Using Graph Algebra to Optimize Neighborhood for Isometric Mapping

Guihua Wen , Lijun Jiang and Nigel R Shadbolt
South China University of Technology, Guangzhou 510641, China
University of Southampton, Southampton SO17 1BJ, United Kingdom
crghwen@scut.edu.cn cssturat@sohu.com nrs@ecs.soton.ac.uk

Abstract

Most nonlinear dimensionality reduction approaches such as Isomap heavily depend on the neighborhood structure of manifold. They determine the neighborhood graph using Euclidean distance so that they often fail to nicely deal with sparsely sampled or noise contaminated data. This paper applies the graph algebra to optimize the neighborhood structure for Isomap. The improved Isomap outperforms the classic Isomap in visualization and time complexity, as it provides good neighborhood structure that can speed up the subsequent dimensionality reducing process. It also has stronger topological stability and less sensitive to parameters. This indicates that the more complicated or even time-consuming approaches can be applied to construct the better neighborhood structure whilst the whole time complexity will not raise .The conducted experiments on benchmark data sets have validated the proposed approach.

1 Introduction

Classic dimensionality reduction approaches can not be reliably applied to find the meaningful low-dimensional structures hidden in the high-dimensional observations for exploratory data analysis such as classification and visualization, so that two new approaches have recently been developed. One is locally linear embedding (LLE) that is based on local approximation of the geometry of the manifold[Roweis and Saul, 2000]. LLE has many variants, such as Laplacian eigenmaps [Belkin and Niyogi, 2003]and Hessian eigenmaps [Donoho and Grimes, 2003], incremental LLE [Kouropteva and Pietikainen, 2005], supervised LLE [de Ridder *et al.*, 2003; Kouropteva and Pietikainen, 2005], integrated LLE with classic PCA or LDA [Abusham *et al.*, 2005; Chang *et al.*, 2004], integrated LLE with SOM [Xiao *et al.*, 2005] etc. The other is Isomap that preserves the manifold geometry at all scales and has better ability to deal with nonlinear subspaces [Tenenbaum *et al.*, 2000]. It also has many variants, such as Landmark Isomap [Silva and Tenenbaum, 2003], supervised Isomap [Geng *et al.*, 2005], spatio-temporal extension of Isomap [Jenkins and Mataric, 2004], incremental extension of Isomap [Law and Jain, 2006], integrated Isomap with fuzzy LDA [Weng *et al.*, 2005] etc. The other related studies are also performed such as the selection of the optimal parameter value for LLE and Isomap [Kouropteva *et al.*, 2002; Shao and Huang, 2005], integration of LLE and Isomap [Saxena *et al.*, 2004] etc. LLE and Isomap have their own superiorities so that they have been being developed simultaneously for various context applications.

Despite Isomap performs well in many cases, it often fails to nicely deal with noisy data or sparsely sampled data [Geng *et al.*, 2005]. This is because in these cases the local neighborhood structure on which Isomap largely depends is critically distorted. This makes Isomap vulnerable to short-circuit errors that some neighors of the current point come from other different folds so that these neighbors are not nearest ones on manifold [Silva and Tenenbaum, 2003]. This can in turn lead to drastically incorrect low-dimensional embedding. Accordingly some supervised Isomap are proposed [Vlachos *et al.*, 2002; Geng *et al.*, 2005] which employ the class labels of the input data to guide the manifold learning. They improve the Isomap in classification and visualization, but they can not work when the class labels of data are not available. Due to that Isomap can not works on the disconnected neighborhood graph, some approaches to constructing connected neighborhood graph are proposed such as the k-connected or k-edge-connected spanning subgraph of the complete Euclidean graph of all data points[Yang, 2004; 2005]. This paper deals with the neighborhood graph from another perspective. It considers the implicit correlation among data points using the path algebra on the neighborhood graph, which is further applied to improve Isomap.

2 Approach to optimize the neighborhood

In many data analysis and pattern recognition tasks, similarity between two objects can be established by direct comparison and induced by mediating objects. Namely, two objects might be considered similar when they are connected by a chain of intermediate objects where all dissimilarities or distances between neighboring objects in the chain are small [Fischer and Buhmann, 2003b]. This concept can be generalized by assuming that object similarity behaves transitive in many applications. Based on this intuition, the path algebra-based clustering approach is proposed that can extract elongated structures from the data in a robust way which is particularly useful in perceptual organization [Fischer and

Buhmann, 2003a; 2003b]. This paper applies it to build the neighborhood graph for Isomap. The intuitive picture that two objects should be considered as neighbors if they can be connected by a mediating path of intermediate objects.

Let $G = (V, V \times V)$ be the complete Euclidean graph of all data points. It weights the edges using Euclidean distance d_e . A path l from v_0 to v_n in G is defined as a sequence of vertex $(v_0, \cdots, v_i, \cdots, v_n)$ such that $(v_i, v_{i+1}) \in V \times V$, $0 \leq i < n$.

Definition Let $P(v_0, v_n)$={$l|l$ be a path from v_0 to v_n in G}, we define path algebra as a set P with two binary operations $\vee$ and $\cdot$ which have the following properties [B.Carre, 1979]:

1. For all $x, y, z \in P$, the $\vee$ is idempotent, commutative, and associative: $x \vee x = x, x \vee y = y \vee x, (x \vee y) \vee z = x \vee (y \vee z)$
2. For all $x, y, z \in P$, the $\cdot$ is associative, and distributive over $\vee$: $(x \cdot y) \cdot z = x \cdot (y \cdot z), x \cdot (y \vee z) = (x \cdot y) \vee (x \cdot z), (y \vee z) \cdot x = (y \cdot x) \vee (z \cdot x)$
3. The set P contains a *zero* element ϕ and a unit element e such that: $\phi \cdot x = \phi, \phi \vee x = x = x \vee \phi, e \cdot x = x = x \cdot e$

Obviously, there are many ways to define the path algebra based on different intuitions. This paper focuses on a simple but important way to define the path algebra as follows, where for all $x, y \in \mathbb{R}$, $\phi = 0$, and $e = \infty$

1. $x \vee y \stackrel{def}{=} min(x, y)$
2. $x \cdot y \stackrel{def}{=} max(x, y)$

In this way a new distance d_m can be defined as follows [Fischer and Buhmann, 2003b]:

$$d_m(v_0, v_n) = min_{l \in P(v_0, v_n)}\{max_{(v_i, v_{i+1}) \in l}\{d_e(v_i, v_{i+1})\}\} \quad (1)$$

The difference between d_m and d_e can be illustrated in Figure 1. This difference can significantly impact the construc-

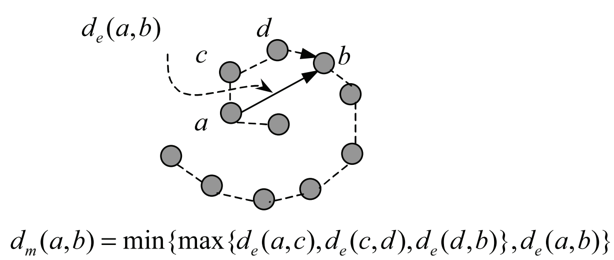

Figure 1: The difference between d_m and d_e

tion of neigborhood graph on data manifold. For example, the categories of 5-nearest neighbors of x using d_e distance,circled as in Figure 2, is

$$N^e(x) = \{square, square, square, circle, circle\}$$

Generally points belonging to the same class are often closer to each other than those belonging to different classes. Obviously the short circuiting problem happens here because points marked by *square* are not the most similar points to x

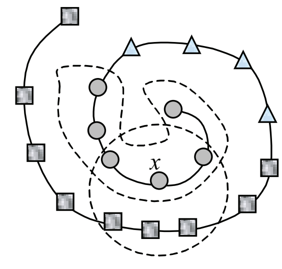

Figure 2: The different neighborhoods using d_m and d_e

in terms of categories. By contrast, if d_m distance is utilized to determine the neighbors for x, the result is

$$N^m(x) = \{circle, circle, circle, circle, circle\}$$

This is a correct result that indicates all neighbors of x lie on the manifold. It seems that d_m can pulls points belonging to the same class closer and propels those belonging to different classes further away. It ensures to preserve the intrinsic structure of the data set and therefore can be applied to improve the Isomap.

3 Proposed Isomap with Optimized Neighborhood

Isomap can perform dimensionality reduction well when the input data are well sampled and have little noise. In the presence of the noise or when the data are sparsely sampled, short-circuit edges pose a threat to Isomap by further deteriorating the local neighborhood structure of the data. Subsequently Isomap generates drastically incorrect low-dimensional embedding. To relieve this negative effect of the noise, d_m can be applied to determine the true neighborhoods as opposed to using Euclidean distance d_e. This can get better estimation of geodesic distances and in turn give lower residual variance and robustness. It also reduces the time complexity because the better neighborhood structure can speed up the subsequent optimization process.

To avoid computational expense to calculate the d_m between any two points, we find an approximate way to build the neighborhood graph using d_m that can be computed efficiently. It first applies d_e to build the neighborhood graph, and then utilize the idea of d_m to optimize this neighborhood graph. The algorithm is given as follows.

Algorithm 1: OptimizeNeighborhood(X, k, m, d)

Input: $X = \{x_i\}$ be the high dimensional data set, *k* be the neighborhood size, *m* be the scope for optimization of neighborhood, and *d* be the dimension of the projected space.

Output: The optimized neighborhood set $N = \{N(x_i)\}$ for all points.

1. Calculate the neighborhood $N(x_i)$ for any point x_i using d_e, where $N(x_i)$ is sorted ascendingly.
2. Compute n, which is the number of points in X.
3. For i=1 to n
4.   For j= 1 to k
5.   Select j-th point from $N(x_i)$, denoted as x_{ij}
6.   For p= 1 to m
7.   Select p-th point from $N(x_{ij})$, denoted as x_{ijp}

8. If $d_e(x_{ij}, x_{ijp}) < d_e(x_i, x_{ik})$ and $x_{ijp} \notin N(x_i)$ and $parent(x_{ijp}) \in N(x_i)$
9. Delete $x_i k$ from $N(x_i)$
10. Insert x_{ijp} to $N(x_i)$ ascendingly
11. Let j=1
12. Break
13. EndIf
14. EndFor
15. EndFor
16. EndFor
17. Return $N = \{N(x_i)\}$ that is the optimized neighborhoods for X

In algorithm 1, step 1 calculates the neighborhood graph using Euclidean distance, as the same as most dimensionality reduction approaches such as Isomap do. Its time complexity $< O(n^3)$. From step 3 to step 16, the neighborhood of the given point x_i determined in step 1 is optimized using d_m. It includes two cycles. Although step 11 reset *j*, *k* and *m* are neighborhood sizes so that they can be regarded as small constants. It means that optimization of neighborhood of a point can be finished in $O(1)$. Therefore the neighborhood graph can be efficiently optimized with additional time complexity $O(n)$. In Step 8, condition $parent(x_{ijp}) \in N(x_i)$ means that x_{ijp} must be reachable from x_i in the neighborhood graph determined in step 1.

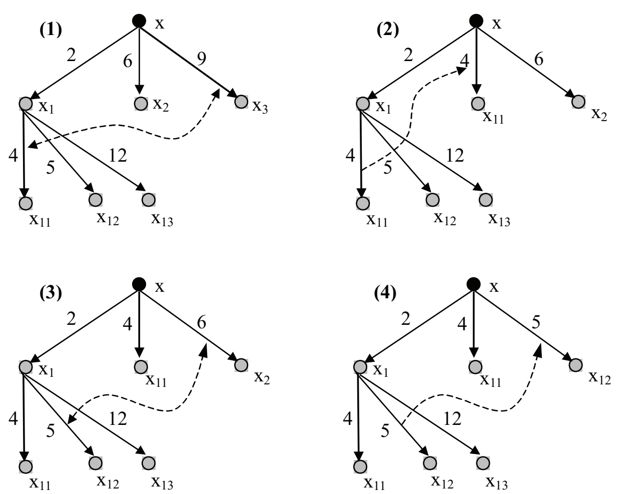

Figure 3: Illustration of optimizing neighborhood of point x

In order to illustrate the idea behind the algorithm 1, we give an example to optimize the neighborhood of point *x*, shown as Figure 3, where *k*=3 and *m*=2. Figure 3(1) shows the neighborhood graph of x determined using Euclidean distance, where only parts are described. It can be observed that x has neighbors $N(x) = \{x_1, x_2, x_3\}$. In algorithm 1, step 5 chooses its first neighbor x_1 and then step 7 selects the neighbor x_{11} for x_1. In step 8, x_{11} will be compared with the largest neighbor of x_1, namely x_3. Because the condition is satisfied, All steps between step 9 and step 12 will be executed. This results in that x_{11} will replace x_3. Consequently the neighorhood of x is optimized as $N(x) = \{x_1, x_{11}, x_2\}$. It is shown as Figure 3(2). Because the neighborhood of point x has changed, the optimization restarts from the smallest neighbor x_1 of the point x and enter the next cycle. Due to x_{11} has been in neighborhood of x, x_{12} will be explored, shown as Figure 3(3). Because the condition in step 8 is satisfied, all steps between step 9 and step 12 will be executed. This results in that x_{12} will replace x_2.Consequently the neighorhood of x is optimized as $N(x) = \{x_1, x_{11}, x_{12}\}$,shown as Figure 3(4). Due to m=2, x_{13} will never be exploited. So far all neighbors of x_1 have been tested. Next cycle will explore the second neighbor of x in step 5, namely x_{11}. It goes through all steps as exploring the first neighbor x_1 of x, which is not explained more here.

Now *OptimizeNeighborhood* can be applied to improve the basic Isomap. For hereafter comparison, we denote this improved Isomap as m2-Isomap, which is described as algorithm 2.

Algorithm 2: m2-Isomap(X, k, m, d)

Input: $X = \{x_i\}$ be the high dimensional data set, *k* be the neighborhood size, *m* be the scope for optimization of neighborhood, and *d* be the dimension of the projected space.

Output: The dimensionally reduced dataset $Y = \{y_i\}$.

1. Utilize OptimizeNeighborhood(X, k, m, d) to calculate the optimized neighborhoods for each point in input data set X, which will be applied to construct the neighbor graph in next step.
2. Construct the weighted neighbor graph $G_e = (V, E)$ for the dataset X, by connecting each point to all its *k*-nearest neighbors, where $(v_i, v_{i+1}) \in E$,if x_j is a member of *k* neighbors of x_i determined in step 1. This edge has weight $d_e(x_i, x_j)$.
3. Employ the G_e to approximately estimate the geodesic distance $d_g(x_i, x_j)$ between any pair of points as the shortest path through graph that connects them.
4. Construct *d*-dimensional embedding from G_e using MDS

Compared with the basic Isomap, m2-Isomap adds step 1 and then modifies the step 2. It should be noted that the edges of neighborhood graph are weighted by d_e instead of d_m. The d_m is only applied to optimize the neighborhood rather than applied to estimate the geodesic distance. The step 3 and step 4 remains the same as the basic Isomap. Therefore the time complexity increases from step 1 where the additional complexity is $O(n)$. However the optimized neighborhood will speed up the step 4 so as to decrease the running time of the whole algorithm. This will be proved empiricially in later experiment.

4 Experimental Results

Several experiments are conducted to compare Isomap with m2-Isomap in visualization and time complexity. Isomap uses the published matlab code [Tenenbaum *et al.*, 2000]. The m2-Isomap is also implemented in MATLAB 7.0. In experiments, there are two parameters involved. Isomap has a parameter *k* to determine the size of local neighborhood whereas m2-Isomap introduces an additional parameter *m* that controls the scope of the neighborhood to be modified, where *k*=5-15 and m =1-10 will be explored.

A. Swiss Roll Data set

Swiss Roll dataset is widely applied to compare many non-linear dimensionality reduction approaches [Roweis and Saul, 2000; Tenenbaum *et al.*, 2000; Balasubramanian and Schwartz, 2002; Geng *et al.*, 2005]. We take many random samples from the Swiss Roll surface shown as Figure 4, and do visualization experiments on them to compare Isomap with m2-Isomap in the following cases: (1)well sampled data without noise (2) well sampled data with noise (3)sparsely sampled data.

Figure 4: Swiss roll surface sample

The quality of visualization can be evaluated subjectively, while it is often quantitatively evaluated by the residual variance [Roweis and Saul, 2000; Tenenbaum *et al.*, 2000; Balasubramanian and Schwartz, 2002; Geng *et al.*, 2005]. The lower the residual variance is, the better high-dimensional data are represented in the embedded space.

Experiment 1 On well sampled noiseless data sets

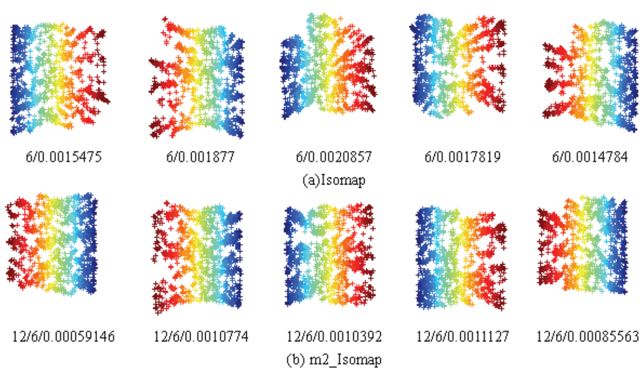

Figure 5: The mapped results of two approaches on five well sampled noiseless data sets

To make comparison on well sampled noiseless data sets, five sample data sets with 1000 points are sampled randomly from Swiss Roll surface. Firstly on the first sampled data set, two approaches are tested to choose the optimal parameters in the given scope, and then these parameters are applied to run the remaining four data sets. The mapped results from left to right corresponding to five data sets are shown as Figure 5, where the parameters and residual variance for each data set are added as caption. For example on the first data set,in Figure 5(a)Isomap, the caption ”6/0.0015475” indicates that the parameter *k* takes 6 whereas the residual variance is 0.0015475. In Figure 5(b)m2-Isomap, the caption ”12/6/0.00059146” indicates that the parameter *k* takes 12,*m* takes 6, and the residual variance is 0.0.00059146. It can be observed that on well sampled data sets without noise, m2-Isomap outperforms Isomap. It gets better mapping results and lower residual variances on any data set.

Experiment 2 On well sampled but noise contaminated data sets

To compare the topologically stability of Isomap and m2-Isomap on data sets with noise, we do experiments on the first data set with 1000 points used in experiment 1, but here it is contaminated by adding random Gaussian noise. The mean of the noise is 0 and the variance is 0.64. It can be seen from Figure 6 that Isomap gets the best result with residual variance 0.001286 when k=6, while m2-Isomap gets the best result with the lower residual variance 0.0010773 with parameters k=12 and m=6. Obviously m2-Isomap outperforms Isomap in terms of visualization and residual variance. It can be also observed from Figure 6 that as *k* increases, the results of Isomap change drastically, but those of m2-Isomap do not change much. This indicate that m2-Isomap is also less sensitive to *k* than Isomap.

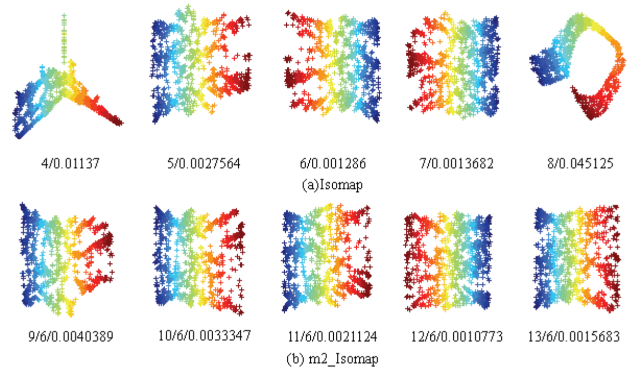

Figure 6: The mapped results of two approaches on a well sampled but noise contaminated data set

Experiment3 On sparsely sampled data sets

Generally, in sparsely sampled data sets, the Euclidean distance between points in neighborhood becomes larger as compared to the distance between different folds of the manifold. This easily makes two approaches faces the problem of short-circuiting. This experiment makes comparison between two approaches about robust to samples with different sampling density. Five data sets with 300,400,500,600 and 700 points respectively are randomly sampled from Swiss Roll surface. Two approaches are tuned to select the best visualization results on each data set. It can be observed from Figure 7 that m2-Isomap outperforms Isomap on each datset in terms of visualization performance and residual variance.

B. Face image data set

This data set consists of 698 images (each contains 64×64 pixels) of a human face rendered with different poses and lighting directions [Tenenbaum *et al.*, 2000]. To compare two approaches on sparsity data set, we choose 350 (half of the data set)images from this data set to form a new data set. The

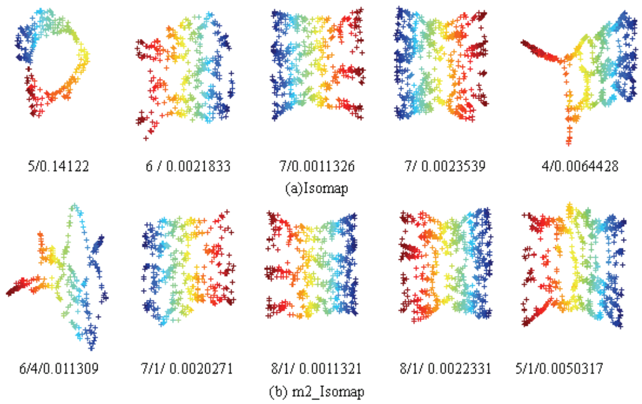

Figure 7: The mapped results of two approaches on five sparsely sampled data sets

mapped results are shown in Figure 8 and Figure 9. In fig-

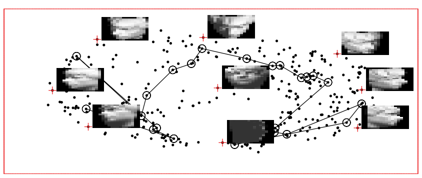

Figure 8: Isomap (k=5)

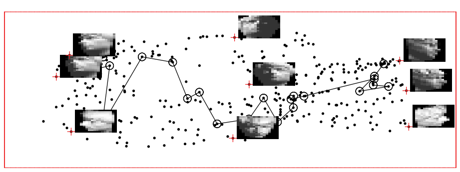

Figure 9: m2-Isomap (k=5 m=1)

ures, the x-axis represents the left-right poses of the faces, and the y-axis represents the up-down poses of the faces. The corresponding points of the successive images from left to right in the middle are marked by circles and linked by lines. The nine critical face samples are marked by plus at the left-bottom corner of each image indicates the point representing the image. It can be observed from these figures that Isomap can hardly reveal the different face poses. The middle left-right line is heavily curved, and the arrangement of the nine face samples is tangle some. By contrast, m2-Isomap puts the middle left-right line better. The nine face samples are also mapped to the approximately right positions corresponding to the face poses. These indicate that m2-Isomap performs better than Isomap.

This also can be supported from their residual variances. It can be observed from Figure 10 that m2-Isomap gets lower

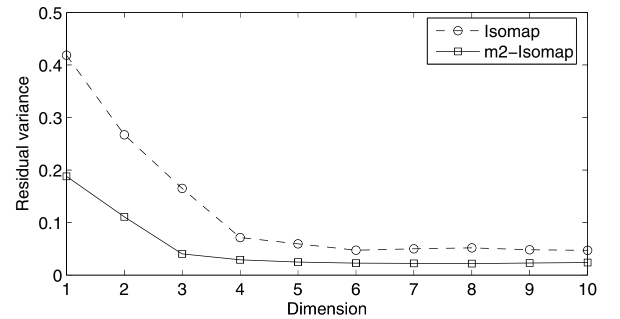

Figure 10: The residual variance of two approaches along the dimension on face images varying in pose and illumination (N=350).

residual variances and the correct intrinsic dimensionality of data set, while Isomap gives the higher estimate to intrinsic dimensionality than that of the data set. The intrinsic dimensionality of the data can be estimated by looking for the ”elbow” at which this curve ceases to decrease significantly with added dimensions.

C. Time complexity comparison

We generate 10 data sets with the same size randomly sampled from Swiss Roll surface. Two approaches run on these ten data sets respectively and then average running time for two approaches are calculated. And then the experiments are performed on different sizes: 500 900 1300 1700 2100 2500 2900 3300. It can be observed from Table 1 that m2-Isomap also exceeds the Isomap in time complexity. The larger the data set size is, the superiorities of m2-Isomap is more obvious. This is because good neighborhood structure of data set is beneficial to speed up the subsequent optimization process.

5 Conclusion

This paper applies the graph algebra to build the neighborhood graph for Isomap. The improved Isomap outperforms the classic Isomap in visualization and time complexity. It also has stronger topological stability and less sensitive to parameters. This indicates that the more complicated or even time-consuming approaches can be applied to construct the better neighborhood structure whilst the whole time complexity will not raise. Because the transitivity of similarity is not absolutely correct for any data set, in the future, we will apply the fuzzy theory and probability theory to define the graph algebra and then to build the neighborhood graph. And besides, the proposed approach should be combined with k-edge-connected spanning subgraph so as to guarantee to build the connected neighborhood graph for any data set.

Acknowledgments

This work is supported by Natural Science Foundation of China(60003019),Hubei Science and Technology Project (2005AA101C17),and UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01.

Table 1: Comparison of time complexity (seconds)

$Datasize \rightarrow$	500	900	1300	1700	2100	2500	2900	3300
Isomap	9.7	56.2	166.9	375.5	707.9	1205.0	1913.2	2833.0
m2-Isomap	9.5	55.8	166.4	373.4	708.2	1199.4	1907.8	2819.3

References

[Abusham *et al.*, 2005] E.E. Abusham, D. Ngo, and A. Teoh. Fusion of locally linear embedding and principal component analysis for face recognition (fllepca). *Lecture Notes in Computer Science*, 3687:326–333, 2005.

[Balasubramanian and Schwartz, 2002] M. Balasubramanian and E.L. Schwartz. The isomap algorithm and topological stability. *Science*, 295(4):7, 2002.

[B.Carre, 1979] B.Carre. *Graphs and Networks*. Oxford University Press, 1979.

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

[Chang *et al.*, 2004] J.P. Chang, H.X. Shen, and Z.H. Zhou. Unified locally linear embedding and linear discriminant analysis algorithm (ullelda) for face recognition. *Lecture Notes in Computer Science*, 3338:296–304, 2004.

[de Ridder *et al.*, 2003] D. de Ridder, O. Kouropteva, and O. Okun. Supervised locally linear embedding. *Lecture Notes in Artificial Intelligence*, 2714:333–341, 2003.

[Donoho and Grimes, 2003] D.L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding, techniques for high-dimensional data. *Proc.Natl.Acad. Sci. U. S. A*, 100(10):5591–5596, 2003.

[Fischer and Buhmann, 2003a] B. Fischer and J.M. Buhmann. Bagging for path-based clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(1):1411–1415, 2003.

[Fischer and Buhmann, 2003b] B. Fischer and J.M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(4):513–518, 2003.

[Geng *et al.*, 2005] X. Geng, D.C. Zhan, and Z.H. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on systems, man, and cybernetics-part B: cybernetics*, 35(6):1098–1107, 2005.

[Jenkins and Mataric, 2004] O. Jenkins and M. Mataric. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 441–448, Alberta, Canada, 2004.

[Kouropteva and Pietikainen, 2005] O. Kouropteva and M. Pietikainen. Incremental locally linear embedding. *Pattern Recognition*, 38:1764–1767, 2005.

[Kouropteva *et al.*, 2002] O. Kouropteva, O. Okun, and M. Pietikainen. Selection of the optimal parameter value for the locally linear embedding algorithm. In *Proceedings of 1th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 359–363, Singapore, 2002.

[Law and Jain, 2006] M.H.C. Law and A.K. Jain. Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:377–391, 2006.

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

[Saxena *et al.*, 2004] A. Saxena, A. Gupta, and A. Mukerjee. Non-linear dimensionality reduction by locally linear isomaps. In *Proceedings of 11th International Conference on Neural Information Processing*, pages 1038–1043, Calcutta, India, 2004.

[Shao and Huang, 2005] C. Shao and H.K. Huang. Selection of the optimal parameter value for the isomap algorithm. *Lecture Notes in Artificial Intelligence*, 3789:396–404, 2005.

[Silva and Tenenbaum, 2003] V.D. Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Neural Information Processing Systems*, 15:705–712, 2003.

[Tenenbaum *et al.*, 2000] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Vlachos *et al.*, 2002] M. Vlachos, C. Domeniconi, and D. Gunopulos et al. Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002.

[Weng *et al.*, 2005] S. Weng, C. Zhang, and Z. Lin. Exploring the structure of supervised data by discriminant isometric mapping. *Pattern Recognition*, 38:599– 601, 2005.

[Xiao *et al.*, 2005] J. Xiao, Z.T. Zhou, and D.W. Hu. Self-organized locally linear embedding for nonlinear dimensionality reduction. *Lecture Notes in Computer Science*, 3610:101–109, 2005.

[Yang, 2004] L. Yang. k-edge connected neighborhood graph for geodesic distanc estimation and nonlinear data projection. In *Proc. 17th Intl Conf. PatternRecognition*, pages 196–199, 2004.

[Yang, 2005] L. Yang. Building k edge-disjoint spanning trees of minimum total length for isometric data embedding. *IEEE Transactions of Pattern Analysis and Intelligence*, 27:1680–1683, 2005.