Effective Improved Graph Transduction

Jun Chen[1, 2] Yu Zhou[1] Yajun Gao[1] Bo Wang[1] Linbo Luo[3] Wenyu Liu[1]
1 Department of Electronics and Information Engineering
Huazhong University of Science and Technology
Wuhan ,430074 , P.R.China
2 China University of Geosciences
Wuhan ,430074 , P.R.China
3 Department of Electronics and Computer Engineering,
Hanyang University,
Seoul 133-791, Korea
chenjun71983@163.com, liuwy@mail.hust.edu.cn

***Abstract*—In this paper, we focus on the problem of shape retrieval and clustering. We put two questions together because they are based on the same method, called Improved Graph Transduction. For shape retrieval, we regard the shape as a node in a graph and the similarity of shapes is represented by the edge of the graph. Then we learn a new distance measure between the query shape and the testing shapes. The main contribution of our work is to merge the most likely node with the query node during the learning process. The appending process helps us to mine the latent information in the propagation. The experimental results on the MPEG-7 data set show that comparing with the existing methods, our method can complete shape retrieval with similar correct rate in less time；For clustering task，the existing literatures in this domain often use the distance measure between the testing data point individual which is proved not enough in the real applications. In this paper, we think about the core concept in semi-supervised learning method, and use a graph to reflect the original distance measure, and combine the density information of the data distribution with the distance measure. Given a set of testing data, we select the original data randomly and use graph transduction iterative on the defined graph. The given algorithm is rapid and steady comparing with the existing clustering method. The experiments show that the novel algorithm is effective for the clustering task.**

***Index Terms*—shape retrieval; unsupervised clustering; graph transduction; reducation of probabilistic transition matrix**

I. INTRODUCTION

Shape retrieval is an important problem in computer vision. There are many different kinds of retrieval methods based on shape matching [1] [2] [3] [4]. Other methods like [5] use the graph transduction to learn a better metric. Reference [5] is general and could be built on top of any existing shape matching method. Since [5] came from a semi-supervised learning method named label propagation, its convergence is as slow as label propagation. Reference [5] gives the convergence property of their method, which shows us that a long time is needed for converge process to get the final result. In this paper, we provide a novel method named Improved Graph Transduction. Our method reduces the iterative process, and speeds up the finish of the iterative process. In addition, our method could mine the implicit information after each iteration process, and take advantage of it to achieve better prediction in the next iteration. In [5], given a dataset of shapes, a query shape and a shape distance function need not to be a metric. A new distance function is expressed by shortest paths on the manifold formed by the known shapes and the query shape. We improve [5] by ranking the labeled result after each iteration round and choose the most likely shape in the testing shapes as believable shape. A believable shape means that it belongs to the same class with the query shape, and then we use the believable shape to help the iterative process. So a different distance function is learned rapidly. The detailed algorithm is showed in subsequent part.Fig.1 shows the shape retrieval results of a query horse shape. The first horse the initial shape; other horse shapes are retrieved shapes, which are in the same class with the first one. The goal of our method is to find these retrieval shapes.

In traditional classification task, labeled data is used to train the classifier, but label data manually is taxing for human. On the other hand, clustering task used non-labeled data to complete the classification or clustering problem now, but generally only distance information (or called similarity between testing data) is used in clustering method, which is not enough for the most situations. Recently a new technique named semi-supervised learning (SSL) appeared. The SSL methods mine the useful information hidden in the unlabeled data which is proved important for the classification result. Following those SSL methods, we focus on the problem that uses unlabeled data to improve the clustering task in this paper. There are many semi-supervised learning methods, like generative models [6], the method assumes a model $p(x,y) = p(y) \bullet p(x \mid y)$ where $p(x \mid y)$ is an identifiable mixture distribution. The method uses large amount of unlabeled data to identify the mixture example

This work was supported in part by Ministry of Science and Technology of China (grant No. 2006BAH02A24) and NSFC(grant No. 60873127,60903172), Education Ministry Doctoral Research Foundation of China (grant No.20070487028).

doi:10.4304/jsw.6.7.1353-1360

components. Then ideally we only need one labeled example per component to fully determine the mixture distribution. Self-training [7], the method used the most confident unlabeled data to help the classification result. Co-training [8], the method uses two different views to get the most confident data to improve the classification rate.

Figure1. Example of the shape retrieval results of a horse shape

Transductive support vector machine (TSVM) [9], the method uses unlabeled data to help the classifier to find the low-density areas in the distribution space. Other methods named graph-based method [10][11][12][13], those methods map the distance measure into a defined graph, and the vertexes of the graph reflect the data points in manifold space and the edges in the graph reflect the distance measure between the data points. These methods usually assume label smoothness over the graph. Many different graph-based methods could be found in the literatures, like Mincut [10], Gaussian random fields and harmonic functions [11], local and global consistency [12], Manifold regularization [13]; other methods about SSL could be found in the [14].

The clustering method given in this paper is coming from the SSL methods and aims at improving the clustering task. We have used the method in the text detections to find the network unhealthy culture. Traditional methods with labeled data are insensitive to the new network unhealthy culture, so undetected cases usually happen. But the classifications through traditional methods with less or no labeled data are unstable. They are not suitable for the detection of the network unhealthy culture. Our method need no training data and has good stability in practical applications. The application in the detection of network unhealthy culture indicates that our method is effective. Moreover, more experiment results show that our method is general in many applications.

Our method could be used in many fields like shape analysis [15], shape matching [16], and effectual shape retrieval [17], which show that the given method is useful.

The rest of the paper is organized as follows. We summarize the related work of semi-supervised learning and distance metric learning in Section II. Section III describes our method. The experiment results are presented in Section IV. Finally, the conclusions are provided in Section V.

II. RELATED WORK

Semi-supervised learning is an interesting area in machine learning since it uses less labeled data to improve the classification performance. Recently a great number of graph-based semi-supervised learning methods have appeared like label propagation [18], graph mincuts [19], Gaussian Fields and Harmonic Functions [20], Graph Kernels [21] [22] [23]. All of them map the distance matrix to a graph. The labeled data is represented by the vertex and the distance relation by the edge in the graph. The edge of the graph is also named transition weight. A large weight means that the label is easy to propagate. Then propagation iteration is used to get the class label of the unlabeled data, and the transition weight is invariable during the propagation process. Through our experiments, we found that if the transition weight is dynamic, we could get more useful information to help the classification performance. This is the original idea of our method.

Distance metric learning problem also has attracted amount of attention recently, it focuses on the selection of suitable distance function from a given set of distance measures. Xing et al. [24] use the solution of convex optimization problem to estimate a Mahalanobis distance. Yang [5] use the graph transduction learning approach based on label propagation to improve the retrieval performance of a given distance measure. Our method is based on [5] but extraordinary different with it for we update the graph transduction parameters on the propagation process to help us use more useful information to get similar retrieval result with [5] and rapid propagation speed.

III. ALGORITHM DESCRIPTION

A. The Framework of Learning Graph Transduction

1) Similarity Measure

In the case of shape retrieval, giving a set of shapes $X=\{x_1, \ldots, x_n\}$. The distance matrix $D = D_{ij}$ is computed by some shape distance functions, for example inner-distance (IDSC) in [5], D_{ij} is the distance between the shape x_i and x_j for $i,j = 2,\ldots,n$. Then we convert the distance to a similarity measure in order to construct an affinity matrix W. Usually, this can be done by using a Gaussian kernel:

$$w_{ij} = \exp(-\frac{D_{ij}^2}{\sigma_{ij}^2}) \tag{1}$$

In our experiments, we use an adaptive kernel size based on the mean distance to K-nearest neighbor distance of the shape x_i, x_j and C is an extra parameter. Both K and C are determined empirically.

$$\sigma_{ij} = C \cdot mean(\{knnd(x_i), knnd(x_j)\}) \tag{2}$$

The meaning in (2) represents the mean distance of the K-nearest neighbor distance of the sample x_i , x_j , and C is an extra parameter.

2)Graph Ttransduction

Firstly, we create a graph where the nodes are all the data points representing shapes. The edge between shapes

i, *j* represents their similarity W_{ij}. Larger edge weights allow labels to propagate through more easily. An $n \times n$ probabilities transition matrix P is defined as a row-wise normalized matrix W. Where P_{ij} is the probability of transit from node *i* to node *j*:

$$P_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \tag{3}$$

Label propagation is formulated as a form of propagation on a graph, where label nodes propagate to neighboring nodes according to there proximity [5], given the definition based on (4). f is an $n \times 1$ matrix as soft labels f for nodes. Initially, the query shape x_1 has $f_0(x_1)=1$, other retrieved shapes xi have $f(x_i)=0$ for $i=2,\cdots,n$.. Then we update the function f as (4). After a suitable number of iteration steps, we get the final f, whose order represents the similarity of shapes with the query shape:

$$f_{t+1}(x_i) = Pf_t = \frac{\sum_{j=1}^{n} w_{ij} f_t(x_j)}{\sum_{j=1}^{n} w_{ij}} \tag{4}$$

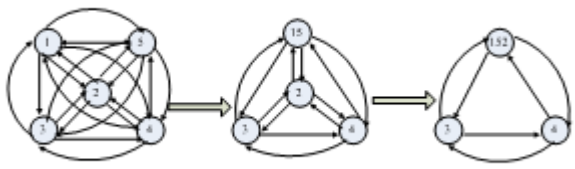
B. Shape Retrieval

1) Ranking the Shape

In [5], each row is normalized and we can know that after each iterative round, the new label of a node is decided by the rank of the iteration results f. The first of the rank is the initial query shape, and the second of it is the most similar with the query one. Based on this viewpoint, we can regard the second rank shape having the same class with the initial shape. In the next round, we change the retrieval task into finding the most similar shape with both of the two same class shapes. This operation will help us to find better result.

2) Update the Transition Probability Matrix

As mentioned above, after we sort the iterative result of f, we get the shape which has the highest similarity with the initial query shape. We use the most similar shape and the initial query shape to do next iteration. Our operation is amalgamating transition probabilities of these two shapes. The merging process can be showed in Fig.2.



(a) (b) (c)

Figure2. Transition probability map

Fig.2 (a) is the fully connected graph which can be seen as the transition probability map corresponding to the probabilistic transition matrix P, and the state 1 represents the initial query shape. After the first iteration cycle, we suppose that the most similar shape is shape 5 which is expressed as state 5 in Fig.2. Before the next iteration, we combine the transition probabilities of shape 1 and shape 5 to get a reduced matrix P， whose map is shown in (b). The operation is realized by adding the each transition probability of the fifth row to corresponding probability of the first row, the same operation to their columns. After we calculate on this reduced matrix P, the further transition probability matrix is calculated and its corresponding map is (c).

The algorithmic process can be simply divided into the following steps:

1) Based on the IDSC result, calculating the weight matrix and the transition probability matrix of the query shape by using (1), (2)*and*(3).

2) Calculating the most similar shape with the query shape by (4). *The most similar shape is in the same class with the query shape.*

3) Combining the transition probabilities of the most similar shape with the query shape, getting a reduced matrix P. The merging results are looked as transition probabilities of a new query shape.

4) According to the reduced matrix P, calculating the next most similar shape with the above- mentioned new query shapes .The step 2) *and* 3) *are expressed by* Fig.3.

5) Repeat the step 2)*to* 4) *until all the same class shapes of the query shape are found.*

6) Repeat the step 1) *to* 5) *to calculate all shapes in data set.*

C. Clustering

As mentioned above, each point x_i gets its f_i, which indicates the class information. Ranking the results of f, and finding the location of the highest increase of f value, the location indicates the divide of different clusters. Those points, which have larger values of f than the point on the divide location, have the same class label of the initial random point x_r. For remaining points, repeating above steps, we can find the points in the same class.

Our algorithm can solve the clustering problem of multiclass. The algorithm transforms multiclass clustering problem into repeated two-class clustering problems. The algorithmic process can be simply divided into the following steps:

1) For all the data points, calculating the weight matrix and the transition probability matrix by using (1), (2) *and* (3).

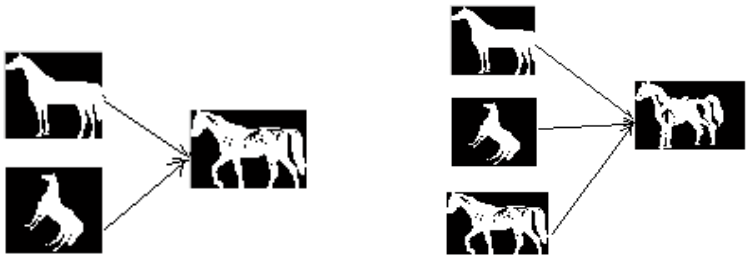
2) Random choosing the initial point xr,and giving it the label $f_0(x_r)=1$.

3) Calculating the soft point labels f by (4).

4) Ordering the f value from small to large, calculating the distance of two adjacent points, getting the point location of the maximum distance as the divide point.

5) The points, which have larger soft label values than the divide point, are in the same class.

6) To the remaining points, which have smaller soft label values than the divide point, repeat the step 1) *to* 5) *until the maximum distance has not obvious difference with other distances.*



(a) (b)

Figure.3 shows the finding process of the similar shapes to the query shape. In (a), the shape on the top left is the initial query shape, and the shape on the lower left is the most similar shape to the above shape, information of these two graph is used to strengthen the process to get the next result. In (b), the new shape which is find in the last time is used to get the next result.

TABLE Ⅰ IS THE RETRIEVAL RATES (BULL'S EYE) OF DIFFERENT METHODS

Algorithm	**CSS**[1]	**SC_TPS**[2]	**IDSC_DP**[3]	**Shape tree**[4]	**IDSC_LP**[5]	**Our Method**
Score	75.44%	76.51%	85.40%	87.70%	91.00%	90.10%

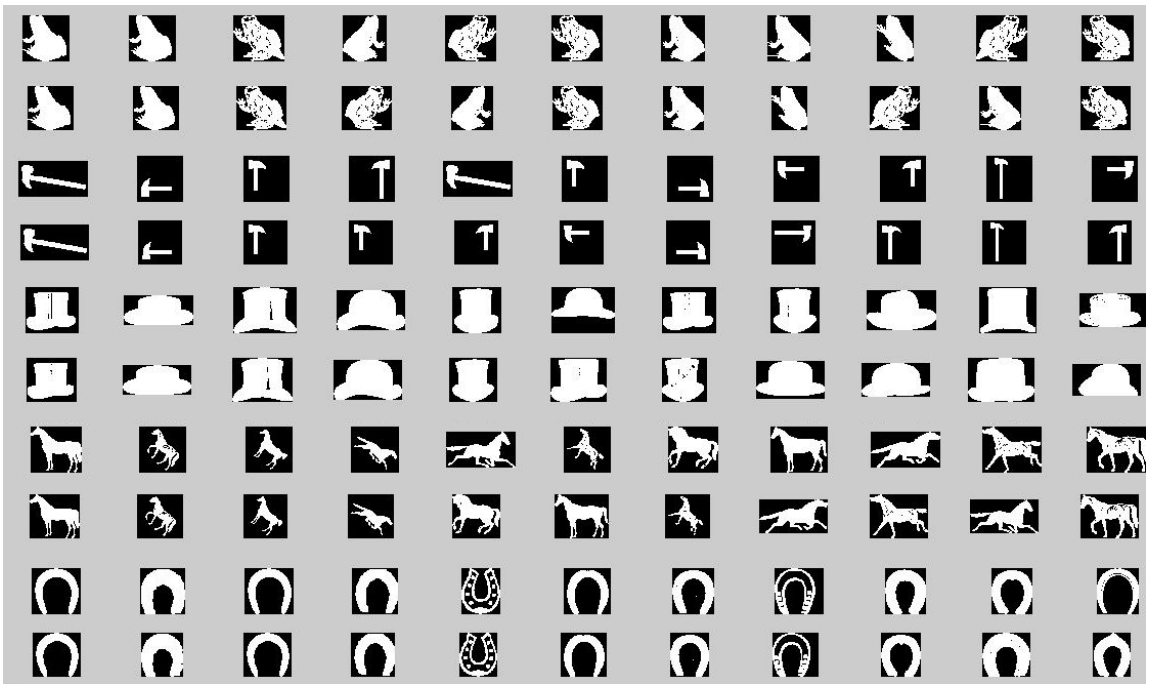


Figure 4. The comparison of IDSC_LP and our method, the shapes on odd lines are the results of [5], and the even lines are our results. The sequences are different, but the correct rates are similar.

From algorithm steps above, we can see that the initial point and the time of iteration stops need no priori information and manual intervention. That is the choice of the initial point is independent with the final clustering results. Moreover, the number of clusters need not be estimated before clustering. Next, the convergence analysis is given to prove the feasibility of the randomly choosing initial point.

D. Convergence Analysis

The core part of the algorithm is information propagation, which can be seen as the transformation of the label propagation. The random initial point can be seen as the point with implied label. The label matrix is $f= [1\ f_u]^T$, the label of the random initial point is 1, fu is the label of other unclassified point. The probabilities transition matrix P can be expressed as:

$$P = \begin{bmatrix} P_{11} & P_{1u} \\ P_{u1} & P_{uu} \end{bmatrix} \tag{5}$$

Refering the convergence proof in [25], we can get the expression:

$$f_u = (I - P_{uu})^{-1} P_{u1} \tag{6}$$

From (6), we can see that the cluster result only relates with the transition matrix *P*, and has nothing to do with the choice of initial point. Therefore, we can randomly choose the initial point. The experiments results also show that the algorithm is convergence after enough iteration steps.

IV. EXPERIMENTAL RESULTS

A. *Shape Retrieval Experiments*

The shape retrieval experiments are on MPEG-7 data set which has 1400 silhouette images grouped into 70 classes. The retrieval rate is measured by the so-called bull's eye score [5]. In our experiment, we use Gaussian Kernel function with the same parameters [5] to calculate the weight matrix and the initial matrix P. The parameters are C=0.25, k=10. Simultaneously, we retrieve 300 the most similar shapes, and construct the affinity matrix W for only those shapes. Here, W is of size 300×300 as opposed to a 1400×1400 matrix. For a query shape, after 20 iterations, we get f to find the most similar shape and the reduced matrix P to find the next similar shape. After 20 times finding, we will get all the 20 different shapes in the same class with the query shape. The number of iteration for each query shape is less than that of [5]. The number of iteration of the update weight is set empirically, and other parameters are same with [5].

In the Table Ⅰ, we list the retrieval rate of different methods. Fig.4. is the comparison of some retrieval result shapes between IDSC_LP and our method. Fig.5 is the retrieval rates and time consuming curve (bull's eye) of IDSC_LP method and our method on the MPEG-7 data set. Comparing with IDSC_LP method, our method spends half of the time to get the final result.

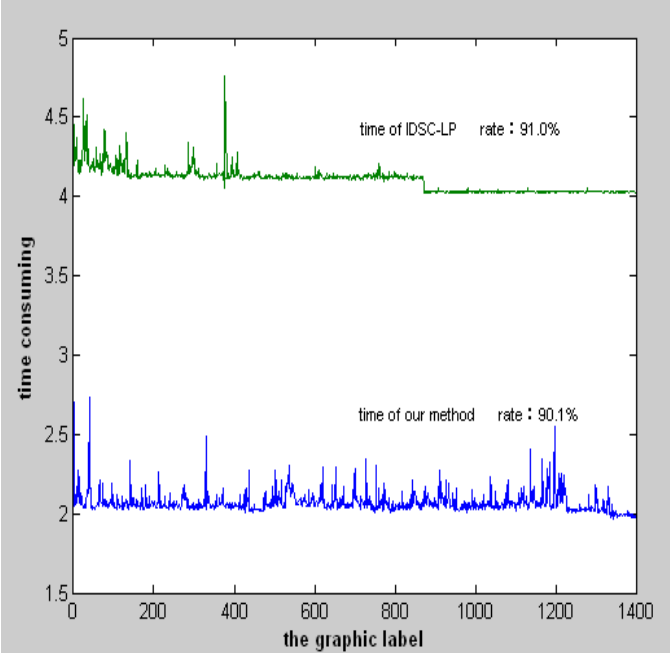


Figure 5. shows that our method spends about 2 seconds to get the retrieval result of each shape with the retrieval rate 90.10%. But the IDSC_LP needs about 4 seconds for each shape with the retrieval rate 91.00%.

B. *Clustering Experiments*

Our experiments are on synthetic datasets as [25] and [26]. Because it is easily to describe two-dimensional data in coordinate space, the clustering results can intuitively show through experiment graphs.

As show in Fig.6, the dataset is the points in two-dimensional space. First, we calculate the distance between each two points, and get the distance matrix D as the input data of the graph transduction.

Specially explaining, the results of the algorithm only depend on the distance matrix D and the density of sample points. That is the absolute space locations have no effect on the clustering results. So the choice of axis is arbitrary.

The blue star point is the random initial point of the first clustering, and the red star point is the random initial point of the second clustering. We can conclude that whichever point is chosen as initial point, the cluster result is accurate. Moreover, number of clustering need not be known in advance. The clustering process needs no priori information and manual intervention.

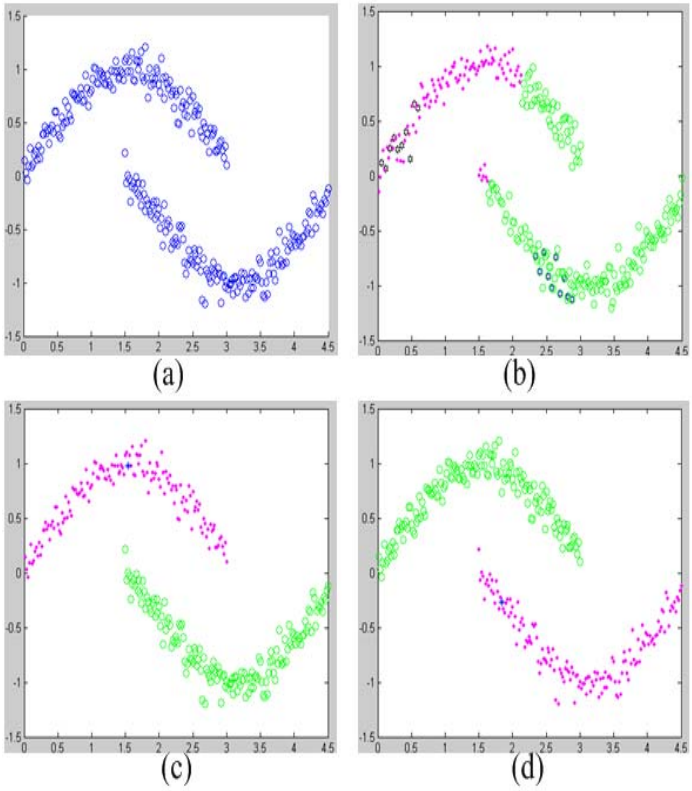


Figure 6. (a) has 301 data points, the numbers of data points for two classes are 150 and 151. (b) is the result of KNN method. Here, we have 20 training points and 281 testing points, and K=10. The training point in different cluster is marked with black and blue color. Green and purple points represent different clustering results; (c) and (d) are the clustering results of our method. In (c), the purple points are the first spin-off cluster, and the green point is the second cluster. The initial point is chosen randomly. Here, the coordinate of it is (1.54, 0.9794), which is marked as blue star point. In (d), the random initial point is also marked as blue star point, and its coordinate is (1.84,-0.2694).

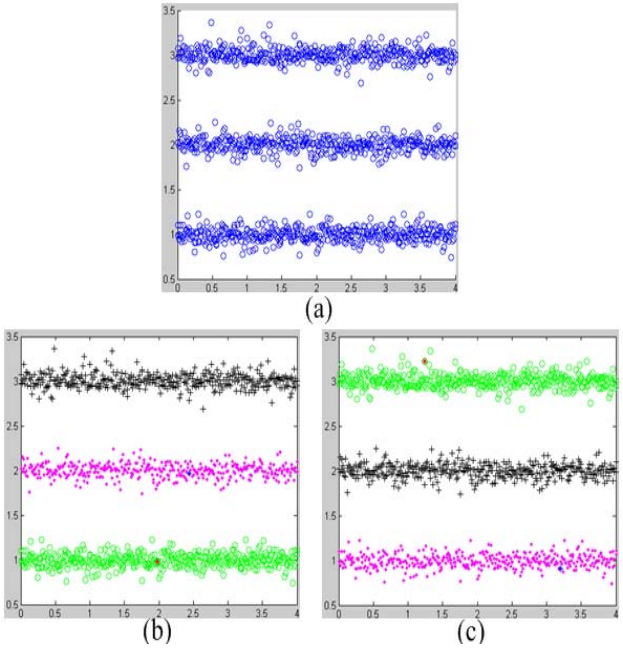


Figure 7. (a) has 1206 data points, every class has 402 data points; (b) and (c) are the clustering results of our method. Purple points are in the first spin-off cluster. Green points are in the second cluster. Black points are in the third cluster. In (b), the coordinate of random initial point in the first clustering is (2.44, 1.9711), which is marked as blue star point. The second clustering initial point coordinate is (1.98, 0.9799), which is marked as red star point. In (c), the initial points are also marked as blue and red star point, and their coordinate are (3.2, 0.9048) and (1.24, 3.2253). In the next figures, the colors in clustering results represent the same meanings as Fig.7.

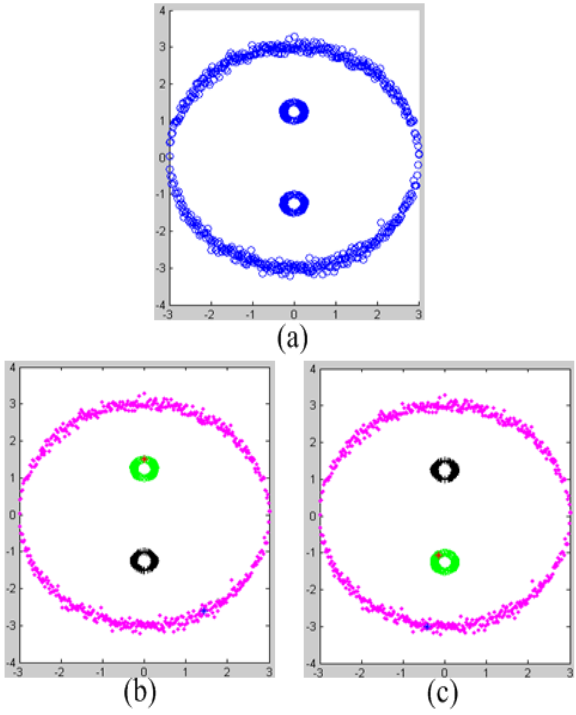


Figure 8. (a) has 1206 data points. The numbers of data points for three classes are 602,102 and 102. The meanings of (b) and(c) are the same as Fig.2. In (b), the initial points coordinate of two cluster are (1.44,-2.6071) and (0.0002, 1.5). In (c), those two coordinates are (0.42, 3.0134) and (-0.1435, -1.07).

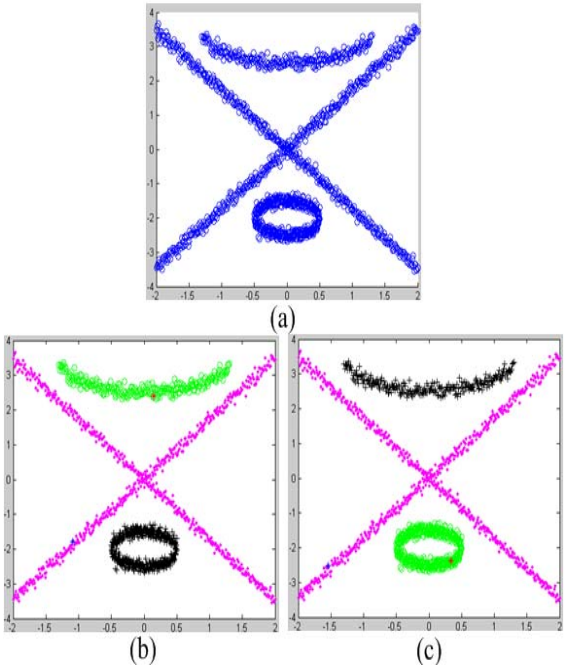


Figure.9 (a) has 1565 data points. The numbers of data points for three classes are 802,502 and 262. The meanings of (b) and(c) are same as Fig.2. In (b), the initial points coordinate of two cluster are (-1.09,-1.9811) and (0.14,2.4033). In (c), those two coordinates are (-1.54,-2.5239) and (0.332,-2.3709).

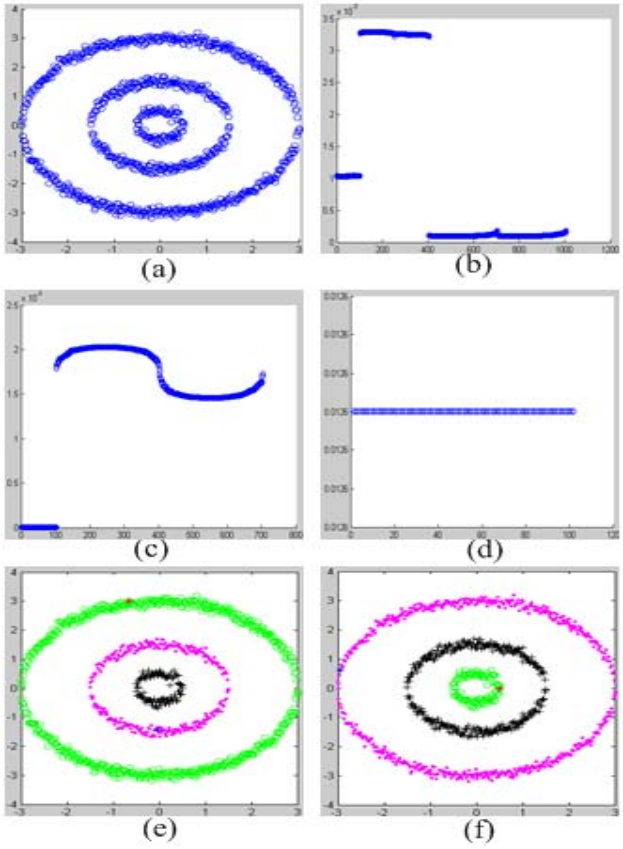


Figure10. (a) has 1006 data points. The quantities of data points for three classes are 602,302 and 102. The ordinate of (b) indicates the f values after the first clustering process. In contrast with the clustering result in (e), we can see that in (b), 302 f values have great difference with others, which correspond with the purple points in (e). The ordinate of (c) indicates the f values after the second clustering process. As we see, the remaining data points also have obvious difference in f

values. So we can get the green points in (e) are in the same class. The ordinate of (d) indicates the f values after the second clustering process. Here, we cannot find a visible difference of f values, so the clustering work stops, the remaining data points are in the same class. In (e), the coordinates of blue and red star points are (-0.04,-1.4448) and (-0.66, 3.0041). In (f), the coordinates of blue and red star points are (-2.94, 0.6168) and (0.5,-0.0143).

V. CONCLUSIONS

In this paper, we present a novel method to solve the problem caused by stationary probability. The intuition is that some useful information could be strengthened after each iteration process which helps to get rapid shape retrieval result. The experiment shows that we have better retrieval rates than mostly previously methods and slightly bad than [5], but we are much faster than [5], which is significative in shape retrieval. In the future, we focus on the work to improve the retrieval rate and solve the multi-class retrieval task. Simultaneously, the method can be used to do unsupervised clustering. Compared with the traditional unsupervised clustering method, our algorithm considers both local distance information and global density information. The algorithm is effective when the high-density zone is very dense. When the high-density zone is relatively sparse, the clustering accuracy reduces. The next work of us focuses on improving the algorithm to solve more general clustering problem. The non-metric problem given by [27] is also the future work which should be helpful for the classification result.

ACKNOWLEDGMENT

We would like to thank Xiang Bai and Xinggang Wang for providing us helpful discussion about shape retrieval, and the anonymous reviewers for their constructive comments. Special thanks go to Xingwei Yang who provided us with his detailed experimental data of their IDSC_LP method.

REFERENCES

[1] Mokhtarian, F., Abbasi, F., Kittler, J., “Efficient and robust retrieval by shape content through curvature scale space,” Smeulders, A.W.M., Jain, R. (eds.) Image Databases and Multi-Media Search, pp. 51–58.1997.

[2] S. Belongie, J. Malik, J. Puzicha,“Shape matching and object recognition using shape contexts,” IEEE Trans. PAMI 24, 705C522,2002.

[3] Ling, H., Jacobs, D.“Shape classification using the inner-distance,” IEEE Trans. PAMI 29, pp. 286–299,2007.

[4] Felzenszwalb, P.F.“Schwartz, J.: Hierarchical matching of deformable shapes,”CVPR,2007.

[5] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, “Improving shape retrieval by learning graph transduction,” ECCV, 2008.

[6] Nigam, K., Ghani, R. “Analyzing the effectiveness and applicability of co-training,” Ninth International Conference on Information and Knowledge Management, pp. 86-93.2000.

[7] Yarowsky, D. “Unsupervised word sense disambiguation rivaling supervised methods,”Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pp. 189-196.1995.

[8] Blum, A., Mitchell, T. “Combining labeled and unlabeled data with co-training,” Proceedings of the Workshop on Computational Learning Theory, 1998.

[9] Bennett, K., Demiriz, A. “Semi-supervised support vector machines,”Advances in Neural Information Processing Systems, pp. 368-374.1999.

[10] Blum, A., Chawla, S. “Learning from labeled and unlabeled data using graph mincuts,” Proc. 18th International Conf. on Machine Learning, 2001.

[11] Zhu, X., Ghahramani, Z., Lafferty, J. “ Semi-supervised learning using Gaussian fields and harmonic functions,”The 20th International Conference on Machine Learning (ICML).2003.

[12] Zhou, D., Bousquet, O., Lal, T., Weston, J., Sch¨lkopf, B. “Learning with local and global consistency,” Advances in Neural Information Processing System. 2004.

[13] Belkin, M., Niyogi, P., Sindhwani, V. “Manifold regularization: A geometric framework for learning from examples,” Technical Report TR-2004-06 University of Chicago.2004.

[14] X. Zhu, “Semi-Supervised Learning Literature Survey,” Technical Report , Dept. of Computer Sciences, Univ. of Wisconsin- Madison, 2008.

[15] Xiang Bai, Longin Jan Latecki and Wen-Yu Liu, Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution, IEEE Transactions on Pattern Analysis and Machine Intelligence 2007

[16] Xiang Bai, Xingwei Yang, Longin Jan Latecki,Wenyu Liu, and Zhuowen Tu, Learning context sensitive shape similarity by graph transduction , IEEE Transactions on Pattern Analysis and Machine Intelligence to appear

[17] Y. Zhang and Z.-H. Zhou. Non-metric label propagation. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09), Pasadena, CA, 2009, pp.1357-1362

[18] Zhu, X. , “Semi-supervised learning with graphs. In: Doctoral Dissertation. Carnegie Mellon University,”CMU–LTI–05–192,2008.

[19] Blum, A. , “& Chawla, S. :Learning from labeled and unlabeled data using graph mincuts,” ICML,2002.

[20] Zhu, X., Ghahramani, Z., Lafferty., J., “Semi-supervised learning using Gaussian fields and harmonic functions,”ICML,2003.

[21] Chapelle, O., Weston, J., & Sch¨olkopf, B. , “Cluster kernels for semi-supervised learning,”NIPS,2002.

[22] Kondor, R. I., & Lafferty, J., “ Diffusion kernels on graphs and other discrete input spaces,” ICML,2002.

[23] Smola, A., & Kondor, R., “Kernels and regularization on graphs,” COLT/KW,2003.

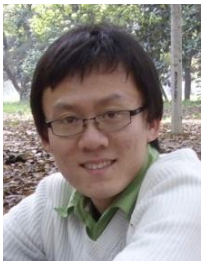
[24] Xing, E., Ng, A., Jordanand, M., Russell, S., “Distance metric learning with application to clustering with side-information,”NIPS, pp. 505–512,2003.

[25] X. Zhu,, “Semi-supervised learning with graphs,” Doctoral Dissertation. Carnegie Mellon University,”CMU–LTI–05–192,2008.

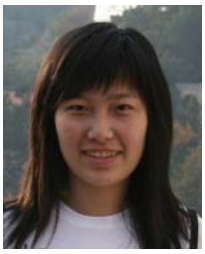
[26] L. Zelnik-Manor and P. Perona "Self-Tuning Spectral Clustering", Eighteenth Annual Conference on Neural Information Processing Systems (NIPS) ,2004.

[27] Xiang Bai and Longin Jan Latecki, Path Similarity Skeleton Graph Matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 2008.

Jun Chen received the B.S. degree in Electronics and Information Engineering from China University of Geosciences(CUG) in 2002 and the ME degree in Electronics and Communication Engineering from China University of Geosciences(CUG) in 2004. And now she is a Ph.D. student at the Department of Electronics and Information Engineering in Huazhong University of Science and Technology(HUST). Her research interests include data mining and machine learning



Yu Zhou received the BS degree in Electronics and Information Engineering from Wuhan Polytechnic University(WHPU) in 2007 and the ME degree in Electronics and Communication Engineering from Huazhong University of Science and Technology(HUST) in 2009. And now he is a Ph.D. candidate at the Department of Electronics and Information Engineering of HUST. His research interests include pattern recognition and machine learning



Yajun Gao undergraduate student in Electronics and Information Engineering from Huazhong University of Science and Technology.Her currently research interests include shape matching and shape retrieval.

Wenyu Liu received the B.S. degree in Computer Science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees, both in Electronics and Information Engineering, from Huazhong University of Science & Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate dean of the Department of Electronics & Information Engineering, HUST. His current research areas include computer graphics, multimedia information processing, and computer vision. He is a member of IEEE System, Man & Cybernetics Society.

The author is the IEEE Student Member.