location q and the query of finding the nearest object, the client submits an uncertain region r_q and the query to the server. The server then returns the set of objects that are potentially a nearest object of some point in r_q , i.e.,{b,c,d}. At last, the client uses the exact location q to find out the actual nearest object, i.e., c.

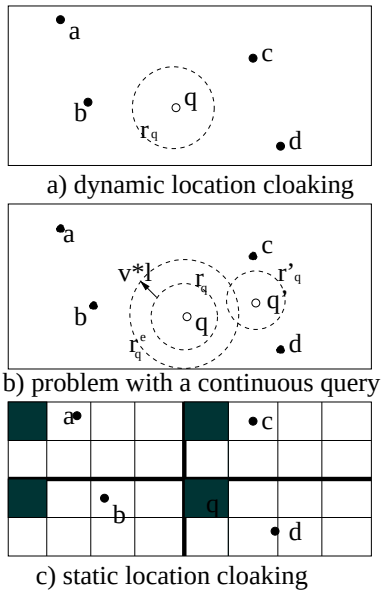


Figure 2. An Illustrative Example

The basic idea seems very straightforward. But it's actually very challenging to effectively transform queries to cloak the precise locations of clients in such a framework.

2 Related work

Location Privacy Protection. Recently the research focusing on privacy-preserving data access has been attracting more and more attention[1, 2, 3, 4, 5, 6, 7].

Although little research has been worked on privacy issues for location-based information access applications, privacy concerns for object tracking applications have been addressed in the literature. A typical solution is to employ a middleware residing in between moving clients and service providers. The middleware collects location information from clients and controls application access to location data. There are two categories of approaches for access control.

The first one makes use of privacy policies to control data access. According to the system architecture(**Figure 3**) proposed by [8], a location server works on top of positioning technologies such as GPS, receiving requests for users' location information with associated privacy statements from various applications. And system components called validators determine whether or not to grant the applications' requests through checking privacy statements from client applications against users' or system administrators' privacy preferences.In [9], a more elaborated access control model utilizes a new data structure called ASM-trie to support the granular representation of moving objects and customer profiles and efficient enforcement mechanism.

An alternative approach uses anonymity through de-

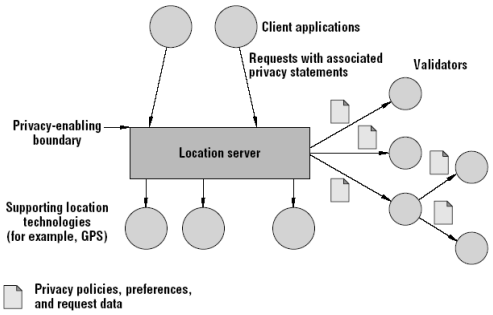


Figure 3. A middleware-based location privacy preserving system

personalization of data before its release. [10] introduces the concept of location k-anonymity in the context of location-based services(LBSs) and provide anonymity by adjusting the resolution based on the density of users in a region through the quadtree-based algorithm. Besides, it identifies a series of privacy threats through location information. But there are still some problems which don't get effectively solved. For example, when requests for multiple objects are issued, an adversary may gain information from tuples overlapping in time and space through some kind of inference; moreover, sophisticated adversaries may mount an identification attack if they can link multiple requests to the same subject and can repeatedly obtain the subject's location information from other sources. Based on the work by [10], [11] extends it to a customizable k-anonymity model by proposing a clique-based spatio-temporal cloaking algorithm. It makes use of the concept of 'clique' from graph theory to find out those objects which are within each other's cloaking boxes and the k-anonymity requirements of which are met to form an anonymity set. [12] updates pseudonyms associated with users based on a concept of mix zones to prevent the service providers to infer the tracks of users. A mix zone for a group of users is defined as a connected spatial region of maximum size in which none of these users has registered any application callback. The middleware system responsible for the anonymizing job should define the mix zones a priori or calculate them separately for each group of users as the spatial areas currently not in any application zone and estimate the anonymity level according to history data and statistical analysis. Users could determine whether to register the service by comparing the anonymity level with their own anonymity demands.

However, both access control and anonymity mechanisms offer little protection when the middleware is owned by an untrusted party. Private data has been inadvertently disclosed over the Internet in the past. Different from the above middleware solutions, this project proposes a client-based framework to support location-based information ac-

cess with location privacy protected.

Spatial Query Processing. There is a large body of research work on spatio-temporal query processing. Early works assumed a static dataset and focused on efficient access methods (e.g., R-tree[13]) and query evaluation algorithms[14]. Recently, a lot of attention has been paid to moving-object databases, where data objects or queries (or both) move. Saltenis et al. [15] proposed the Time-Parameterized R-tree (TPR-tree) for indexing linearly moving objects. Tao et al. [16] optimized the performance of the TPR-tree and extended it to the TPR*-tree. Continuous spatial queries have been investigated by assuming known movement trajectories (e.g., [17]) or without knowing object movement patterns (e.g., [18]). Cheng et al. developed algorithms for evaluating probabilistic queries over imprecise object locations when the querying location is accurate [19] or imprecise [20]. Distributed approaches for monitoring continuous range queries were investigated in [21]. The main idea is to shift some load from the server to the mobile clients. Spatial queries over road networks have also been studied recently (e.g., [22]). A new type of location-based spatial queries, i.e., nearest surrounder search, was introduced and studied in [23].

While all previous works studied point-based k-nearest-neighbor (kNN) search, only a recent work by Hu and Lee investigated region-based kNN search with respect to a query rectangle [24]. It defined 'range nearest neighbor query' as follows:

Given a dataset in the d-dimensional space R^d, the set of range nearest neighbors (RNN) for a hyperrectangle $\Omega \subset R^d$ (boundary inclusive), denoted as RNN(Ω), is defined as the set of the nearest neighbors (NN) for all points in Ω. That is, RNN(Ω) = $\{$NN(p) | p $\in \Omega\}$, where NN(p) denotes point p's nearest neighbor. Ω is called the query range. By definition, it is a d-dimensional hyperrectangle. This definition could also be extended to define the k-range-nearest-neighbor (kRNN) query. kRNN(Ω) is the set of kNNs for all points in Ω, that is, kRNN(Ω) = $\{$kNN(p) | p $\in \Omega\}$.

According to [24], there are two important observations: 1) all objects within the region should be returned since they are nearest objects to themselves; 2) all potential nearest objects outside the region must be nearest objects to some point on the region boundary. Thus, a region-based query can be decomposed into a range query and a kNN query with respect to the region boundary. When the boundary is composed of several line segments, finding out kNN of the boundary is equivalent to separately finding out kNN of those line segments. And this problem has been gently settled by Tao et al[17]. In our framework, these achievements have been absorbed to construct the server components of handling kRNN queries.

To address energy issues, novel index structures have been developed to answer location-point queries [25] and nearest neighbor queries [26] in wireless broadcast systems. These indexing techniques have also been extended to mobile point-to-point systems [27, 26].Furthermore, an overview of data management issues for LBS applications was presented in [28]; and a tutorial on data management techniques in LBS was taught at ICDE '04 [29]. Monitoring of continuous spatial queries based on a notion of safe region was studied in [30].

3 Robust Static Location Cloaking

3.1 The Problem of Naive Location Cloaking

How should we transform a location-based query before it is sent to the server? In essence, the problem is how to cloak the current location by an uncertain region. In this paper we assume the area of uncertain region is used to specify the privacy requirement. For example, a user can specify it's acceptable to be located within an area of 1 square mile when she is in a shopping mall or within an area of 10 square miles when she is in the Disneyland. There are two optional cloaking methods:

- Dynamic Cloaking. As described in the 'Introduction' section, a random uncertain region is dynamically generated based on current location, e.g., r_q in **Figure 2a**.

- Static Cloaking. The service area is pre-partitioned into a set of grid cells, each of which is further divided into a number of sub-cells. As shown in **Figure 2c**, the service area is partitioned into 4 cells with each consisting of 8 sub-cells. Given current location q, the sub-cell covering q and the corresponding sub-cells,i.e.,dummy cells, in other cells, constitute the (discrete) uncertain region (i.e., shaded sub-cells in **Figure 2c**).

The idea behind two methods is somewhat straightforward, so we call them Naive Location Cloaking. However, both don't work well for continuous queries. Consider the example shown in **Figure 2b**. Suppose the client issues another query at location q' with an uncertain region r'_q . If the system knows the maximum moving speed of the object , v, it can draw an ever-expanding region ($r^e{}_q$) of the previous region r_q based on v and the elapsed time l since the last query. Thus, one can figure out that the client must reside in the intersection region of $r^e{}_q$ and r'_q , which is smaller than the expected area for privacy requirement if $r'_q \not\subset r^e{}_q$. The same problem still exists when adopting static cloaking method.

3.2 The Algorithm of Robust Static Location Cloaking

To solve the problem described above, we improve the approaches of Naive Static Location Cloaking and propose an algorithm named Robust Static Location Cloaking.

When adopting static cloaking method, as mentioned, for each cloaking cell, the server could still compute the intersection of ever-expanding region of the previous query region and current query region and point out that the client could reside in it; therefore the user's cloaking requirement could not be met. So the idea hit us that we could always pick up the cells lying in the expanding region. Thus even if the server gets the intersection areas, the total size of them will not be smaller than the user requirement.

For those expanding areas from dummy cells, this method is OK because we could choose any cells within them. But there is a problem for the real cell in which the real location lies. Sometimes the real cell may not completely lie in the expanding region and just overlap it. The intersection area could be small and it's possible that the user requirement could not be met. For example, the cell size is 1 unit and the user requires 4.5 units as cloaking area. In this case, we give 5 cells as cloaking cells. But if the intersection of the 'real' cell and the expanding region is smaller than 0.5 units, the user requirement could not be met. Therefore we plan to add one extra cell to compensate. Thus, in the example above, we will use $\lceil 4.5/1 \rceil + 1 = 6$ cells as cloaking cells instead of 5.

So far it seems that the problem has been settled gently. But is it really safe and robust? Negative. If the server is smart enough, it can notice that there is always a fixed place occasionally sending out cells overlapping the expanding region while other places always send cells lying within the expanding regions. Furthermore assume that the algorithm of the client is known to the server, the real cell could be easily discovered. In order to get rid of such a embarrassing situation, we propose a trick again. When the real cell overlaps the expanding region, we just randomly pick up dummy cells within their own expanding region; when the real cell lies within its expanding region, we randomly select an expanding region from the remaining ones and randomly fetch one cell overlapping it to cover the real cell. Such a cell is called fake cell contrasting against the real cell. Thus in the eyes of the server, it's no longer able to find such a strange position just mentioned.

At this point, the main idea has been presented. Unfortunately the client still has to be faced with some problems. That is, sometimes, the client may not be able to give suitable cloaking cells.

Consider the following situation. Shortly after the client sends out a query wherein the real cell totally lies in its corresponding expanding region and there is a fake cell overlapping its corresponding expanding region, the client wants to issue another query. This time, the real cell has moved to the boundary of expanding region and overlapped it. According to the original plan, the client should find out a cell completely lying within its corresponding expanding region from each expanding region. But due to the short interval between two queries, the intersection of last fake cell and its corresponding expanding region, i.e., the actual cloaking area, haven't expanded sufficiently to contain a cell. In this case, we let the client not send any query and just use the last returned superset cached in the client to approximately evaluate the kNN of real location submitted by the external user.

There is another case. The last real cell didn't overlap its corresponding expanding region and the current real cell doesn't either. And the interval between two queries is also short. According to the original plan, we should pick up a cell as the fake cell to cover the real cell. But similar to the case above, the intersection of last fake cell and its corresponding expanding region, i.e., the actual cloaking area, haven't expanded sufficiently to contain a cell. We have to continue to select the fake cell from this place. Theoretically this is OK; but when implementing this algorithm we found that, if this case occurs continuously, the client's efficiency will be lowered considerably. The reason will be presented later. Therefore in this case, we adopt the same method, i.e., using the last cached superset to do an approximate processing.

The following is the details of complete algorithms. First of all, we assume that the partition of the service area is fixed. We also assume that once the user specifies the cloaking requirement at the very beginning, she will use it for next continuous queries.

Algorithm 1 is the algorithm processing the first query issued by the external user.

Algorithm 1 Robust Static Cloaking (issuing the first query)

1: Compute the real cell
2: Compute the total number of cloaking cells by the formula N =$\lceil S_r/S_c \rceil$ +1, where S_r is the cloaking requirement specified by the external user and Sc is the cell size
3: Randomly select cloaking cells in the service area
4: Send out the query with cloaking cells
5: Get the result superset and refine it to get the final answer
6: Keep the history records about which cells are selected and responding timestamps

And Algorithm 2 is the algorithm handling the general situation.

Next we want to talk more about the implementation and

Algorithm 2 Robust Static Cloaking (general situation)

```
1: Compute the real cell
2: compute all expanding region according to the history records
3: if the last real cell overlapped its corresponding expanding region then
4:   if this time the real cell overlaps its corresponding expanding region then
5:     Randomly pick up a cell completely lying within its corresponding expanding region from each expanding region except the real cell's corresponding one
6:   else
7:     Randomly select an expanding region and randomly find a cell overlapping it as the fake cell; as to the remaining expanding regions except the real cell's corresponding one, randomly pick up a cell completely lying within its corresponding expanding region from each.
8:   end if
9: else
10:  if this time the real cell overlaps its corresponding expanding region then
11:    if the last fake cell's corresponding expanding region could contain a cell then
12:      Randomly pick up a cell completely lying within its corresponding expanding region from each expanding region except the real cell's corresponding one
13:    else
14:      Approximately process the query using the cached last returned superset and exit
15:    end if
16:  else
17:    if the last fake cell's corresponding expanding region could contain a cell then
18:      Randomly select an expanding region and randomly find a cell overlapping it as the fake cell; as to the remaining expanding regions except the real cell's corresponding one, randomly pick up a cell completely lying within its corresponding expanding region from each
19:    else
20:      Approximately process the query using the cached last returned superset and exit
21:    end if
22:  end if
23: end if
24: Send out the query with cloaking cells
25: Get the result superset and refine it to get the final answer
26: Update the history records
27: Keep the information whether the real cell overlaps its corresponding region
28: Keep the information which is the fake cell this time
29: Cache the superset
```

usage of 'expanding regions'. For each cloaking cell to be computed, we have kept a list of history records, which contains locations of cells which are ever selected as the cloaking cells and the corresponding timestamps. Then we could get the differences between the current query's timestamp and each history timestamp and using that we could get a series of round corner rectangles. The corner radius of each equals to the product of the maximum moving speed of the external user and the corresponding time difference. So in fact we are using the intersection of all round corner rectangles to represent the actual 'expanding regions' and the real location must lie in it. After that, if we want to find out a cloaking cell completely lying within the expanding region, we could find out the set of all cells lying within each round corner rectangle and get the intersection of these sets. And then we just randomly pick up one from then intersection. If a fake cell is needed, we arbitrarily start from a round corner rectangle and find out all of its overlapping cells; keep the ones each of which can overlap or be contained by all other round corner rectangles and drop others. Do the same thing to other round corner rectangles. Finally we could get a set from each round corner rectangle. The fake cell could be got by computing the union of these sets and randomly picking up one from the result.

So far we could explain why we execute the 13th step in the algorithm described above. Just think about such a scenario: the last real cell didn't overlap its corresponding expanding region and the current real cell doesn't either. And the interval between two queries is also short. At this time, we need to a fake cell. Due to the short interval, it's very possible for the intersection between the last 'fake' cell and its corresponding expanding region don't have enough time to grow to contain a cell. That is to say, we can only select this position to give out the fake cell. There is a problem with that. If this case occurs continuously with short interval, we have to always choose this position to give out the fake cell because the expanding region always don't have enough time to grow to contain a cell. As a result, this position has a longer and longer list of history records and therefore that means we used more and more round corner rectangles to represent the expanding region. Every time this case occurs, we must validate whether this expanding region contains an available cell within it first. If not we have to fetch a fake cell from the long list of round corner rectangles with cost about O(N^2). With the list becoming longer and longer, the performance of the client will drop sharply. In order to avoid this awful situation, no doubt that adopting the method described in the algorithm is a wise trade-off.

Figure 4. The spatial dataset used in experiments

3.3 Experiments

How to partition the service area is also an issue. We have designed a simulation to find out the optimal partition theme through experiments. Meanwhile we could also get the data reflecting how the approximate processing adopting in the algorithm affects the exact results of queries.

The server is an R-tree-based spatial database [13] enhanced with the capabilities to handle kRNN queries, where the algorithms proposed by Hu and Lee [24] are directly implemented by us. And the server runs on a Linux machine (Red Hat Linux release 7.3). The dataset we use is the coordinate set of meaningful places of California(**Figure 4**), which has about 2,000,000 elements and has been normalized to a 1.0*1.0 square.

The client runs on a pocket PC emulator supplied by the MS Visual Studio 2003 and communicates with the server through a High-Speed LAN.

First of all the query point sequence is generated as follows. We first generate the moving trail following the 'random walk' mobility model. In random walk model, an object selects a speed randomly from a configured range and an orientation and then move during a fixed time interval; upon arrival it repeats the behavior just described.

And then we work out the query points from the trail according to the exponential distribution.

In order to find the optimal partition, we have prepared a couple of groups of settings with different partition themes and k values. For each group, we issue the queries in sequence with one cloaking requirement which is within the range 0.000001 $\sim$ 0.000050; and then repeat with another different cloaking requirement which is also within the range and then again and again. And then for each group we record the average superset size, the percentage of the queries which are processed approximately, the percentage of the queries which are returned incorrect answers amongst the approximately processed queries and the average relative error, i.e. the approximate NN's distance to the precise NN's distance.

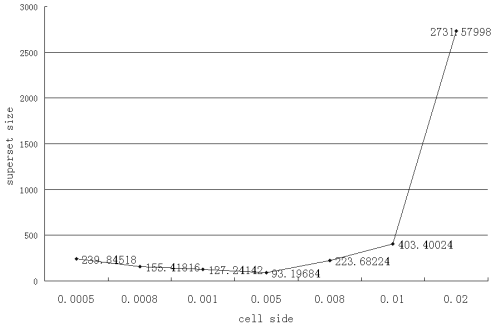


Figure 5. Superset size (k = 1)

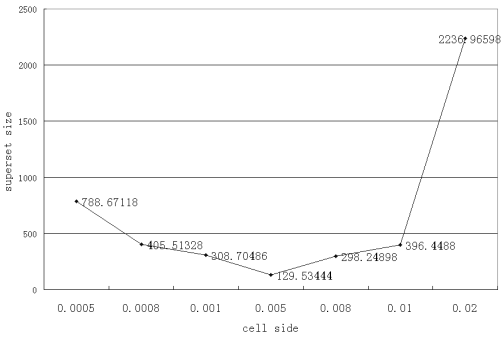


Figure 6. Superset size (k = 5)

Figure 5&6 is drawn according to the experiment result.

We could draw a conclusion from the experiment result that when the cell side equals to 0.005, the average superset size is the minimum and therefore the energy used to download and refine the superset could be saved when the client runs on a Pocket PC or other kinds of mobile equipments.

And **Figure 7** $\sim$ **12** reflect the experiment result concerning the approximate processing technique adopted in the algorithms.Wherein **Figure 7&8** is about the percentage of queries handled approximately. **Figure 9&10** is about the average accuracy of answers to the queries, which is calculated by $\frac{approximate\_NN\_distance}{precise\_NN\_distance}$ and should be always larger than 1. And **Figure 11&12** is about the percentage of inaccurate answers.

As we can see, although the percentage of queries approximately processed is relatively high, the percentage of the queries which are returned incorrect answers amongst the approximately processed queries is low, and the relative error is very small, almost 0. So the approximate processing merely affects the result very slightly.

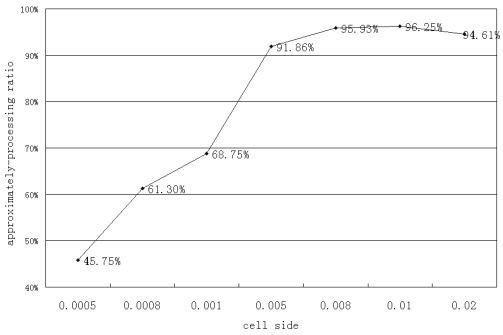


Figure 7. Approximately processing ratio (k = 1)

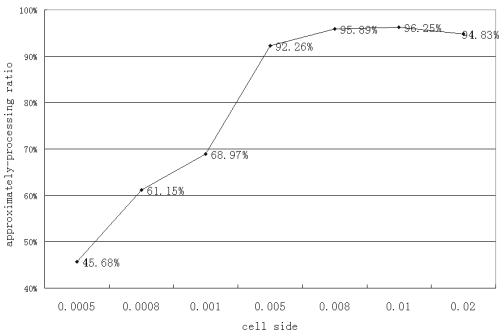


Figure 8. Approximately processing ratio (k = 5)

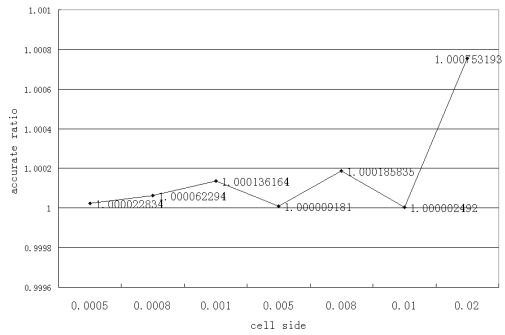


Figure 9. Accurate ratio (k = 1)

4 Accelerating Rectangle-shaped RNN Search

As mentioned above, in our framework, we equipped the server with the capability of handling the rectangle-based region nearest neighbor queries and the algorithms proposed by Hu et al[24] was adopted. According to them, the server handles the queries as follows. It first finds out all of nearest neighbors of the boundary of rectangle submitted by client, and then finds out all data points inside the rectangle, i.e., handles the range query, and finally applies a union

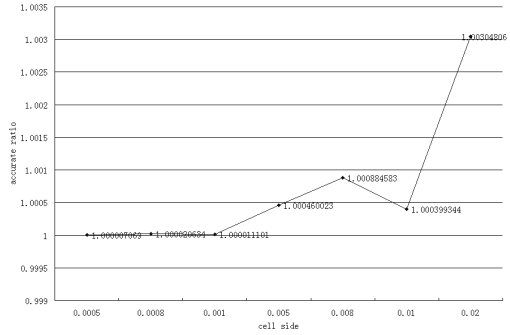


Figure 10. Accurate ratio (k = 5)

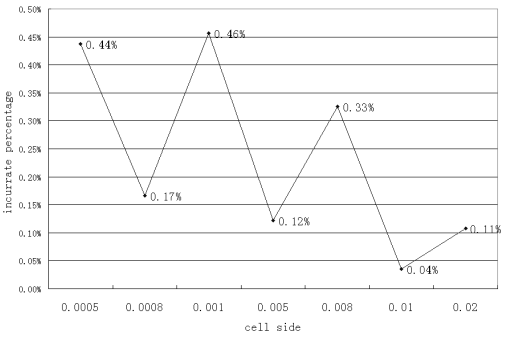


Figure 11. Inaccurate percentage (k = 1)

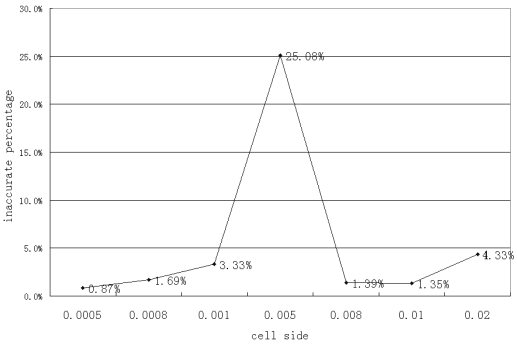


Figure 12. Inaccurate percentage (k = 5)

operation upon the result sets out of the two operations.

Obviously both of operations require visiting the R-tree and we found that both may visit the same node of R-tree and therefore result in duplicate disk I/O. Based on the analysis above, we suggest that the range query could be processed first; for the fetched data points we could apply in-memory processing algorithm of RNN algorithms to judge which are the nearest neighbors of rectangle boundary. After that the RNN algorithms are applies. When encountering the R-tree node which needs to be fetched out via disk visiting according to the RNN algorithms, we check whether all data points contained within it have been processed in the phase of handling range queries. If yes, the require disk I/O is avoided and the node and its children

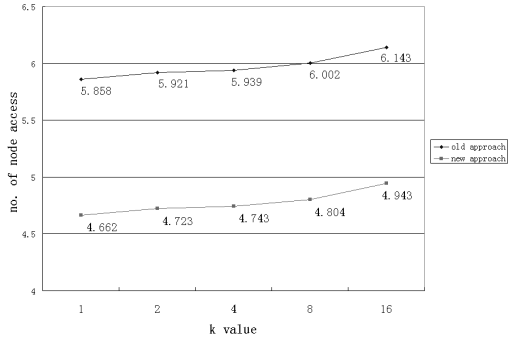


Figure 13. Node access (cell side = 0.001)

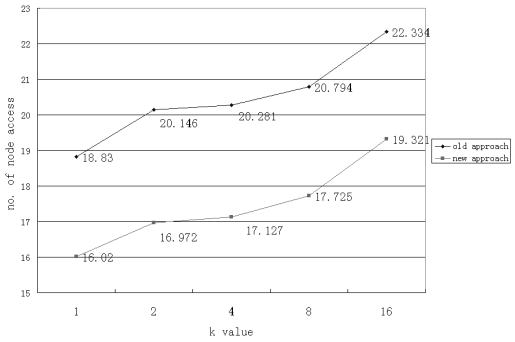


Figure 14. Node access (cell side = 0.01)

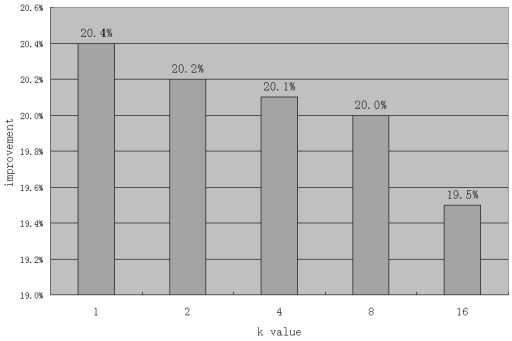


Figure 15. Improvement (cell side = 0.001)

nodes will not be visited.

A comparative experiment has been conducted to prove that the improved approach can significantly reduce the disk I/O. Assume the static caching is adopted and the non-leaf nodes have been cached in the memory. And the experiment result is represented in **Figure 13 $\sim$ 16.**

In the first experiment, we fixed the query rectangle as a square with its side equaling 1/1000 * service area side. And we record the disk I/O of two approaches with different k values. Each figure point is the average of disk I/O resulted in by 1000 randomly generated queries. And In the second experiment, we change the query square side from 1/1000 * service area side to 1/100 *service area side.

In the first experiment, using the improved approach the disk I/O is reduced by 19.5% $\sim$ 20.4% and in the second

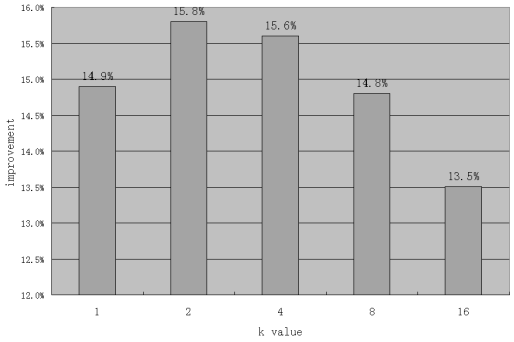


Figure 16. Improvement (cell side = 0.01)

experiment, the disk I/O is reduced by 13.5% $\sim$ 14.9%.

5 Conclusions

In this paper we study how to provide the location-based service users with privacy-preserving data access in a client based style instead of deploying middleware architecture. And we mainly focus on how to handle nearest neighbors queries in such a context. A named 'Robust Static Location Cloaking' algorithm is proposed along with related experiment results showing its effectiveness and how to configure the system parameters to lower the cloaking overhead most. Extra we have improved the kRNN algorithms to reduce the database I/O occurring in the server side.

As for future work, we will try to extend the query type to NS(nearest surrounder) query, reverse-kNN query and etc. Moreover it's interesting and attractive to realize the privacy preserving data access when moving objects datasets rather than stationary spatial datasets are manipulated.

References

[1] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In VLDB, 2004.

[2] L. Xiong, S. Chitti, L. Liu. Topk queries across multiple private databases. In IEEE ICDCS, June 2005.

[3] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In IEEE ICDE, 2005.

[4] N. Zhang and W.Zhao Distributed Privacy Preserving Information Sharing In VLDB, 2005

[5] C. C. Aggarwal On k-Anonymity and the Curse of Dimensionality In VLDB, 2005

[6] C.Yao, X.S. Wang, S. Jajodia Checking for k-Anonymity Violation by Views In VLDB, 2005

[7] X.Xiao and Y.Tao Personalized Privacy Preservation In SIGMOD, 2006

[8] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. In IEEE Pervasive Computing, 2(1):56-64, 2003.

[9] M.Youssef, V.Atluri, N.R. Adam Preserving Mobile Customer Privacy: An access Control System for Moving Objects and Customer Profiles In MDM, 2005

[10] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking In ACM MobiSys. 2003.

[11] B. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy in IEEE ICDCS, June 2005.

[12] A. R. Beresford and F. Stajano. Location privacy in pervasive computing IEEE Pervasive Computing, 2(1):46-55, 2003.

[13] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In SIGMOD Conference, Boston, Massachusetts, pages 47-57, 1984

[14] G. R. Hjaltason and H. Samet. Distance browsing in spatial databases TODS, 24(2):265-318, 1999.

[15] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects in SIGMOD, 2000.

[16] Y. Tao, D. Papadias, and J. Sun. The TPR*-tree: An optimized spatio-temporal access method for predictive queries in VLDB, 2003.

[17] Y. Tao, D. Papadias, and Q. Shen. Continuous nearest neighbor search. In VLDB Conference, Hong Kong, China, pages 287-298, 2002.

[18] M. F. Mokbel, X. Xiong, and W. G. Aref. SINA: Scalable incremental processing of continuous queries in spatio-temporal databases in SIGMOD, 2004.

[19] R. Cheng, D. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments in TKDE, 16(9), 2004.

[20] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Querying private data in moving-object environments CERIAS Tech Report 2005-45, Purdue University, 2005.

[21] B. Gedik and L. Liu. MobiEyes: Distributed processing of continuously moving queries on moving objects in a mobile system in EDBT, 2004.

[22] M. Kolahdouzan and C. Shahabi Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases in VLDB, 2004

[23] C. K. Lee, W.C. Lee, and H. V. Leung. Nearest surrounder search in ICDE, 2006.

[24] Haibo Hu and Dik Lun Lee. Range Nearest Neighbor Search. In TKDE, 2005.

[25] J. Xu, B. Zheng, W.C. Lee, and D. L. Lee. Energy efficient index for querying location-dependent data in mobile broadcast environments in Proc. IEEE ICDE, pp.239-250, March 2003.

[26] B. Zheng, J. Xu, W.C. Lee, and D. L. Lee. Grid-partition index: A hybrid method for nearest-neighbor queries in wireless location-based services accepted to appear in VLDB Journal (VLDBJ), 2005.

[27] J. Xu, B. Zheng, W.C. Lee, and D. L. Lee. The D-tree: An index structure for planar point queries in location-based wireless services IEEE Trans. Knowledge and Data Engineering (TKDE), 16(12): 1526-1542, Dec. 2004.

[28] D. L. Lee, W.C. Lee, J. Xu, and B. Zheng. Data management in location-dependent information services. IEEE Pervasive Computing, 1(3): 65-72, July-September 2002.

[29] W.C. Lee, J. Xu, and B. Zheng. Data management in location-dependent information services. Tutorial presented at IEEE ICDE '04, Boston, MA, March 2004.

[30] H. Hu, J. Xu, and D. L. Lee. A generic framework for monitoring continuous spatial queries over moving objects in Proc. ACM SIGMOD, Baltimore, MD, June 2005.

An Adaptive and Intelligent Controller for Cluster-based Web Server

Chan Ka Ho, Chu Xiaowen
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
khchan@comp.hkbu.edu.hk, chxw@comp.hkbu.edu.hk

Abstract

Nowadays, it is often desirable to isolate performance of different services and classes of requests from each other. A lot of works have been done on the controller of standalone server to allocate resources for different classes of requests. However, web contents and traffic keep increasing. Therefore, people would like to implement cluster-based web servers. In the cluster environment, there are problems in senses of achieving performance isolation, distribution of requests and resources. This paper introduces basic web server, cluster based web servers design and implementation.

Keyword: QoS, Web Servers, Delays, Dispatcher

1 Introduction

Because of wide spread usage of web services, web servers experience an extreme variation in demand. This effect may be caused by the different kind of access file. During overloaded, not all requests can be served at a timely manner. Therefore providing good services to premium users is an essential task for most of the web servers.

Researchers have proposed the *proportional delay differentiation* model for web servers. In [10], a classical PI controller is proposed to guarantee the delay ratios between different classes. As absence of accurate model for the nonlinear web server, the web server is modeled by a second order system. Moreover, the parameters are found in the system identification phase. However, classical PI controller cannot get satisfactory results on some of the performance metrics, such as settling time and oscillation. Fuzzy controller is proposed to solve the problem of the classical PI controller for absolute delay guarantee in [25]. The main drawback of this controller is large amount of parameters to be tuned. It is especially difficult to make initial approximate adjustment as suggested in [18] and also depends on the quality of the expert knowledge.

Since well-equipped web server may be overloaded, it is important to implement web server as a clustered web servers. Numbers of controller are introduced in previous paragraphs, but controller for delay ratio differentiation on cluster-based web servers has not yet launched.

Delay times of requesting web pages are analyzed in [16, 17]. However, the technology advanced and contents differences make the analysis outdated. Therefore, analysis of delays of up-to-date web contents is important for enhancing performance of web servers. It can help to provide information of human behaviour for the controller.

The rest of the paper is organized as follows: Section 2 introduces some background information. Section 3 presents the basic idea of HTTP web server. The architecture of the cluster-based web servers is described in Section 4. Finally, Section5 concludes the whole paper.

2 Background

This section presents some technical background on cluster-based servers and controller.

2.1 Cluster-based Servers

A cluster consists of a number of computers connected by a network, usually a high-speed LAN [8]. The computers in the cluster can be categorized as front-end dispatching machine and the back-end web server nodes. The main motivations of building cluster-based web server are reducing the client perceived latencies and minimizing the network load.

- One single node in the cluster acts as front-end, also known as *dispatcher*. It is the point of contact with clients. When request arrives, the dispatcher decides which server in the cluster is responsible to serve the request. In decision making phase, the dispatcher takes account the current load and loading capabilities of the

servers. Besides, locality [20] is also essential for the decision making.

- At the very beginning, the back-end node is capable of serving any requests as normal web servers. However, the back-end servers are responsible to a subset of contents for the web sites because of the request distribution strategy.

2.2 Quality of Service and Controller

Assume the connection delay denotes the time interval between the arrival and accept of a connection request from clients. Let the processing time be the time between the start of web server processes processing a request and return of the response to clients. For *Quality of Service, QoS*, connection delay is focused, and we simply define "delay" as the connection delay in the remaining part of this paper.

Research on providing delay differentiated services to web systems is popular in recent years. In [9], an admission control method based on PI controller is proposed. However, the controller design is based on a system model which assumes the web server can be modeled by M/G/1/PS model. According to [19], it is not accurate enough to control the complex and non-linear web server system.

Besides, feedback control theory has been applied to web systems for differentiated services in [10]. Furthermore, queuing theory has been introduced in [24] to provide differentiated delay services.

A feedback control mechanism has been used to adjust parameters KeepAlive and MaxClient of Apache in [23]. As a result, the Apache shows quick convergence and stability. However, the two parameters do not directly address metrics of interest to the web site.

3 A Measurement Study of Delay Performance

Apache [1] is one of the most widely used web servers. It is an open-source HTTP server. Moreover, it is a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. Besides, there are wide varieties of add-ins modules for the HTTP web server. Therefore, we have chosen to implement Apache on our testbed.

3.1 Working Principle

Apache is started up at the very beginning. During the startup phase, large amount of processes, known as server processes, are started in order to handle requests. The number of server processes may change throughout the working time of the Apache. The server processes wait for requests come in. When requests reach the server machine via TCP, the requests are queued up in a connection queue. If there is any free server process, requests in the queue will be served in the FIFO manner.

The server process analysis the request. Then, it allocates the objects requested, which may be in the cache memories or in any storage devices. Finally, the objects will be transferred to the client.

3.2 Delays

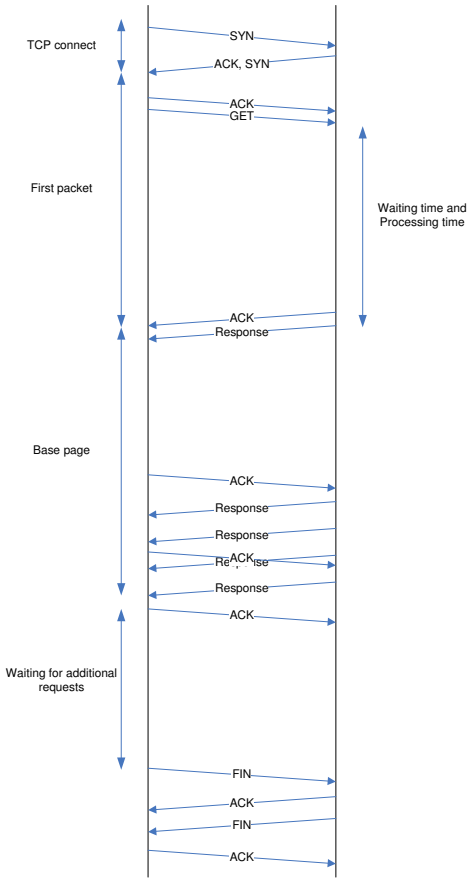


Fig 1. HTTP packet flows, it is not drawn in scale

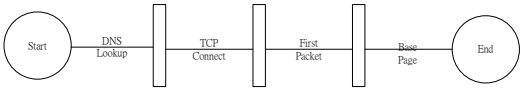


Fig 2. Delays of web server request

The HTTP request and response sequence is shown in fig. 1. The sequence includes the 3-way handshaking, request, acknowledgements and responses. According to [11], web page download time can be separated into six distinct components. However, for simplicity and usage of our research, we have defined the download time to four components as shown in fig. 2. Total download time is a useful indicator of site performance. Therefore, to improve the

performance, we must measure the components separately, to isolate bottlenecks and design appropriate changes. The following section will describe the components.

- DNS Lookup Time
  The Domain Name System (DNS) allows a host to have an alphanumeric name to replace its own IP address. The DNS lookup time is the time for the system to translate a name into IP address. This process involves querying a series of DNS servers throughout the DNS hierarchy [21].

- TCP Connection Time
  Once the IP address of target web server is known, 3-way handshaking begins to establish the TCP connection. The setup of TCP connection is essential before the request sending to the web server. In Apache, there are two parameters, including size of listen queue and number of processes, governed this delay [22].Within these two parameters, it is clear that there is a greater impact for changing the parameter of number of processes.

- Time between send and receive of first packet
  The first packet usually includes several HTTP response, such as HTTP 200 response. If the server response is HTTP 200, we can equate this with the server delay. The delay includes queuing time, locating of static pages or construct of the dynamic pages. However, if the response received is HTTP 302 and so on, redirection of pages involved. We need to handle the pages in another way. In short, it is the main delay concerned for the controller system.

- Time between receive of first and last packet
  After the first packet has been received and acknowledged. TCP packets of the whole html page are sent continuously. This delay time is typically longer because large volume of data is necessary to be transferred.

The delays analysis is essential to enhance performance of web servers. To build a robust and efficient controller, it is important to train up the controller.

3.3 Delay Analysis

In order to analysis the series of delays for web request, we would like to have a list of most well known and busy web sites. The URL of these web sites are collected from [6, 7]. After the collection of the delays from these different sites, we can have some statistical operation from the information. For example, we can calculate mean connection delay, minimum connection delay and maximum connection delay. These are useful information for the design of a robust controller for web servers and dispatcher.

With the list of web sites, a C programme is developed. The programme sends requests to the web servers on the list one-by-one and then stored up the delays on a text files. The programme with the same list is executed on different machines in different countries. As we would like to analysis the delays from different countries in the approximately the same time to the same server, the standalone running experiment is not suitable.

In order to meet the target, we would like to change the programme to work on a distributed manner. There are a controller machine and many experimental machines throughout the world. When the controller sends the URL of target web server to the experimental machines, the HTTP web requests are sent at approximately the same time to the target web server. Then, the delays are collected by individual machine and sent back to the controller for the log writing. In order to make sure all the HTTP requests are sent at approximately the same time, the next URL will be sent from the controller machines when all the delays from experimental machines are collected.

Since there are too many machines which are running on top of different operating systems, such as Linux, Windows and MacOS [5], the experiment are still in the phase of programming and testing. There are no experimental results and analysis discussion in this paper.

3.4 Human Behaviour Analysis

Number of types of web contents keeps increasing. Ten years ago, most of the web pages are static and lack of users interaction. However, for current web contents involve more and more interaction. The change of contents is led by the development of security measures, web 2.0 and the RSS [12].

In [16, 17], researchers have studied the web delays and web server performance several years ago. As the content type changes, the browsing habits of human are changed as well, for example people would like to spend minutes to read short static content before, but people are willing to spend hours to use online banking or read blog, newsgroup and so on. Therefore, the analyzes of delay and web server performance are outdated. Therefore, we would like to do the analysis again to get up-to-date information.

We have requested people in different target groups to be our samples. Then, we use Ethereal [2] to capture packets flow information. With the aid of the Ethereal, we can mark down useful information of every time accurately. For example, we can get how often web servers terminate the connection, and how the client side would like to re-connect, how much information would a specific client would like to retrieve and so on. Because the study consists of a large sample size, we can get a great picture of the browsing habits of different people. With these information, we can

build a much more robust and efficient controllers for the web servers containing up-to-date web contents.

4 Cluster-based Web Server

The cluster-based web servers are different from standalone web server. As the number of server nodes is more, the resource and usage measurement become more difficult. Therefore, the controllers proposed for the standalone server is not applicable in the cluster-based web servers. It is essential for the cluster-based web server to provide locality and load balancing [20]. The advantages like enhancing performance, increasing secondary storage scalability and specialized back-end nodes can be achieved. In order to minimize the disturbance of the clients, we need to provide a client-blinded service [15]. The following subsections will discuss the cluster-based web server thoroughly. First of all, the architecture of the cluster-based web servers will be introduced. Then, there will be a discussion of the dispatcher design.

4.1 Cluster-based Servers Architecture

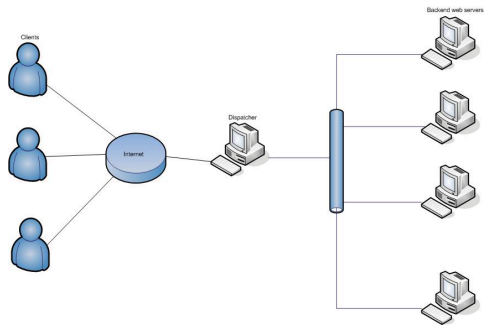


Fig 3. Cluster-based Server Architecture

There are front-end dispatcher and back-end web servers in the cluster. The machines are connected to a high speed local area network. The dispatcher is acted as the point of contact with clients. Then, the dispatcher chooses a suitable back-end server for the request according to back-end loading information, documents distribution and the request information. The cluster setup is as shown in fig. 3.

4.2 Dispatcher Design

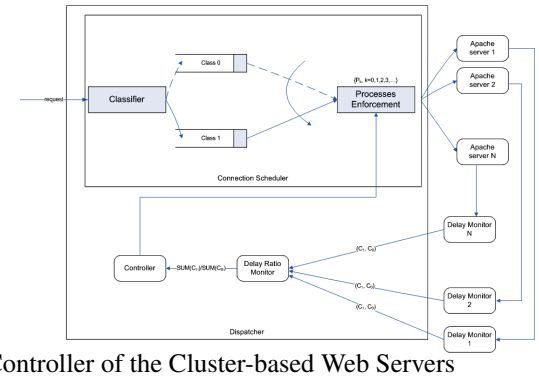


Fig 4. Controller of the Cluster-based Web Servers

In [20, 14, 13], Aron raised the idea of Locality-Aware Request Distribution (LARD) for the dispatching of requests to the cluster web servers. The idea of this dispatching algorithm considers the request of client and the status among back-end servers. However, the LARD considers all requests at the same priority. Since more and more web sites would like to provide better service for premium class users, the LARD dispatching algorithm is not suitable. Therefore, adaptive controller for the cluster-based web servers is interesting for us to design and implement.

Monitoring and controlling modules are essential for controllers. However, in the environment of cluster-based web servers, it is hard to design and implement one single monitor. In our design, we would like to implement individual monitor on each back-end server node. The front-end dispatcher delivers the requests to back-end nodes according to the request object. Then, the target back-end server node handle the request according to its own controlling and monitoring algorithm. Moreover, there is another controller on the dispatcher. The dispatcher implements its own TCP queue that is used to queue up all the requests. The controller on the dispatcher uses information from back-end servers to determine how to control the request processing rate of different priority group users. The control of request processing rate is to control the rate of request redirection from the queue to back-end servers.

4.3 Controller in Cluster-based Web Server

There is one front-end dispatcher, three back-end web servers and three client machines. All the machines are interconnected with a gigabit switch. The back-end servers are dual AMD CPU machines. On the other hand, the dispatcher is a P4 machine, and the clients are P3 machines. All the machines are operated on top of Fedora Core 4 [3]. Apache 1.3.9 [1] is setup on the back-end web servers. A performance measurement tool, httperf [4], is setup on top of every client machines.

The dispatcher is designed according to a basic assumption, which is the request objects are co-related. For example, if user requests the information of staff, he/she will also read the Email address of the staff. Therefore, the request objects are cached on the same back-end web server. The dispatcher analyzes the first request object from the user. Then, it decides which back-end server is responsible for handle the requests. And then, all similar requests will be forwarded to the target web server. At the same time, the dispatcher is also responsible for forwarding the response from the target web server to the clients. As discussed in previous section, there will be controllers on top of front-end dispatcher and the back-end web server nodes.

In order to build robust controlled cluster-based web servers which is more suitable for up-to-date web services, we would like to integrate our delays investigation result to our controller. Therefore, there is not yet any experimental result for the controller of cluster-based web server.

5 Conclusion and Future Works

With the wide spread usage of the web services and the number of access to popular web sites is ever increasing, single web server for the serving purpose become less feasible. The use of cluster-based web servers is more suitable for the growth trend. According to the investigation of the network usage, we can see that content aware and load balancing are extremely essential for the cluster-based web servers. Furthermore, controller works well in single web server, it is foreseeable that the performance of cluster-based web servers will be enhanced after applying the controller. Besides, we have series of testing to analysis up-to-date web contents, such as RSS 2.0 and web 2.0, performance and human behaviour to these new content types. We can have a noticeable performance improvement for the controlled cluster-based web servers system.

There are lots of improvement areas for the dispatcher. We have proposed to implement a classical PI controller on top of the front-end dispatcher and individual controller on each back-end server machines. In order to provide the best services, it is a must to analysis the performance of different combinations of controllers. Besides, we can see that the controlling mechanism operates on top of cluster-based systems, so the oscillating effect between steady points is harder to control. In order to solve this problem, decoupling and optimization are essential for every controller.

Occasionally, hardware may be upgraded or replaced due to malfunction. The configuration of back-end web servers may be different, so the performance may be varied. Therefore, Weighted Round Robin technique is not suitable for this environment. In order to get rid of this problem, distribution technique includes delay information of back-end servers are crucial in decision making phase.

References

[1] Apache Project. http://www.apache.org/.

[2] Ethereal. http://www.ethereal.com/.

[3] Fedora Core 4 – Red Hat Linux. http://www.redhat.com/en_us/USA/fedora/.

[4] httperf. http://www.hpl.hp.com/research/linux/httperf/.

[5] MacOS. http://www.apple.com/macosx/.

[6] PC Magazine. http://www.pcmag.com/category2/0,1874,7488,00.asp.

[7] web100. http://www.web100.com/listings/all.html.

[8] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, and P. Gauthier. Cluster-based scalable network services. *Proceedings of the Sixteennth ACM Symposium on Operating System Principles*, 1997.

[9] A. Kamra, V. Misra, and E. Nahum. Yaksha: A Controller for Managing the Performance of 3-Tiered Websites. *Proceedings of the 12th IEEE International Workshop on Quality of Service*, 2004.

[10] C. Lu, Y. Lu, T. F. Abdelzaher, J. A. Stankovic, and S. H. Son. Feedback Control Architecture and Design Methodology for Service Delay Guarantees in Web Servers. *IEEE Transactions on Parallel and Distributed Systems*, 2005.

[11] Chris Loosley. E-commerce Response Time: A Reference Model. *Proceedings of CMG2000 International Conference*, 2000.

[12] Heinz Wittenbrink. *RSS and Atom: Understanding And Implementing Content Feeds And Syndication*. Packt Publishing, 2005.

[13] M. Aron, D. Sanders, P. Druschel, and W. Zwaenepoel. Scalable Content-aware Request Distribution in Cluster-based Network Servers. *Proceeding of the 2000 Annual Usenix Technical Conference*, 2000.

[14] M. Aron, P. Druschel, and W. Zwaenepoel. Cluster Reserves: A Mechanism for Resource Management in Cluster-based Network Servers. *Proceeding of the ACM Sigmetrics 2000 International Conference on Measurement and Modeling of Computer Systems*, 2000.

[15] M. C. V. Cardellini, E. Casalicchio, and P. S. Yu. The state of the art in locally distributed web-server systems. *ACM Computing Surveys*, 2001.

[16] M. Habib and M. Abrams. Analysis of Sources of Latency in Downloading Web Pages. *World Conference on the WWW and Internet*, 2000.

[17] P. Mills and C. Loosley. A Performance Analysis of 40 e-Business Web Sites. *CMG Journal of Computer Resource Management*, 2001.

[18] P. Pivonka. Comparative Analysis of Fuzzy PI/PD/PID Controller Based on Classical PID Controller Approach. *Proceedings of IEEE World Congress on Computational Intelligence*, 2002.

[19] V. Paxson and S. Floyd. Wide Area Traffic: The Failure of Passion Modelling. *IEEE/ACM Transactions on Networking*, 1995.

[20] V. S. Pai, M. Aron, G. Banga, M.Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum. Locality-Aware Request Distribution in Cluster-based Network Servers. *Proceedings of the 8th Conference on Architectural Support for Programming Languages and Operating Systems*, 1998.

[21] W. Richard Stevens. *TCP/IP Illustrated, Volume1: The protocols*. Addison-Wesley, 1994.

[22] W. Richard Stevens, Bill Fenner, and Andrew M. Rudoff. *UNIX Network Programming Volume 1, Third Edition: The Sockets Networking API*. Addison Wesley, 2003.

[23] X. Liu, L. Sha, Y. Diao, J. L. Hellerstein, and S. Parekh. Online Response Time Optimization of an Apache Web Server. *Proceedings of the 11th International Workshop on Quality of Service, pages 461-478*, 2003.

[24] Y. Lu, T. F. Abdelzaher, C. Lu, L. Sha, and X. Liu. Feedback Control with Queuing-Theoretic Prediction for Relative Delay Guarantees in Web Server. *Proceedings of IEEE Real Time and Embedded Technology and Applications Symposium*, 2003.

[25] Y.Wei, C. Lin, X.-W. Chu, T. Voigt, and F. Ren. Fuzzy Control for Guaranteeing Absolute Delays in Web Servers. *to appear in International Journal of High Performance Computing and Networking*, 2006.

Location Estimation in Library Environment based on Enhanced Fingerprint approach

Wilson M. Yeung

Abstract

As ubiquitous computing gained much attention in recent years, location estimation in wireless LAN becomes a hot topic. Previous research work suggests the use of the averaged Received Signal Strength (RSS) as fingerprint can achieve high accuracy for location estimation. In a library environment, however, the accuracy of such traditional approach is barely acceptable. It is because library contains considerably large number of metal bookshelves, and limited number of access points. Worse yet, the layout of these access points in the library is fixed for connection to the Internet, and therefore it is hard to change the environment to adapt for location estimation system. In this paper, we introduce an enhanced fingerprint (EFP) algorithm, and tested it in a library environment. The experiment result showed that the proposed EFP algorithm can have more than 30% of improvement in accuracy over traditional approaches without changing anything in the library environment.

1 Introduction

Wireless communication technology has gratefully advanced and deployed in every city around the world in recent years. People can use mobile devices to communicate, and retrieve information from Internet. Nowadays, many buildings have installed wireless LAN (802.11b/g) to provide pervasive network access, and therefore residents can connect to the network anywhere using their laptop computers or mobile devices such as PDA and mobile phones. Besides providing network access, location-aware services can also be provided through the wireless LAN. These services enable the users to retrieve the most appropriate information based on the users' location.

In order to provide location-aware service, user's location determination is required. Location estimation in wireless LAN can be accomplished by retrieving signal strength from the nearby Access Points and applying a positioning algorithm to compute the user's location. Many wireless LAN positioning algorithms and systems, such as the RADAR system [1] [2], the HORUS (JC & IT) system [10] [11] [12] and the Nibble System [3], have been proposed in the literature. These location estimation systems acquire good performance mostly in office environment. However, given an specific indoor environment (a library, for instance), with many obstacles (e.g. bookshelves), the accuracy of these system will be degraded. It is because the obstacles may block or reflect [9] the signal to or from the access points, and therefore the signal strength is not as stable as in the office environment. In the library environment, metal bookshelves act like walls [9], and therefore distorted the wireless LAN signal [8] to a certain extent that makes locating a user in such an environment to be very challenging.

In this paper, we present an enhanced positioning algorithm which operates on the basis of the fingerprint approach. Experimental results had shown to us that the proposed algorithm can attain more accurate results for location estimation in a library environment, as compared to other previously proposed algorithms.

The rest of the paper is organized as follows. In Section 2, we introduce the related works for wireless LAN positioning. Section 3 presents our proposed algorithm. In Section 4, we describe our test site and the experiment setup. Experiment results and algorithms performance evaluation are presented in Section 5. And finally, in Section 6 a summary of our study and research findings.

2 Related Works

Previous research work in wireless LAN positioning has been focused in two broad approaches: (1) Fingerprint; and (2) Probability.

The basic principle of fingerprint-based [8] [5] [4] algorithm is shown in Figure 1. There are two phases, the off-line phase and the online phase, in fingerprint-based algorithm. In the off-line phase, the Received Signal Strength (RSS) from the nearby access points at known locations is saved into a database. Hence, each location in the database has a RSS tuple, and these RSS tuples are used as a loca-

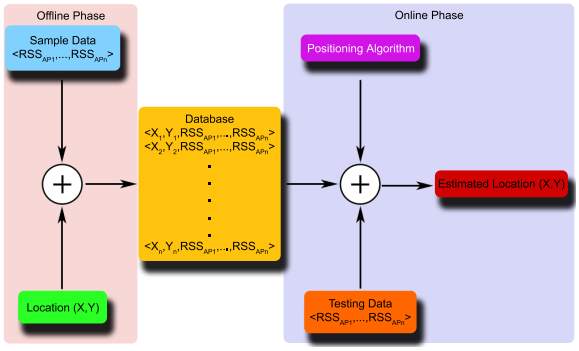


Figure 1. Fingerprint Approach

tion fingerprint [8] [5] [4]. As shown in Figure 1, in the next phase, i.e. the online phase, the received (testing) RSS tuple is matched with all fingerprints stored in the database, and the similarity, or the distance in signal space, is calculated on the basis of a positioning algorithm. The location of the most similar fingerprint is then reported as the estimated location or the current location of the user. For example, the RADAR system [1] [2] and the fingerprint algorithm by Wong et al. [9] are based on this approach to search for the nearest neighbor(s) in the signal space (NNSS) [1] [2]. The nearest neighbor, or the interpolation of K neighbors, will be reported as the users' current location. Both algorithms used Euclidean distance as the similarity measurement, and it is formulated as:

$$D = \sqrt{\sum_{i=0}^{N} (RSS_{AP_i} - R\hat{SS}_{AP_i})^2} \qquad (1)$$

where N is the total number of Access Point (AP), RSS_{AP_i} is a database entry and $R\hat{SS}_{AP_i}$ is the received RSS tuple. Sometimes the access points involved in the RSS tuple of database entry or the received one are different, i.e. RSS from some access points can be missing. A previous research work proposed [9] to find a way to assign the minimum value of signal strength as the RSS of the missing access points trying to compensate the missing value problem.

On the other hand, rather than using the averaged RSS tuple, probability-based algorithms collect and store the RSS distribution of nearby access points of known locations during the off-line phase. In the online phase, it uses the received RSS tuple to investigate the probability of all known locations. The location with the highest probability is reported as the result. The HOURS system [10] [11] [12], for instance, is a location determination system based on this approach. Besides that, the HOURS system proposed a location clustering technique [11] [12] which can reduce the cost for location estimation. The idea of location clustering is based on the access points may not cover all locations, i.e. some locations are covered by all access points while others may be covered by a few of access points. Based on this characteristic, we may cluster the locations which are covered by same access points as a group, and estimate the user's location based on one of these location clusters, and hence to reduce the cost for location estimation.

3 The Enhanced Fingerprint Algorithm (EFP)

We introduce an enhanced fingerprint algorithm which could achieve better performance than the original fingerprint algorithm. Pervious research works [1] [2] [9] has been focused on using an averaged RSS tuple as a location fingerprint. Rather than using one averaged RSS tuples, as a location fingerprint, to represent a location, we use more than one RSS tuples to represent a location. In other words, one location can have more than one location fingerprint. We cannot, however, use all the training RSS tuples as fingerprints; otherwise it greatly increases the computation time. Therefore we need to choose the most representable RSS tuples as fingerprints of a location. To sum up, we need to solve two significant problems: (1) How to choose the most representable RSS tuples? (2) How many RSS tuples we should use?

We solve the problems as follows:

For each location, we calculate the RSS distribution of each access point. Afterwards, we can get the RSS value with the highest occurrence, i.e. the highest probability, for each access point, and form a RSS tuple. We name this RSS tuple as a reference point in the signal space. The reference point can be formulated as: Ref_l $=\{RSS'_{AP_1}, RSS'_{AP_2} ... RSS'_{AP_n}\}$, where l is a known location, and RSS'_{AP_i} is the RSS value with the largest probability in RSS distribution of Access Point (AP) i. The representativeness of a RSS tuple should be according to the Euclidean distance in the signal space between the RSS tuple and the reference point, i.e. the tuple having a shorter distance to the reference point should get a higher probability, and the tuple further away from the reference point should get a lower probability.

Before calculating the Euclidean distance, we should remove all repeated RSS tuples, in other words, only the unique RSS tuple will be used in calculating the distance. We then calculate the Euclidean distance between the unique training RSS tuples and the reference point, and sort the distance in ascending order. In spite of using all tuples, we only select a portion of it.Say, the Top L% of the RSS tuples will be saved as the fingerprint of a specific location. As a consequence, each location can have one or more fingerprints.

In the online phase, the received RSS tuple will match with candidate locations' fingerprints. For each location, we calculate the Euclidean distance between the received RSS tuple and each fingerprint, and the smallest value of the distance among all combination will be promoted as the value of its location. Finally, the location with the smallest value will be reported as the estimation result.

As the number of fingerprints increases, the number of comparison between fingerprints and the received RSS tuple also increases. In other words, the algorithm may need more computation time for location estimation. In order to reduce the computation time, we also applied a clustering technique. Rather than location clustering, we group RSS tuples based on the access points combination. For example, there are two RSS tuples, T_1 , T_2, where T_1 is obtained at location A and T_2 is obtained at location B. T_1 and T_2 will be putted in the same group if the access points involved in T_1 and T_2 are exactly the same. In other words, the RSS tuples in the same group are constructed by same set of access points, and these RSS tuples can be taken at different locations. Theoretically, there can be at most 2^n- 1 groups, where n is the number of access point. In practice, however, one location is generally covered by at least r access points. Therefore the maximum number of cluster should be :

$$N = 2^n - \sum_{i=1}^{r} C_{i-1}^{n} \quad (2)$$

4 Experiment Setup

In this section, we describe the experiment setup. Our testbed is the fifth floor of a seven-storey library building. The area of testbed is about $636.52m^2$. On the test site, six access points were detected. Two of them are installed on the fifth floor while another two are installed on the split-level floor between the forth and the fifth floor. The remaining two access points are installed on the forth floor and the third floor (see Figure 2).

The thickness of most of the metal bookshelves is 1.34m, except the large moveable bookshelf (located on the top-right corner of the floor plan). We want to emphasize here that the wireless LAN (i.e. the access points) are designed and installed for providing wireless network connection only. In the experiment, we did not reallocate or install any access point. Besides, we did not change the layout, including the position of bookshelves.

We used an HP iPAQ rx3417 Pocket PC, which has a 802.11b wireless LAN card built-in, as our signal measurement device. We used the software called WiFi Graph [7] to record the signal strength from the access points. In the testbed, we defined 147 sampling location (the squares in Figure 2). We obtained samples of the RSS tuples in each of the 4 directions, say North, East, South, and West. For

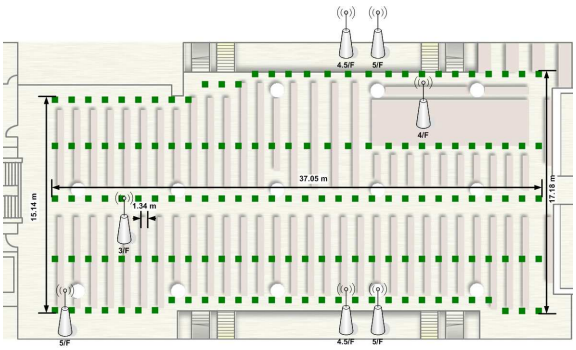


Figure 2. Floor plan

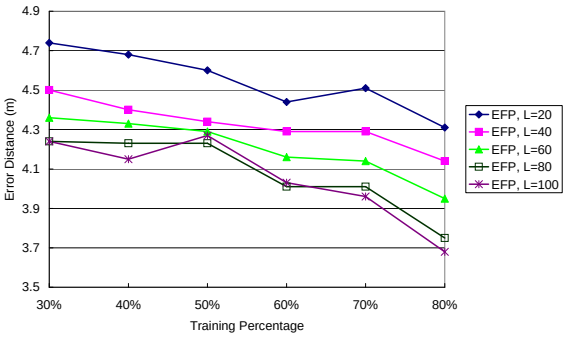


Figure 3. Average error distance of Enhanced Fingerprint algorithm

each direction, 50 samples of the signal strength received from the neighboring access points are obtained, and therefore we got 29,400 samples of data in total. To evaluate the performance of the proposed algorithm, we tried to train the algorithm with different percentage of our samples (30%, 40%, 50%, 60%, 70%, 80%), and use the remaining samples for testing.

5 Results and Discussion

In this section, we evaluate the performance of the proposed algorithm. We also compared our result with another four algorithms, they are the Fingerprint algorithm by Wong et al. [9] (FP), RADAR [2] [1], Joint Clustering (JC) [12] [11], and Weighted Center of Gravity (WCG) [9] [6]. Both the FP and RADAR use averaged RSS tuple as location fingerprints.

Figure 3 shows the average error distance of the proposed algorithm. As mentioned in the above, the top L% of the RSS tuples, according to their Euclidean distance from the reference point (in signal space), are used as the

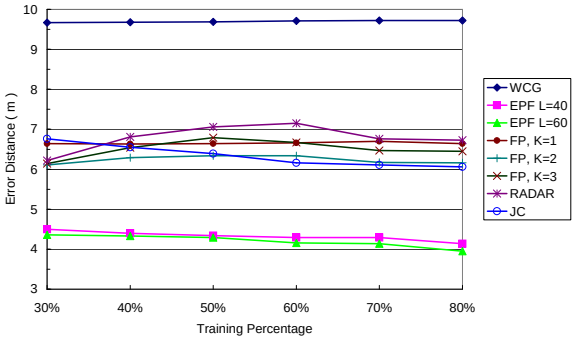


Figure 4. Average error distance

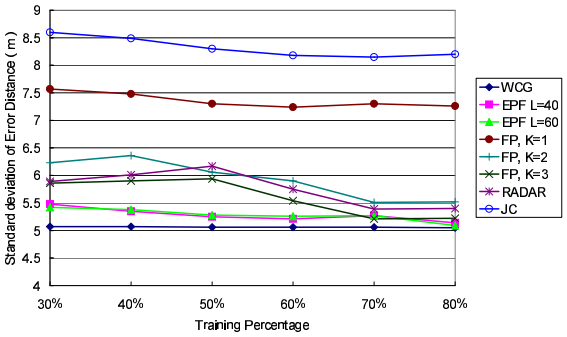


Figure 5. Standard deviation of error distance

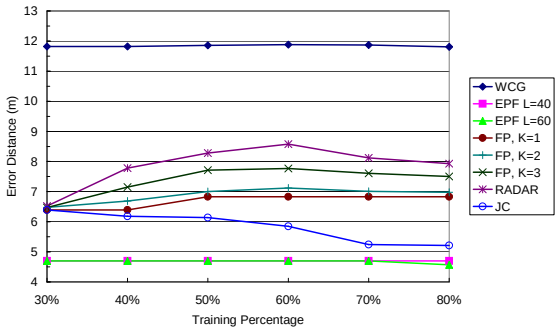


Figure 6. 67^{th} Percentile of Error Distance

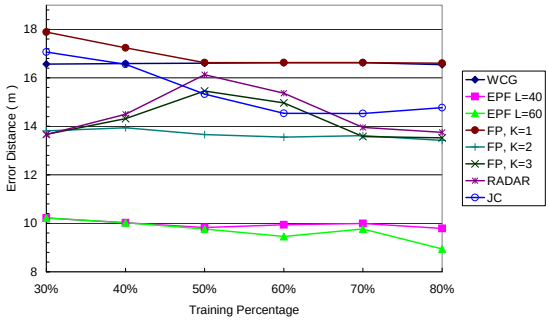


Figure 7. 90^{th} Percentile of Error Distance

fingerprint of the location. It's clear that the accuracy increases with the training percentage. When L was increased from 20% to 80%, the performance was in general improving. However, when L=100, i.e. all unique samples were used as fingerprint, the accuracy did not improve significantly, as compare to when L=80. It seems that some of the samples are just noise, and when all samples were used as the fingerprint, these noisy samples would lower the performance. Despite of the accuracy, computation overhead also increases with L. It is obvious that the more fingerprint, the more the computation time is needed. For the rest of the comparisons, we used L=40 & L=60 for our references, as we think L = 40 & 60 should get a reasonable balance between computation time and accuracy.

Figure 4 and Figure 5 show the average error distance and the standard deviation of the error distance for each of the algorithms against different percentage of training. For the rest of the paper, we focus on the result using 80% of the data for training, unless otherwise specified. We observe that our proposed algorithm got a smaller average error distance than the other algorithms. The average error distance of the enhanced fingerprint (EFP) algorithm with L=40 and L=60 are 4.15m and 3.95m, respectively. On the other hand, the average error distance of WCG is 9.72m. The value K in the Fingerprint (FP) means the result location is the interpolation [9] of the most probable K candidate locations. Among these three different values of K, FP with K=2 got the best performance where the average error distance is 6.16m. The RADAR algorithm, another fingerprint-based algorithm, yielded 6.73m as the average error distance. The accuracy of the Joint Clustering (JC) algorithm is improving as the training percentage increases, and the average error distance of JC is at 6.06m.

Figure 6 and Figure 7 show the 67^{th} and 90^{th} percentile of the error distance. As shown in Figure 7, the 90^{th} percentile of the error distance of EFP L=40 & L=60 are 9.79m and 8.94m, while RADAR and JC yield 13.75m and 14.77m under the same value. The 90^{th} percentile of the error distance of WCG is slightly smaller than the corresponding value of FP with K=1. And their values are 16.55m and 16.61m respectively.

The accuracy for each of the algorithms can be summarized by the cumulative distribution function (CDF) of the

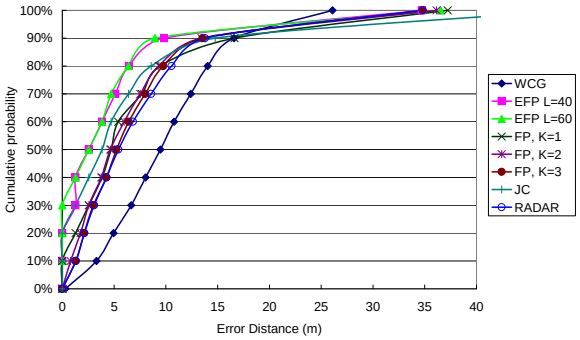


Figure 8. Cumulative Distribution Function of Error Distance

Algorithm	Seconds
WCG	2.01
FP K=1	26.73
FP K=2	27.00
FP K=3	25.26
RADAR	25.70
JC	2.56
EFP L = 0.4	98.71
EPF L = 0.6	141.75

Table 1. Time Consumed to estimate 14,800 samples

error distance, which is shown in Figure 8. The CDFs for EPF L=40 & L=60 are similar, except at 30%. The CDFs of JC and EPF L=40 are similar from 0% to 30%, and beyond 30%, there is a difference of 1m for the rest of the experiment. The end of CDF for JC is not shown in Figure 8, it is because the maximum error distance of JC is at 49.08m, and it is out of our plottings' range. The performance of FP (K=1, 2, 3) is lower than that of JC, we can obverse it through the CDFs in Figure 8. Although the maximum error distance of WCG is the smallest one among all algorithms, the overall performance is not as good as the other algorithms. It is because the result of WCG is based on access points' location and the signal propagation model. The current signal propagation model of WCG, however, cannot adapt well in an environment like the library and therefore the accuracy of WCG is lower.

As for the computation time, the computation time for each of the algorithms to estimate 50% (14,800) of the samples is shown in Table 1. This is the time required to produce 14,800 location estimation within our library setting. WCG and JC use less than 3 seconds to finish the computation. FP and RADAR consumed similar amount of time, it is because both of them need to match the 147 location fingerprints, for each round of estimation. Although the EFP achieved a higher accuracy, the computation time used is many times than that of JC, FP and RADAR. It is because computation time is proportion to the number of fingerprints involved. Obviously, there is a trade off between accuracy and computation time when do location estimation in such an environment. However, even if it take 141.75sec to calculate the recommendation for the 14,800 samples, that means our proposed EFP just take less than 10msec to return a location estimation. This is still fast enough for any kind of location-based application.

6 Conclusion

In this paper, we introduced an enhanced fingerprint algorithm (EFP), and we tested it in a library environment. The accuracy of the EFP improved for more than 30% over the tradition approach. The basic principle of this algorithm is to use one or more RSS tuple as location fingerprints, instead of the traditional approach of using one averaged RSS tuple. As the number of fingerprints increases, the computation time also increases correspondingly. Therefore, we applied a clustering technique based on the combination of the access points. The enhanced fingerprint algorithm can achieve an average accuracy of about 4m and we can estimate a user within 10m 90% of time in our library environment.

7 Future Work

Our future work includes further research on improving the accuracy and on the reduction of the computation time for location estimation. In order to deploy a real location-aware service in the library, the estimation precision of the algorithm must also be further improved. There are two strategies to enhance the accuracy. One strategy is modifying the method of comparison. Currently, we are using the Euclidean distance as measurement. Another strategy is enhancing the method of fingerprint selection which can ignore those noisy samples during the training phase, and therefore improve the accuracy. In addition, the latter strategy may also reduce computation time. Furthermore, we would like to investigate on clustering techniques to further reduce the computation time for location estimation.

In spite of fingerprint, we will also investigate the triangulation algorithm. We will explore the signal propagation model and evaluate the current method of creating empirical model. Currently, we use the sample data of known locations to get the averaged RSS value and calculate the distance from corresponding location to the access point. By collecting the RSS-distance tuples, we can simply form a empirical model. However, a pervious research [2] suggests that such kind empirical model cannot represent the

practical situation. It is because the locations which have a line of sight to the access point must obtain a stronger signal reading than other location. We have conducted a preparatory experiment and confirmed this situation in library environment. Therefore we will try to separate the empirical data and take different strategy on modeling the signal propagation.

References

[1] P. Bahl, A. Balachandran, and V. Padmanabhan. Enhancements to the RADAR user location and tracking system. Technical report, Microsoft Corporation, February 2000.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[3] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic room location service for wireless networked environments. *Lecture Notes in Computer Science*, 2201:18–??, 2001.

[4] K. M.-K. Chu, J. K.-Y. Ng, and K. R. Leung. A new approach for locating mobile stations under the statistical directional propagation model, to appear in proceedings of the ieee 20th international conference on advanced information networking and applications (aina 2006), april 18 - 20, 2006 vienna university of technology, vienna, austria.

[5] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *IEEE INFOCOM*, March 2004.

[6] K. K.-H. Kan, S. K.-C. Chan, and J. K.-Y. Ng. A dual-channel location estimation system for providing location services based on the gps and gsm networks. In *Proceedings of the 17th International Conference on Advanced Information Networking and Applications*, pages 7–12. IEEE Computer Society, March 2003.

[7] Kasuei Consultant Group. WiFi Graph, 2004.

[8] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis. On indoor position location with wireless lans. *The 13th IEEE International Symposium on Personal, Indoor, and MObile Radio Communications (PIMRC 2002)*.

[9] W. H. Wong, J. K. Ng, and W. M. Yeung. Wireless lan positioning with mobile devices in a library environment. In *Proceedings of ICDCS-MDC 2005 Workshop, Columbus, Ohio, USA.*, pages 633–636, June 2005.

[10] M. Youssef and A. Agrawala. On the optimality of wlan location determination systems. Technical Report UMIACS-TR 2003-29 and CS-TR 4459, University of Maryland, College Park, March 2003.

[11] M. Youssef, A. Agrawala, and U. Shankar. Wlan location determination via clustering and probability distributions, March 2003.

[12] M. A. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh. A probabilistic clustering-based indoor location determination system. Technical Report UMIACS-TR 2002-30 and CS-TR 4350, Department of Computer Science and UMIACS, University of Maryland, March 2002.