Proceedings of 2005 4th IEEE International Conference on
Development and Learning

Kernel Isomap on Noisy Manifold

Heeyoul Choi, Seungjin Choi
Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu
Pohang 790-784, Korea
Email: {hychoi, seungjin}@postech.ac.kr

***Abstract*— In the human brain, it is well known that perception is based on similarity rather than coordinates and it is carried out on the manifold of data set. Isomap [1] is one of widely-used low-dimensional embedding methods, where approximate geodesic distance on a weighted graph is used in the framework of classical scaling (metric MDS). In this paper we consider two critical issues missing in Isomap: (1) generalization property; (2) topological stability and present our *robust kernel Isomap* method, armed with such two properties. The useful behavior and validity of our robust kernel Isomap, is confirmed through numerical experiments with several data sets including real world data.**

I. Introduction

Human brain distinguishes an object from others based on the similarity or (dissimilarity) between objects. In the human brain, perception process is based on similarity rather than coordinates. Especially, the recognition of a new data is based on the similarity between the new data point and training data points and the perception is carried out on the manifold of data set using the similarities. Recently, various manifold learning methods have been developed in machine learning community and their wide applications stated to draw an attention in pattern recognition, signal processing, robotics as well as in developmental learning community. Isomap is one of representative isometric mapping methods, which extends metric multidimensional scaling (MDS), considering approximate geodesic distance on a weighted graph, instead of Euclidean distance [1].

Classical scaling (that is one of metric MDS) with Euclidean distances as the dissimilarities, is explained in the context of PCA [2], so that it provides a generalization property (or projection property) where new data points (which do not belong to a set of training data points) can be embedded in a low-dimensional space, through a mapping computed by PCA. In the same manner, a non-Euclidean dissimilarity can be used, although there is no guarantee that the eigenvalues are nonnegative.

The approximate geodesic distance matrix used in Isomap, can be interpreted as a kernel matrix [3]. However, the kernel matrix based on the doubly centered approximate geodesic distance matrix, is not always positive semidefinite. We mainly exploit a constant-adding method such that the geodesic distance-based kernel matrix is guaranteed to be positive semidefinite. Mercer kernel-based Isomap algorithm has a generalization property so that test data points can be successfully projected using a kernel trick as in kernel PCA [4], whereas general embedding methods (including Isomap) do not have such a property.

We can use this projection property to make retrieval system where query is a data point such as image. For example, in the information retrieval problem, the document itself could be a query in contrast to the conventional information retrieval system. This retrieval is also important to recognize a new point based on the training data points because we compare new data with training data sets on the manifold in order to understand it. This is also important in robot vision. When a robot see an object, the robot can recognize it based on the manifold which is constructed from training data set.

In addition to the positive semidefinite problem, some noise data make it very difficult to find the low-dimensional manifold, because of short-circuit edges which connect directly two subspaces through the outside of the manifold [5]. For example, one noise data point between two surfaces can connect the surfaces if the number or size of neighborhood is large enough. To overcome this situation, we use the concept of network flow in graph. First, we assume that in the original manifold the data points are uniformly distributed and the network flows have proper values. Removing the noise point which have extraordinary value of flow, we can recover the smooth nonlinear manifold from the noisy data set.

In this paper, we present kernel Isomap in Sec. II. Then, we suggest a new approach, network flow, for the topological stability in Sec. III. In Sec. IV, numerical experimental results confirm the validity and high performance of our *robust kernel Isomap* algorithm on noisy data set and show the retrieval system using projection property.

II. Kernel Isomap

A. Isomap as Kernel Method

The classical scaling, that is one of metric MDS, is a method of low-dimensional embedding based on pairwise similarity between data points. In general, Euclidean distance is used as a measure of dissimilarity (or similarity) in MDS. The basic idea in Isomap [1] is to use geodesic distances on a

0-7803-9225-6/05/$20.00 ©2005 IEEE.

neighborhood graph in the framework of the classical scaling, in order to incorporate with the manifold structure, instead of subspace. The sum of edge weights along the shortest path between two nodes, is assigned as geodesic distance. The top n eigenvectors of the geodesic distance matrix, represent the coordinates in the n-dimensional Euclidean space.

As pointed out in [6], metric MDS can be interpreted as kernel PCA. In a similar fashion, Isomap can be considered as a kind of kernel method [3]. We can take the approximated distances $\boldsymbol{D}$ used in Isomap and consider the following kernel:

$$\boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{H}, \tag{1}$$

where $\boldsymbol{D}^2$ means element-wise square of $\boldsymbol{D}$, $\boldsymbol{H}$ is the centering matrix, given by $\boldsymbol{H} = \boldsymbol{I} - \frac{1}{N}\boldsymbol{e}\boldsymbol{e}^T$, and $\boldsymbol{e} = [1,\ldots,1]^T \in \mathbb{R}^N$.

However, this kernel is not guaranteed to be positive semidefinite. The reason why the kernel matrix of Isomap is not positive definite in the smooth manifold, is mainly the approximation of the geodesic distance and noise. So, we propose kernel Isomap which has the noise robustness and projection property.

B. Kernel Isomap

Given N objects with each object being represented by an m-dimensional vector $\boldsymbol{x}_i$, $i = 1,\ldots,N$, the kernel Isomap algorithm finds an implicit mapping which places N points in a low-dimensional space. In contrast to Isomap, the kernel Isomap can project test data points onto a low-dimensional space, as well, through a kernel trick. The kernel Isomap mainly exploits the additive constant problem, the goal of which is to find an appropriate constant to be added to all dissimilarities (or distances), apart from the self-dissimilarities, that makes the matrix $\boldsymbol{K}$ to be positive semidefinite. In fact, the additive constant problem was extensively studied in the context of MDS [7], [2] and recently in embedding [8]. The matrix $\widetilde{\boldsymbol{K}}$ induced by a constant adding method, has a Euclidean representation and becomes a Mercer kernel matrix. The kernel Isomap algorithm is summarized below.

Algorithm Outline: Kernel Isomap

Step 1. Identify the k nearest neighbors (or ϵ-ball neighborhood) of each input data point and construct a neighborhood graph where edge lengths between points in a neighborhood are set as their Euclidean distances.

Step 2. (Shortest Path Problem) Compute approximate geodesic distances, D_{ij}, containing shortest paths between all pairs of points and define $\boldsymbol{D}^2 = \left[D_{ij}^2\right] \in \mathbb{R}^{N\times N}$.

Step 3. Construct a kernel matrix $\boldsymbol{K}(\boldsymbol{D}^2)$ based on the approximate geodesic distance matrix $\boldsymbol{D}^2$ as Eq. (1).

Step 4. Compute the largest eigenvalue, c^*, of the matrix

$$\begin{bmatrix} \boldsymbol{0} & 2\boldsymbol{K}(\boldsymbol{D}^2) \\ -\boldsymbol{I} & -4\boldsymbol{K}(\boldsymbol{D}) \end{bmatrix}, \tag{2}$$

and construct a Mercer kernel matrix $\widetilde{\boldsymbol{K}} = \widetilde{\boldsymbol{K}}(\widetilde{\boldsymbol{D}^2})$ by

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K}(\boldsymbol{D}^2) + 2c\boldsymbol{K}(\boldsymbol{D}) + \frac{1}{2}c^2\boldsymbol{H}, \tag{3}$$

where $\widetilde{\boldsymbol{K}}$ is guaranteed to be positive semidefinite for $c \geq c^*$.

Step 5. Compute the top n eigenvectors of $\widetilde{\boldsymbol{K}}$, which leads to the eigenvector matrix $\boldsymbol{V} \in \mathbb{R}^{N\times n}$ and the eigenvalue matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$.

Step 6. The coordinates of the N points in the n-dimensional Euclidean space are given by $\boldsymbol{Y} = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^T$.

A main difference between the conventional Isomap and our kernel Isomap, lies in Step 4 which is related to the *additive constant problem* that was well studied in metric MDS. The additive constant problem aims at finding a value of constant, c, such that the dissimilarities defined by

$$\widetilde{D}_{ij} = D_{ij} + c(1 - \delta_{ij}), \tag{4}$$

have a Euclidean representation for all $c \geq c^*$ and δ_{ij} is the Kronecker delta. Substituting $\widetilde{D}_{ij}$ for D_{ij} in Eq. (4) gives Eq. (3). For $\widetilde{\boldsymbol{K}}$ to be positive semidefinite, it is required that $\boldsymbol{x}^T\widetilde{\boldsymbol{K}}\boldsymbol{x} \geq 0$ for all $\boldsymbol{x}$. Cailliez showed that c^* is given by the largest eigenvalue of the matrix Eq. (2) (see Sec. 2.2.8 in [2]). Though several other constant-shifting methods can also be used to make the geodesic kernel matrix to be positive semidefinite, in this paper we use Eq. (4).

The matrix $\widetilde{\boldsymbol{K}}$ is a Mercer kernel matrix, so its (i,j)-element is represented by

$$\widetilde{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi^T(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j), \tag{5}$$

where $\phi(\cdot)$ is a nonlinear mapping onto a feature space or a low-dimensional manifold. The coordinates in the feature space can be easily computed by projecting the centered data matrix onto the normalized eigenvectors of the sample covariance matrix in the feature space,

$$\boldsymbol{C} = \frac{1}{N}\left(\boldsymbol{\Phi}\boldsymbol{H}\right)\left(\boldsymbol{\Phi}\boldsymbol{H}\right)^T, \tag{6}$$

where $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1),\ldots,\phi(\boldsymbol{x}_N)]$.

C. Generalization Property

As in kernel PCA, we can project a test data point $\boldsymbol{t}_l$ in the low-dimensional space by

$$[\boldsymbol{y}_l]_i = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{N} [\boldsymbol{v}_i]_j k(\boldsymbol{t}_l, \boldsymbol{x}_j), \tag{7}$$

where $[\cdot]_i$ represents the ith element of a vector and $\boldsymbol{v}_i$ is the ith eigenvector of $\widetilde{\boldsymbol{K}}$. The geodesic kernel for the test data point, $k(\boldsymbol{t}_l, \boldsymbol{x}_j)$, in Eq. (7), is constructed by the kernel matrix Eq. (3) for a set of training data points and geodesic distances, D_{lj}, between test data points $\boldsymbol{t}_l$ and all training data points $\boldsymbol{x}_j$, $j = 1, \ldots, N$. As in Eq. (4), D_{lj} is also modified by

$$\widetilde{D}_{lj} = D_{lj} + c. \tag{8}$$

Note that the geodesic distance $\widetilde{D}_{lj}$ in the feature space, has a Euclidean representation. Hence, the following relation holds:

$$\widetilde{D}_{lj}^2 = [\phi(\boldsymbol{t}_l) - \phi(\boldsymbol{x}_j)]^T [\phi(\boldsymbol{t}_l) - \phi(\boldsymbol{x}_j)]. \tag{9}$$

Taking into account that $\{\phi(\boldsymbol{x}_j)\}$ are centered, we have

$$\frac{1}{N} \sum_{j=1}^{N} \widetilde{D}_{lj}^2 = \phi^T(\boldsymbol{t}_l)\phi(\boldsymbol{t}_l) + \frac{1}{N} \sum_{j=1}^{N} \phi^T(\boldsymbol{x}_j)\phi(\boldsymbol{x}_j). \tag{10}$$

Then, it follows from Eq. (9) and Eq. (10) that the kernel for the test data point $\boldsymbol{t}_l$, is computed as

$$\begin{aligned} k(\boldsymbol{t}_l, \boldsymbol{x}_j) &= \phi^T(\boldsymbol{t}_l)\phi(\boldsymbol{x}_j) \\ &= -\frac{1}{2}\left(\widetilde{D}_{lj}^2 - \widetilde{K}_{jj} - \frac{1}{N}\sum_{i=1}^{N} \widetilde{D}_{li}^2 - \widetilde{K}_{ii}\right) \end{aligned} \tag{11}$$

The L-Isomap [9] involving landmark points and an out-of-sample extension of Isomap (and other manifold learning methods) [10], also shares a similar spirit with our projection method in Eq. (7) and Eq. (11). However, their algorithm depends on only the geodesic distances between test data point and training data points and does not consider the kernel matrix. That is, their formula which ignores the kernel part in Eq. (11), can be considered as a special case of ours.

In addition, they uses geodesic distances, D_{ij}, (not guaranteed to have a Euclidean representation in the feature space) instead of constant-shifted distances, $\widetilde{D}_{ij}$. Therefore, our kernel Isomap is a natural extension of Isomap using a kernel trick as in kernel PCA.

D. Kernel Isomap vs. Kernel PCA

Relationship between MDS and PCA is essential to understand the relationship between kernel Isomap and kernel PCA. For MDS, the Euclidean distance D_{ij} from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is

$$D_{ij}^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\boldsymbol{x}_i - \boldsymbol{x}_j). \tag{12}$$

Let the inner product matrix be $\boldsymbol{B}$, where

$$\boldsymbol{B}_{ij} = \boldsymbol{x}_i^T \boldsymbol{x}_j. \tag{13}$$

From some calculations, like Eq. (1), we can get

$$\boldsymbol{B} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{H} = \boldsymbol{V}_M \boldsymbol{\Lambda}_M \boldsymbol{V}_M^T = \boldsymbol{X}^T\boldsymbol{X}, \tag{14}$$

where $\boldsymbol{V}_M$ and $\boldsymbol{\Lambda}_M$ are the eigenvector and the eigenvalue matrix of $\boldsymbol{B}$, respectively. Finally, the recovered data $\boldsymbol{Y}_M = \boldsymbol{\Lambda}_M^{1/2}\boldsymbol{V}_M^T$. For PCA,

$$\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{V}_P \boldsymbol{\Lambda}_P \boldsymbol{V}_P^T, \tag{15}$$

where $\boldsymbol{V}_P$ and $\boldsymbol{\Lambda}_P$ are the eigenvector and the eigenvalue matrix of $\boldsymbol{X}\boldsymbol{X}^T$, respectively. The recovered data $\boldsymbol{Y}_P = \boldsymbol{V}_P^T\boldsymbol{X}$. With $\boldsymbol{V}_P = \boldsymbol{X}\boldsymbol{V}_M$, the relation of $\boldsymbol{Y}_P$ and $\boldsymbol{Y}_M$ is following.

$$\boldsymbol{Y}_P = \boldsymbol{V}_P^T\boldsymbol{X} = (\boldsymbol{X}\boldsymbol{V}_M)^T\boldsymbol{X} = \boldsymbol{V}_M^T\boldsymbol{B} = \boldsymbol{\Lambda}_M^{1/2}\boldsymbol{Y}_M. \tag{16}$$

That is, in the case of Euclidean distance, PCA and MDS are equivalent to each other with the ambiguity of the scale.

In the case of kernel Isomap and kernel PCA, kernel Isomap has a positive semidefinite kernel matrix and has a projection property like kernel PCA. In kernel PCA [4], however, nonlinear kernel function should be chosen very carefully to make kernel matrix. Different kernel function makes different performance which depends on how properly the kernel reflects the manifold of data set. To choose the proper kernel for data set, the prior knowledge is required. Especially, for complex data set such as swiss roll data, it is very difficult to find the proper kernel function. In kernel Isomap, the kernel matrix Eq. (1) is obtained from the geodesic distances, so we do not need to choose a certain kernel function. Moreover, the kernel matrix reflects the manifold properly.

III. Topological Stability

It was pointed out in [5] that Isomap could be topologically unstable, depending on the neighborhood size in constructing a neighborhood graph. The size of neighborhood is also important in locally liner embedding (LLE) [11]. A relatively large neighborhood size might result in short-circuit edges which destruct the manifold structure of data points. An easy way to avoid this short-circuit edges, is to decrease the neighborhood size, but determining the size is not a easy job. Moreover, a too small neighborhood size could produce disconnected manifolds. The nodes causing short-circuit edges are considered as outliers. Here we present a heuristic method of possibly eliminating such critical outliers, in order to make the kernel Isomap to be robust.

To this end, we consider network flows and define the total flow for each node, in terms of the number of shortest paths passing through the node. We claim that nodes causing short-circuit edges have enormous total flow values. Thus, evaluating total flow value for each node is a preprocessing step, to eliminate critical outlier nodes.

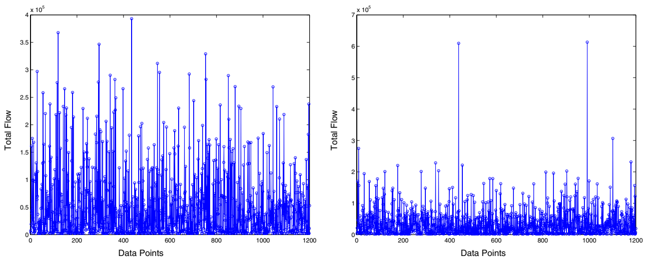
A. Network Flow

Definition 1: [12] Let $\mathcal{G}$ be a directed graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. A *network flow* is a non-negative function defined on the edges; the value $\eta(\epsilon_k)$ is the value of flow in the edge ϵ_k.
In this paper we assign the number of Dijkstra geodesic paths (shortest paths) passing on the edge, as a network flow value, $\eta(\epsilon_k)$.

Definition 2: The *total flow*, $f(v_k)$, of a node v_k is the sum of the network flows of edges connecting to the node v_k, i.e.,

$$f(v_k) = \sum_{v_i \in \mathcal{N}_k} \eta(\epsilon(i,k)), \tag{17}$$

where $\mathcal{N}_k$ is the neighborhood of the node v_k (i.e., a set of nodes connecting to the node v_k) and $\epsilon(i,k)$ denotes the edge connecting v_i and v_k.



(a) (b)

Fig. 1. Total flow of each data points for the case of *noisy Swiss Roll* data (with isotropic Gaussian noise with variance 0.25). The number of data points is 1200 (a) k-nearest neighborhood size is 5. (b) k-nearest neighborhood size is 6. Some points having extremely large total flow are considered as noise data.

Here, we assume that on a manifold, data points are scattered uniformly. So, if the total flow of a point is extremely high, the point can be considered as an noise which is off the manifold and connects directly two subspaces. In Fig. 1, we can see the changes of the total flows as the number of neighbor increases. In (a), the total flows are well distributed, whereas we can see extremely high values of total flow in (b). In this case, not only conventional Isomap but also kernel Isomap cannot find the true manifold because of the noise and the projected data space is distorted (See Fig. 9 (a)). So, in Fig. 1 (b) we consider 3 data points as noise data which have extremely high total flow compared with other points. It is not clear to determine the optimal threshold of total flow which separates critical outliers from data points. Through empirical study, we used the half of the largest total flow as a threshold only for the case that data points are abnormally scattered (with several erroneous peaks).

In our algorithm, when it finds the geodesic distances it saves the paths. So, it does not require any extra computational time to the algorithm which finds shortest path.

B. Robust Kernel Isomap Algorithm

Table I shows the outline of the final algorithm. This algorithm overcomes the short-circuit edge problem and gives total noise-robust property as well as projection property. Before calculating the kernel matrix, it eliminates the noise data point which have extremely high total flow value. Then, the geodesic distance matrix is calculated again and kernel Isomap is used to find an manifold.

TABLE I
ROBUST KERNEL ISOMAP.

Construct a neighborhood graph.
Calculate geodesic paths.
Calculate total flows of nodes.
Eliminate outliers having extremely high total flow values.
Apply the Kernel Isomap to this preprocessed data set as in Sec. II-B.

IV. NUMERICAL EXPERIMENTS

A. Kernel Isomap

We compared our kernel Isomap algorithm to the conventional Isomap algorithm, using Swiss roll data that was also used in Isomap. Noisy Swiss roll data was generated by adding isotropic Gaussian noise with zero mean and 0.25 variance (see Fig. 2 (a)). In the training phase, 1200 data points were used and the neighborhood graph was constructed using $k = 4$ nearest neighbors of each data point, respectively. As in Isomap, the shortest paths were computed using the Dijkstra's algorithm, in order to calculate approximate geodesic distances.

An exemplary embedding result (onto 3-dimensional feature space) for Isomap and kernel Isomap, is shown in Fig. 2 (b) and (c). The generalization property of our kernel Isomap is shown in Fig. 2 (d) where 3000 test data points are embedded with preserving local isometry well. In this figure, comparing (c) with (b), we can also see the noise robustness of kernel Isomap. Even though the conventional Isomap also looks like robust algorithm in 2 dimensional manifold, in 3 dimensional space, it is not robust any longer. The manifold in (b) is not smooth while (c) is.

Applications of Projection Property: Projection property can be applied to build a retrieval system where query is a data point such as image whereas traditional information retrieval is based on keywords. Our retrieval system searches similar data points with the projected point of query data on low-dimensional space. Actually, new data is recognized by the relationship with training data set on the manifold which is constructed by training data set. We built this system using triangle data and US Postal Service (USPS) handwritten digits.

First, triangle data set is composed of totally 400 triangle images (20 shapes of triangle and 20 rotations from 0^o to

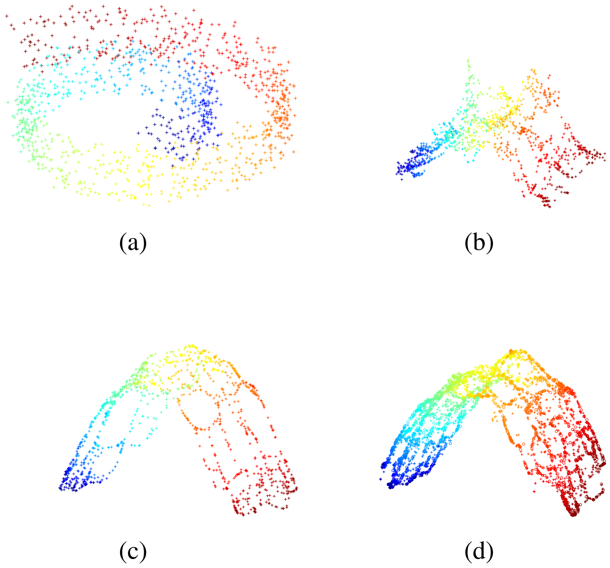


Fig. 2. Comparison of the conventional Isomap with our kernel Isomap for the case of *noisy Swiss Roll* data: (a) noisy Swiss Roll data; (b) embedded points using the conventional Isomap; (c) embedded points using our kernel Isomap; (d) projection of test data points using the kernel Isomap. The modification by the constant-adding in the kernel Isomap improves the embedding with preserving local isometry (see (c)) as well as allowing to projecting test data points onto a feature space (see (d)).

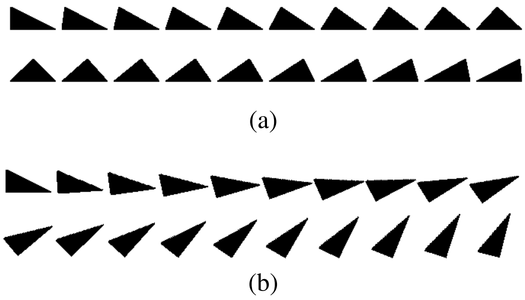


Fig. 3. The Composition of Triangle Data Set: (a) 20 different shapes; (b) 20 rotations of an triangle image

78^o for each triangle). See Fig. 3. The dimension of triangle image is 68 x 68. The low-dimensional space is shown in Fig. 4. The result of kernel Isomap was a slightly rotated rectangular of this one, so we applied Independent Component Analysis (ICA) to find proper coordinates. Then, we found same triangle images with query image disregarding the rotation. The results of retrieval are shown in Fig. 5 and Fig. 6. These results show that kernel Isomap allows a machine to recognize an object based on selected features and this could be applied into robot vision (for example, face recognition disregarding lighting effects). That is, the characteristics of an object can be understood from the viewpoint of selected features.

We applied this projection property into real world data,

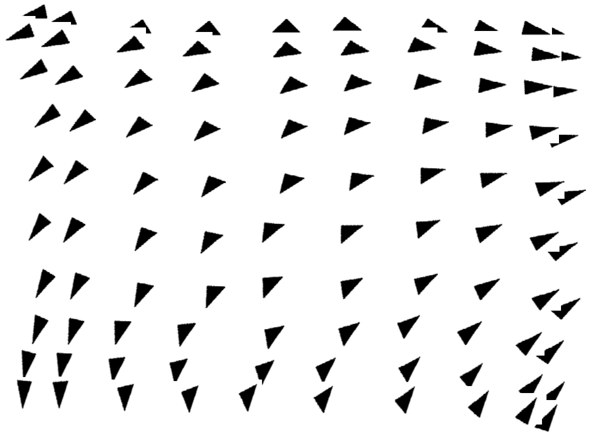


Fig. 4. Triangle images on the 2-dimensional space. Vertical axis means the shape of triangles and horizontal axis means the rotation of triangles.

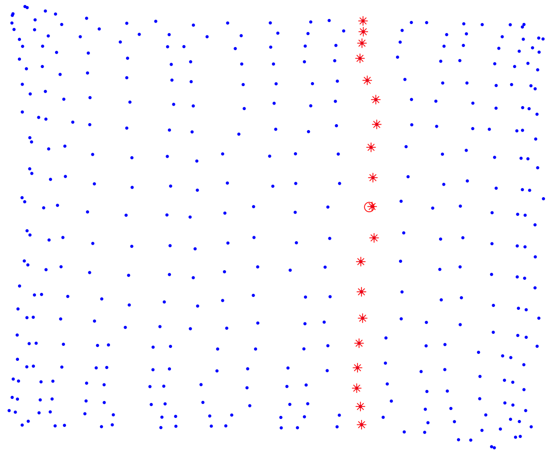


Fig. 5. The result of retrieval. The circled point is a query and the asterisk points are the results

USPS data set. We used a portion of the USPS, which contains digit '7' and '9'. Fig. 7 is the digit images projected on the 2-dimensional space. We found same digits with query digit disregarding the height-width ratio. The result of retrieval is shown in Fig. 8. This result shows that kernel Isomap could be applied into retrieval system which searches similar data points regarding only selected features.

B. *Topological Stability*

In previous section, the performance and the generalization property of kernel Isomap were confirmed. However, if the neighborhood size increases, the manifold is distorted as in Fig. 9 (a). This problem might come from noisy data or sparse data. Fig. 9 shows that our proposed algorithm, robust kernel Isomap, outperforms conventional Isomap. Isomap and kernel Isomap have some problems with large number (> 5) of neighborhood whereas our proposed Isomap is robust. If the

(a) (b)

Fig. 6. Query point and the results in triangle data (a) Query image; (b) The Results which have the same shape with query image

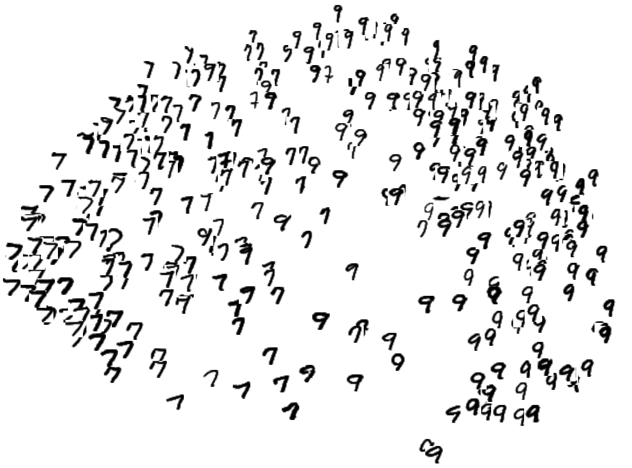


Fig. 7. Digit images on the 2-dimensional space. Vertical axis means the height-width ratio and horizontal axis means how the upper line of digit is rolled.

data set is sufficiently dense, then robust kernel Isomap can find the low-dimensional manifold even in high-level noise data set.

V. Conclusion

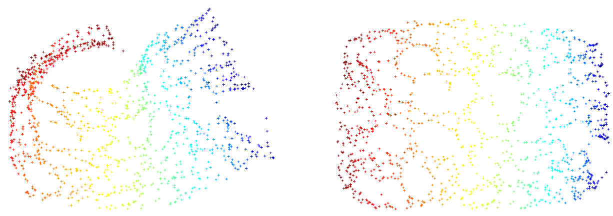
We have presented the kernel Isomap algorithm where the approximate geodesic distance matrix could be interpreted as a kernel matrix and an adding-constant method was exploited so that the geodesic distance-based kernel became Mercer kernel. Main advantages of the kernel Isomap could be summarized as follows: (a) generalization property (i.e., test data points can be projected onto the feature space using the kernel trick as in kernel PCA); (b) robustness against low-level noisy data. The generalization property will derive the kernel Isomap to be useful for pattern recognition problems. We tested the generalization property in retrieval system with real world data.

Moreover, we proposed a technique to improve the robustness on high-level noisy data where short-circuit edges could be a problem to find the proper manifold. As shown above, with the network flow, we could find the manifold from the noisy data set. The concept of network flow to detect noise data, can be applied into other manifold learning methods or clustering algorithms.

The final algorithm, robust kernel Isomap, has generalization property and robustness for both high and low level noisy data set. These properties are expected to be very useful in robot vision or document processing.

(a) (b)

Fig. 8. Query point and the results in USPS handwritten digits (a) Query image; (b) The Results which are the same degree of rolling with query image



(a) (b)

Fig. 9. Embedded manifolds according to different methods for the case of *noisy Swiss Roll* data with the number of neighborhood 6: (a) Conventional Isomap; (b) robust Kernel Isomap

Acknowledgment

This work was supported by ITEP Brain Neuroinformatics program, KISTEP International Cooperative Research Program, National Core Research Center for Systems Bio-Dynamics, and KOSEF 2000-2-20500-009-5.

References

[1] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[2] T. Cox and M. Cox, *Multidimensional Scaling*, 2nd ed. Chapman & Hall, 2001.

[3] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proc. Int'l Conf. Machine Learning*, Banff, Canada, 2004, pp. 369–376.

[4] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[5] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford, "The Isomap algorithm and topological stability," *Science*, vol. 295, January 2002.

[6] C. K. I. Williams, "On a connection between kernel PCA and metric multidimensional scaling," *Machine Learning*, vol. 46, pp. 11–19, 2002.

[7] F. Cailliez, "The analytical solution of the additive constant problem," *Psychometrika*, vol. 48, no. 2, pp. 305–308, 1983.

[8] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1540–1551, 2003.

[9] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems*, vol. 15. MIT Press, 2003, pp. 705–712.

[10] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmap, and spectral clustering," in *Advances in Neural Information Processing Systems*, vol. 16. Cambridge, MA: MIT Press, 2004.

[11] S. T. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[12] B. Bollobás, *Modern Graph Theory*. Springer, 1998.