Robust kernel Isomap

Article *in* Pattern Recognition · March 2007

DOI: 10.1016/j.patcog.2006.04.025 · Source: DBLP

2 authors:

Heeyoul "Henry Choi
Handong Global University
42 PUBLICATIONS **381** CITATIONS

SEE PROFILE

Seungjin Choi
Pohang University of Science and Technology
298 PUBLICATIONS **4,836** CITATIONS

SEE PROFILE

Robust kernel Isomap

Heeyoul Choi *, Seungjin Choi **

Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea

Abstract

Isomap is one of widely-used low-dimensional embedding methods, where geodesic distances on a weighted graph are incorporated with the classical scaling (metric multidimensional scaling). In this paper we pay our attention to two critical issues that were not considered in Isomap, such as: (1) generalization property (projection property); (2) topological stability. Then we present a *robust kernel Isomap* method, armed with such two properties. We present a method which relates the Isomap to Mercer kernel machines, so that the generalization property naturally emerges, through kernel principal component analysis. For topological stability, we investigate the network flow in a graph, providing a method for eliminating critical outliers. The useful behavior of the robust kernel Isomap is confirmed through numerical experiments with several data sets.

Key words: Isomap, Kernel PCA, Manifold learning, Multidimensional scaling (MDS), Nonlinear dimensionality reduction.

1 Introduction

Manifold learning involves inducing a smooth nonlinear low-dimensional manifold from a set of data points drawn from the manifold. Recently, various methods (for example see [1, 2, 3]) have been developed in machine learning

* He is currently in the Department of Computer Science, Texas A&M University.
Email: heeyoul@gmail.com (H. Choi)
**Corresponding author. Tel.: +82-54-279-2259; Fax: +82-54-279-2299
Email: seungjin@postech.ac.kr (S. Choi)
URL: http://www.postech.ac.kr/∼seungjin (S. Choi)

community and their wide applications started to draw an attention in pattern recognition and signal processing. Isomap is one of representative isometric mapping methods, which extends metric multidimensional scaling (MDS), considering Dijkstra's geodesic distances (shortest paths) on a weighted graph, instead of Euclidean distances [1].

Classical scaling (one of metric MDS) is closely related to principal component analysis (PCA) [4]. The projection of the centered data onto the eigenvectors of the data sample covariance matrix, returns the classical scaling solution. Classical scaling, where Euclidean distances are employed as dissimilarities, can be explained in the context of PCA, 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. A relationship between kernel PCA and metric MDS was investigated in [5].

The geodesic distance matrix used in Isomap, can be interpreted as a kernel matrix [6]. However, the kernel matrix based on the doubly centered geodesic distance matrix, is not always positive semidefinite. We mainly exploit a constant-shifting method such that the geodesic distance-based kernel matrix is guaranteed to be positive semidefinite. The method which incorporates a constant-shifting method into Isomap, is referred to as *kernel Isomap*, since the geodesic distance matrix with a constant-shifting technique is guaranteed to be a *Mercer kernel* matrix. This Mercer kernel Isomap algorithm has a generalization property, enabling us to project test data points onto an associated low-dimensional manifold, using a kernel trick as in kernel PCA [7], whereas, in general, most of embedding methods (including Isomap, LLE, and Laplacian eigenmap) do not have such a property. We investigate several constant-shifting methods and compare them in the context of the kernel Isomap.

Some critical outliers on noisy data manifold result in short-circuit edges, connecting two submanifolds directly, which often cause a topological instability [8]. For example, a single node (corresponding to a data point) which lies between closely apart surfaces, can connect these two surfaces if the size of neighborhood is relatively large in constructing a neighborhood graph (see Fig. 1). In such a case, the manifold structure is totally collapsed by an outlier. In order to overcome this situation, we introduce the total flows of nodes using the network flows in a graph and present a method of eliminating critical outliers. We assume that data points are almost uniformly distributed in the noise-free data manifold, such that the total flows of those nodes are proper values. Outlier nodes causing short-circuit edges, increase total flows dramatically. Thus we eliminate a few nodes which have extraordinary total flows.

We show that this simple technique is quite helpful for preserving topological stability in kernel Isomap. The method which incorporates this pre-processing into the kernel Isomap, is referred to as *robust kernel Isomap*.

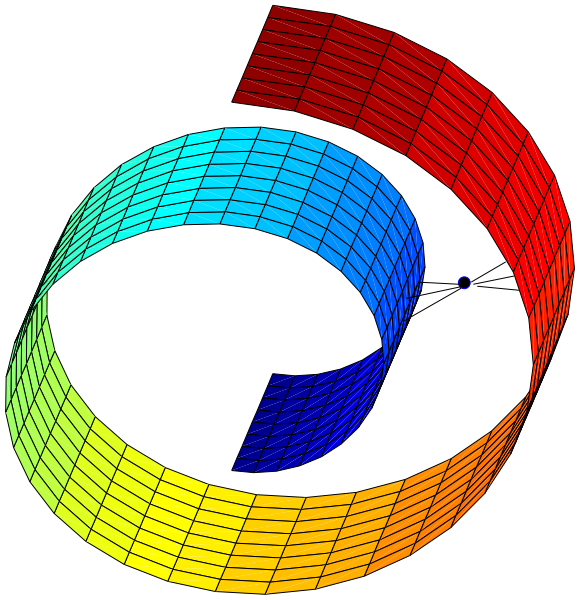


Fig. 1. A critical outlier connecting two surfaces on a manifold, could devastate the manifold structure. In such a case, Isomap (or other manifold learning methods) fails to find an appropriate low-dimensional embedding.

The rest of this paper is organized as follows. Next section explains one of main contributions in our paper, i.e., kernel Isomap algorithm, where the construction of a Mercer kernel matrix in the context of Isomap is illustrated, considering several different constant-shifting methods. A connection between the kernel Isomap and kernel PCA is also discussed. Sec. 3 describes another contribution of this paper, illustrating a method of eliminating critical outliers through monitoring total flows in a graph. This method reduces the possibility of topological instability in kernel Isomap. Numerical experiments are shown in Sec. 4, in order to confirm the useful behavior of our robust kernel Isomap algorithm. Finally conclusions are drawn in Sec. 5.

2 Kernel Isomap

2.1 Isomap as a kernel machine

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 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.

Following the connection between the classical scaling and PCA, metric MDS can be interpreted as kernel PCA [5]. In a similar manner, the geodesic distance matrix in Isomap, can be viewed as a kernel matrix [6]. The doubly centered geodesic distance matrix $\boldsymbol{K}$ in Isomap is of the form

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

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

However, the kernel matrix $\boldsymbol{K}$ in Eq. (1), is not always positive semidefinite. The main idea for kernel Isomap is to make this $\boldsymbol{K}$ as a Mercer kernel matrix (that is positive semidefinite) using a constant-shifting method, in order to relate it to kernel PCA such that the generalization property naturally emerges.

2.2 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, which leads to $\widetilde{\boldsymbol{K}}$. In fact, the additive constant problem was extensively studied in the context of MDS [4] and recently in embedding [9]. The matrix $\widetilde{\boldsymbol{K}}$ induced by a constant-shifting 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 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. Compute geodesic distances, D_{ij}, that are associated with the sum of edge weights along 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 matrix $\boldsymbol{K}(\boldsymbol{D}^2) = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{H}$, given in 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 $\widetilde{\boldsymbol{K}} = \widetilde{\boldsymbol{K}}(\widetilde{\boldsymbol{D}}^2)$ that is of the form

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

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

Step 5. Compute 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 the column vectors of $\boldsymbol{Y} = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{V}^\top$.

A main difference between the 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 $c \geq c^*$, which makes the matrix $\boldsymbol{K}$ to be positive semidefinite. The δ_{ij} is the Kronecker delta. Substituting $\widetilde{D}_{ij}$ for D_{ij} in Eq. (4) gives Eq. (3). Cailliez showed that the smallest number c^* which guarantees $\widetilde{D}_{ij}$ in Eq. (4) to be positive semidefinite, is given by the largest eigenvalue of the matrix in Eq. (2) [10].

Kernel matrix consists of inner products between $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$ where $\phi(\boldsymbol{x}_i)$ represents a nonlinear transformation of $\boldsymbol{x}_i$ into a feature space. The kernel trick involves using only inner product values, $\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)$, instead of working with nonlinearly-transformed values themselves $\{\phi(\boldsymbol{x}_i)\}$. The constant-shifting method guarantees that $\widetilde{\boldsymbol{K}}$ in Eq. (3) is positive semidefinite. Then, it follows from Mercer's theorem that we have

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

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, $\boldsymbol{C} = \frac{1}{N} (\boldsymbol{\Phi H}) (\boldsymbol{\Phi H})^\top$, where $\boldsymbol{\Phi} = [\phi(\boldsymbol{x}_1) \ldots \phi(\boldsymbol{x}_N)]$. Through a kernel trick as in kernel PCA, we explain how test data points are embedded in a low-dimensional space in the kernel Isomap.

2.3 Generalization property

The generalization property (or projection property) of the kernel Isomap, involves the embedding of test data points in an associated low-dimensional space. In other words, the generalization property means the ability to determine a point $\boldsymbol{y}_l$ embedded in the low-dimensional space, given a test data point $\boldsymbol{t}_l$. The generalization property naturally emerges from the fact that $\widetilde{\boldsymbol{K}}$ (geodesic kernel with the constant-shifting employed in our kernel Isomap) is a Mercer kernel. We first illustrate a connection between kernel Isomap and kernel PCA, which is a a direct consequence of the relation between classical scaling and PCA. This illustration immediately clarifies the projection property of kernel Isomap because it is the same as kernel PCA. One distinction to usual cases where a kernel function is defined everywhere, is that our method requires the calculation of geodesic kernel $k(\boldsymbol{t}_l, \boldsymbol{x}_j)$ for $\{\boldsymbol{x}_j\}$ belonging to a set of training data points. In other words, we use a data-driven kernel, instead of a everywhere-defined kernel function that can not be determined in our case.

The Mercer kernel matrix (geodesic kernel matrix) $\widetilde{\boldsymbol{K}}$ in the kernel Isomap can be written as $\widetilde{\boldsymbol{K}} = (\boldsymbol{\Phi H})^\top (\boldsymbol{\Phi H})$. The eigendecomposition of $\widetilde{\boldsymbol{K}}$ is of the form

$$\widetilde{\boldsymbol{K}} = \boldsymbol{V \Lambda V}^\top. \tag{6}$$

Premultiplying both sides of Eq. (6) by $\boldsymbol{\Phi H}$ leads to

$$\boldsymbol{\Phi H} \widetilde{\boldsymbol{K}} \boldsymbol{V} = \boldsymbol{\Phi H V \Lambda}. \tag{7}$$

Then we have

$$\boldsymbol{\Phi H H \Phi}^\top \boldsymbol{\Phi H V} = \boldsymbol{\Phi H V \Lambda}. \tag{8}$$

Note that the sample covariance matrix in the feature space, $\boldsymbol{C} = \frac{1}{N} (\boldsymbol{\Phi H}) (\boldsymbol{\Phi H})^\top$, is given by $N\boldsymbol{C} = \boldsymbol{\Phi H H \Phi}^\top$. Thus, Eq. (8) can be written as

$$N\boldsymbol{C}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Lambda}, \tag{9}$$

where $\boldsymbol{U} = \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{V}$ is the eigenvector matrix of $\boldsymbol{C}$.

Let $\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_n] \in \mathbb{R}^{m\times n}$. Then one can easily see that $\boldsymbol{u}_i^\top \boldsymbol{u}_i = \lambda_i$ for $i = 1, \ldots, n$, where λ_i is the i-th diagonal element of $\boldsymbol{\Lambda}$. Let $\widetilde{\boldsymbol{u}}_i = \lambda_i^{-\frac{1}{2}} \boldsymbol{u}_i$. Centering $\boldsymbol{\Phi}$ and projecting onto the unit vector $\widetilde{\boldsymbol{u}}_i$ leads to the eigenvector $\boldsymbol{v}_i$ scaled by $\lambda_i^{\frac{1}{2}}$, i.e.,

$$(\boldsymbol{\Phi}\boldsymbol{H})^\top \widetilde{\boldsymbol{u}}_i = \lambda_i^{\frac{1}{2}} \boldsymbol{v}_i. \tag{10}$$

As in kernel PCA, we can carry out the projection in Eq. (10) without computing $\widetilde{\boldsymbol{u}}_i$. That is, the embedding a test data point $\boldsymbol{t}_l$ in the low-dimensional space is done by

$$[\boldsymbol{y}_l]_i = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{N} [\boldsymbol{v}_i]_j k(\boldsymbol{t}_l, \boldsymbol{x}_j), \tag{11}$$

where $[\cdot]_i$ represents the ith element of a vector.

The geodesic kernel for the test data point $\boldsymbol{t}_l$, $k(\boldsymbol{t}_l, \boldsymbol{x}_j) = \phi(\boldsymbol{t}_l)^\top \phi(\boldsymbol{x}_j)$, in Eq. (11), is calculated in the same way as $\widetilde{\boldsymbol{K}}$ in Eq. (3), where geodesic distances, D_{lj}, between test data points $\boldsymbol{t}_l$ (one or several test data points) and all training data points $\boldsymbol{x}_j$, $j = 1, \ldots, N$, are required to be computed.

As in Eq. (4), D_{lj} is also modified by

$$\widetilde{D}_{lj} = D_{lj} + c. \tag{12}$$

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

$$\begin{aligned} \widetilde{D}_{lj}^2 &= [\phi(\boldsymbol{t}_l) - \phi(\boldsymbol{x}_j)]^\top [\phi(\boldsymbol{t}_l) - \phi(\boldsymbol{x}_j)] \\ &= \phi^\top(\boldsymbol{t}_l)\phi(\boldsymbol{t}_l) + \phi^\top(\boldsymbol{x}_j)\phi(\boldsymbol{x}_j) - 2\phi^\top(\boldsymbol{t}_l)\phi(\boldsymbol{x}_j). \end{aligned} \tag{13}$$

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

$$\frac{1}{N}\sum_{j=1}^{N} \widetilde{D}_{lj}^2 = \phi^\top(\boldsymbol{t}_l)\phi(\boldsymbol{t}_l) + \frac{1}{N}\sum_{j=1}^{N} \phi^\top(\boldsymbol{x}_j)\phi(\boldsymbol{x}_j). \tag{14}$$

Then, it follows from Eqs. (13) and (14) that the kernel for the test data point $\boldsymbol{t}_l$, is computed as

$$
\begin{aligned}
k(\boldsymbol{t}_l, \boldsymbol{x}_j) &= \phi^\top(\boldsymbol{t}_l)\phi(\boldsymbol{x}_j) \\
&= -\frac{1}{2}\left(\widetilde{D}_{lj}^2 - \phi^\top(\boldsymbol{t}_l)\phi(\boldsymbol{t}_l) - \phi^\top(\boldsymbol{x}_j)\phi(\boldsymbol{x}_j)\right) \\
&= -\frac{1}{2}\left(\widetilde{D}_{lj}^2 - \frac{1}{N}\sum_{i=1}^{N}\widetilde{D}_{li}^2 + \frac{1}{N}\sum_{i=1}^{N}\widetilde{K}_{ii} - \widetilde{K}_{jj}\right). \qquad (15)
\end{aligned}
$$

The L-Isomap [11] involving landmark points and an out-of-sample extension of Isomap (and other manifold learning methods) [12], also shares a similar spirit with our projection method in Eqs. (11) and (15). Using our notation, their kernel for a test data point is described by

$$
k(\boldsymbol{t}_l, \boldsymbol{x}_j) = -\frac{1}{2}\left(D_{lj}^2 - \frac{1}{N}\sum_{i=1}^{N}D_{li}^2\right). \qquad (16)
$$

The kernel in Eq. (16) can be viewed as a special instance of our kernel in Eq. (15), where last two terms involving the geodesic kernel matrix for training data points are ignored. In addition, the kernel in Eq. (16) 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.

2.4 Other constant-shifting methods

Several other constant-shifting methods can also be used to make the geodesic kernel matrix to be positive semidefinite. A *negative constant* can be added to the original dissimilarities so that

$$
\widetilde{D}_{ij} = |D_{ij} + c(1 - \delta_{ij})|, \qquad (17)
$$

has a Euclidean representation for all $c < c'$ [10] where c' is the smallest eigenvalue of the matrix in Eq. (2). The performance of a negative constant adding algorithm is very similar to one of the constant adding algorithm in Sec. 2.2.

On the other hand, instead of distances, squared distances can also be shifted, i.e.,

$$
\widetilde{D}_{ij} = (D_{ij}^2 + k(1 - \delta_{ij}))^{1/2}. \qquad (18)
$$

The smallest number k^* such that $\widetilde{D}_{ij}$ has an Euclidean representation for all $k \geq k^*$, is $k^* = -2\lambda_n$ where λ_n is the smallest eigenvalue of $\boldsymbol{K}(\boldsymbol{D}^2)$.

The matrix $\boldsymbol{K}(\boldsymbol{D}^2)$ can be directly modified to be positive semidefinite,

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} - \lambda_n \boldsymbol{I}, \tag{19}$$

where λ_n is the smallest eigenvalue of $\boldsymbol{K}(\boldsymbol{D}^2)$.

One can show that Eq. (18) is asymptotically equivalent to Eq. (19). Substituting Eq. (18) into Eq. (1), leads to

$$\begin{aligned}\widetilde{\boldsymbol{K}}(\boldsymbol{D}^2) &= -\frac{1}{2}\boldsymbol{H}(\boldsymbol{D}^2 - k(\boldsymbol{I} - \boldsymbol{e}_N\boldsymbol{e}_N^\top))\boldsymbol{H} \\ &= -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{H} + \frac{k}{2}\boldsymbol{H}(\boldsymbol{I} - \boldsymbol{e}_N\boldsymbol{e}_N^\top)\boldsymbol{H} \\ &= \boldsymbol{K}(\boldsymbol{D}^2) + \frac{k}{2}\boldsymbol{H}. \end{aligned} \tag{20}$$

As the number of data points, N, increases, $\boldsymbol{H}$ approaches $\boldsymbol{I}$. Thus Eq. (20) approaches

$$\widetilde{\boldsymbol{K}}(\boldsymbol{D}^2) \simeq \boldsymbol{K}(\boldsymbol{D}^2) + \frac{k}{2}\boldsymbol{I}. \tag{21}$$

Choosing $k = -2\lambda_n$, Eq. (21) becomes equivalent to Eq. (19).

Constant-shifting methods show slightly different behavior. Fig. 2 (a) shows 50 smallest eigenvalues and 50 largest eigenvalues of the kernel matrix $\boldsymbol{K}$ of the Isomap, for the case of noisy Swiss Roll data. Some of smallest eigenvalues are negative in this case. A direct modification of the geodesic kernel matrix $\boldsymbol{K}$ using the constant-shifting Eq. (19), results in a linear shift of eigenvalues such that the smallest one is not negative (see Fig. 2 (b)). On the other hand, the Cailliez's constant-shifting method that was adopted in the kernel Isomap, leads to a nonlinear shift of eigenvalues such that most information is preserved in first few eigenvalues and the rest of eigenvalues are close to zero (but nonnegative) (see Fig. 2 (c)).

3 Topological stability

It was pointed out in [8] 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 linear embedding (LLE) [13]. A relatively large neighborhood size might result in short-circuit edges (for example, see Fig. 1) which destruct the manifold structure of data points. An easy

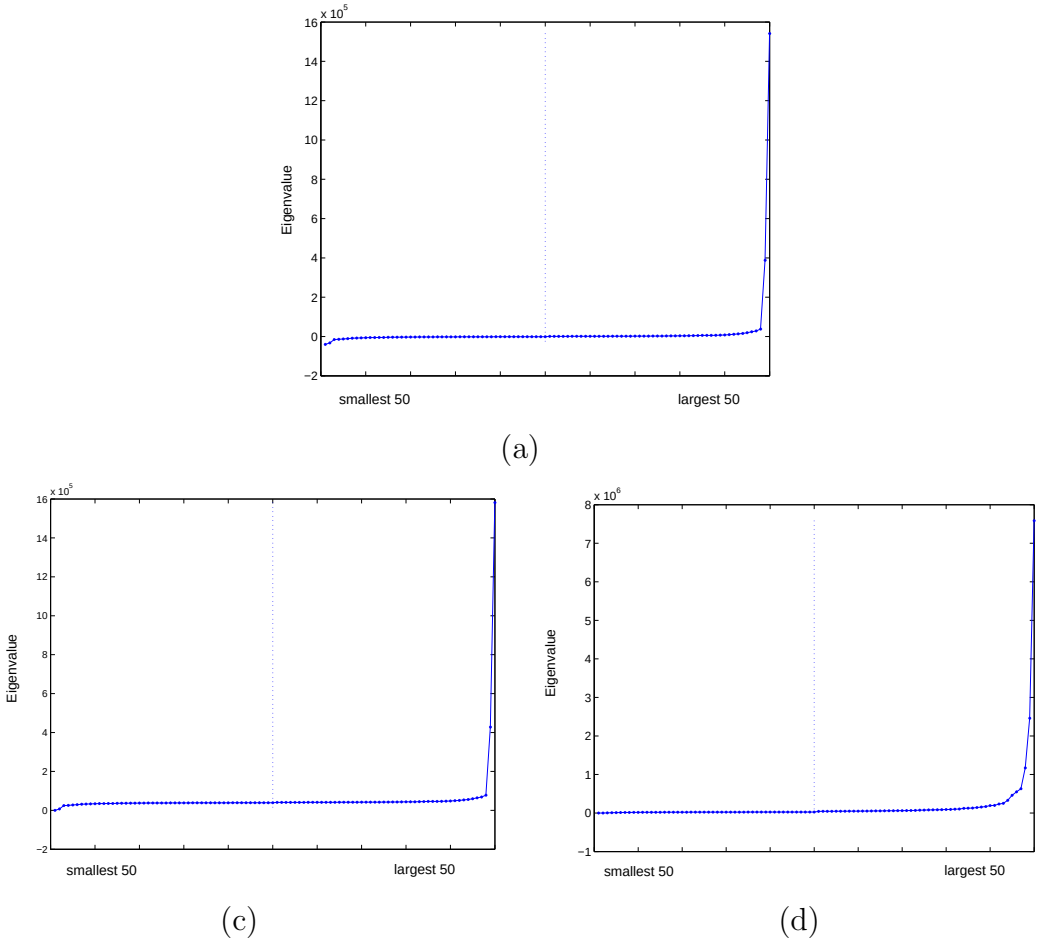


Fig. 2. For the case of noisy Swiss Roll data, smallest 50 eigenvalues and 50 largest eigenvalues of the kernel matrix in Isomap and kernel Isomap (where geodesic kernel matrix is modified to be positive semidefinite), are shown. In each graph, 50 smallest eigenvalues and 50 largest eigenvalues are shown in the left-half and the right-half, respectively: (a) Isomap; (b) kernel Isomap with a constant-shifting Eq. (19); (c) kernel Isomap with a constant-shifting in Sec. 2.

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 between all pairs of nodes 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.

Definition 1 ([14]) *Let $\mathcal{G}$ be a directed graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. A* network flow *f 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. That is,

$$\eta(\epsilon_k) = \sum_{i=1}^{N} \sum_{j=1}^{N} \varphi(k, i, j), \tag{22}$$

where

$$\varphi(k, i, j) = \begin{cases} 1 & \text{if } \epsilon_k \in \text{path}(i, j), \\ 0 & \text{otherwise,} \end{cases} \tag{23}$$

where path(i, j) denotes the geodesic path from the ith node to the jth node.

Then we define the *total flow*, $f(v_k)$, of a node v_k as the sum of the network flows of edges connecting to the node v_k, i.e.,

$$f(v_k) = \sum_{v_i \in \mathcal{N}_k} \eta(\epsilon(i, k)), \tag{24}$$

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.

We assume that a data manifold consists of uniformly scattered data points. In such a case, critical outliers which are located off the manifold and connect two sub-manifolds directly (for example, see Fig. 1), are expected to have extremely high total flow values. It is not clear how 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).

For example, see Fig. 4 where total flows of nodes for the case of noisy Swiss Roll Data, are shown, for two different neighborhood size, $k = 5$ and $k = 6$. For the case of $k = 6$, Isomap is not topological stable (see Fig. 5 (a)). In this case, a few nodes have extremely high total flow values (see Fig. 4 (b)). On the other hand, for the case of $k = 5$, total flow values are well distributed (see Fig. 4), so Isomap is topologically stable.

Calculating total flow values of nodes does not require any extra computational complexity, because it uses geodesic paths already computed for the kernel Isomap. We simply eliminate a few nodes which have extremely high total flow values, before constructing the geodesic kernel matrix. Once outliers are removed, then we calculate geodesic paths again to construct the geodesic kernel matrix and apply the kernel Isomap to find a low-dimensional manifold. This is referred to as *robust kernel Isomap* which is summarized in Table 1.

Our proposed scheme is more appropriate for handling with critical outliers (which has extremely high total flow values) rather than evenly-distributed noise. However, even in the case of Gaussian noise with dense data set, if the neighborhood size is small enough, we can see a few short-circuit edges. In general, when a low-dimensional manifold is folded in a high-dimensional space, short-circuit edges are most-likely to appear. Controlling the neighborhood size properly, leads us to detect the outliers and unfold the manifold successfully, which is confirmed in numerical experiments.

Table 1
Algorithm outline: 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.

4 Numerical experiments

We carried out numerical experiments with three different data sets: (a) noisy Swiss Roll data; (2) head-related impulse response (HRIR) data; (3) spoken English letters data, in order to show the useful behavior of our kernel Isomap or robust kernel Isomap algorithm.

4.1 Experiment 1: Noisy Swiss roll data

Noisy Swiss Roll data was generated by adding isotropic Gaussian noise with zero mean and variance=0.25 to the original Swiss Roll data that was used in Isomap (see Fig. 3 (a)). In the training phase, 1200 data points were used to construct a neighborhood graph with the neighbor size, $k = 4$. As in Isomap, geodesic distances were computed by calculating shortest paths using the Dijkstra's algorithm.

Exemplary embedding results (onto 3-dimensional feature space) for Isomap and kernel Isomap, are shown in Fig. 3 (b) and (c), where kernel Isomap provided smooth embedded manifold, in contrast to the Isomap. The geodesic kernel matrix modified by a constant-shifting method, led the kernel Isomap to find a smooth embedded manifold.

The generalization property of the kernel Isomap algorithm is shown in Fig. 3 (d), where 3000 test data points are embedded appropriately with preserving local isometry.

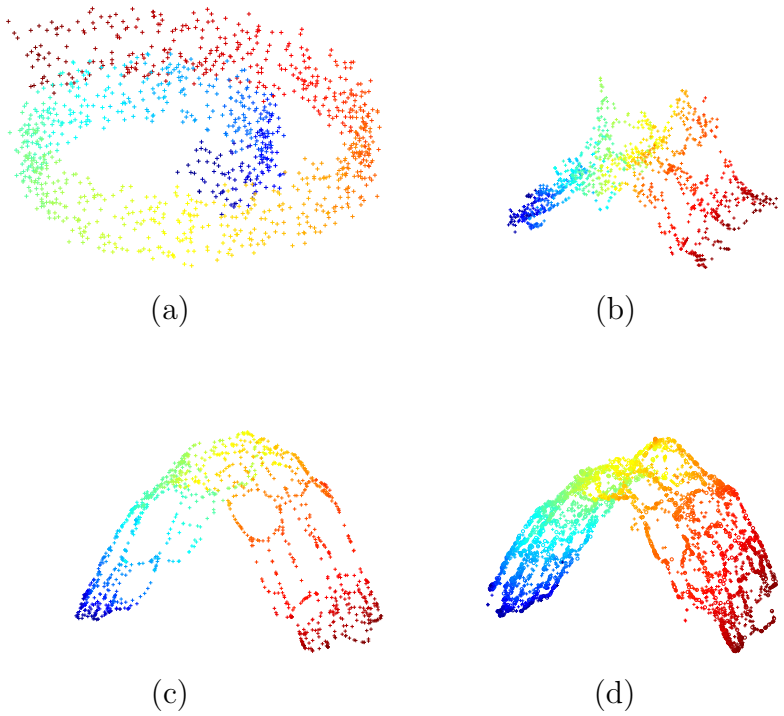
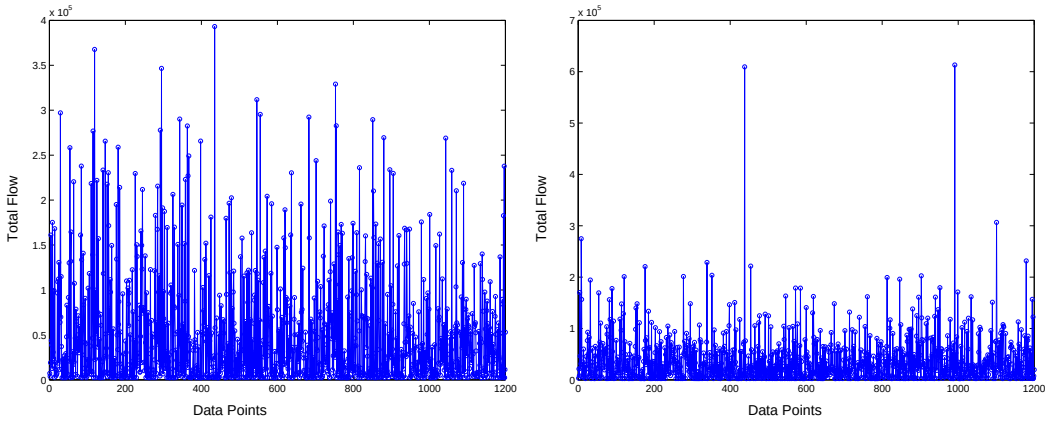


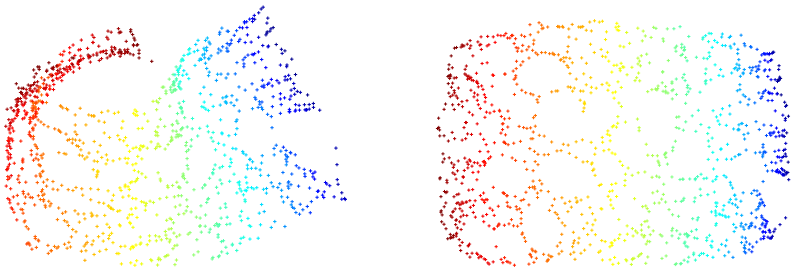
Fig. 3. Comparison of the Isomap with the kernel Isomap for the case of *noisy Swiss Roll* data: (a) noisy Swiss Roll data; (b) embedding result by the Isomap; (c) embedding result by the kernel Isomap; (d) projection of test data points using the kernel Isomap. The modification by the constant-shifting 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)).

If the neighborhood size increases in constructing a neighborhood graph, Isomap could be topological unstable that was pointed out in [8]. For the case of noisy Swiss Roll data, we set $k = 6$ in constructing a neighborhood graph for both Isomap and kernel Isomap. In such a case, 2-dimensional manifolds are shown in Fig. 5, where the 2-dimensional feature space computed by the Isomap is distorted, but the robust kernel Isomap identifies the correct 2-dimensional manifold as in the case for $k = 4$. This numerical experiment verifies the useful behavior of the robust kernel Isomap for noisy data where some critical outliers exist. If data points are sufficiently dense, then the robust kernel Isomap can find an appropriate low-dimensional manifold of data, even for the case of data with high-level noise.



(a) (b)

Fig. 4. 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.



(a) (b)

Fig. 5. Embedding results for the case of noisy Swiss Roll data, with the neighborhood size, $k = 6$: (a) Isomap; (b) robust Kernel Isomap. The performance of Isomap with $k = 6$, is severely degraded. Isomap even fails to find an appropriate low-dimensional manifold. In contrast, the robust kernel Isomap eliminates critical outliers as a preprocessing, hence, it is still able to find a correct low-dimensional manifold, even with $k = 6$.

4.2 Experiment 2: HRIR data

It was recently shown in [15] that the low-dimensional manifold of the head-related impulse responses (HRIRs) could encode the perceptual information related to the direction of sound source. The locally linear embedding (LLE) [13] was applied to find a nonlinear low-dimensional feature space of HRIRs [15]. In this experiment, we use the public-domain CIPIC HRIR data set [16] and apply the Isomap as well as the kernel Isomap, comparing the embedding results of these methods.

The detailed description for HRIR data sets can be found in [16]. We mainly

pay our attention to the HRIRs involving sound sources specified by different elevation angles (see Fig. 6). The database contains HRIRs sampled at 1250 points around the head for 45 subjects. Azimuth is sampled from $-80°$ to $80°$ and elevation from $-45°$ to $230.625°$. Each HRIR is a 200-dimensional vector corresponding to a duration of about 4.5ms. A HRIR of the right ear for the 18th subject is shown in Fig. 7.

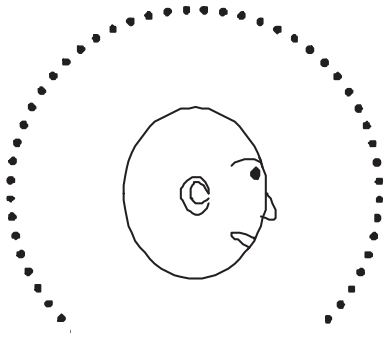


Fig. 6. HRIRs are measured for sound sources at different locations. Locations of sound sources vary according to different elevation angles in interaural-polar coordinates. Elevations are uniformly sampled in $360/64 = 5.625°$ steps from $-45°$ to $230.625°$.

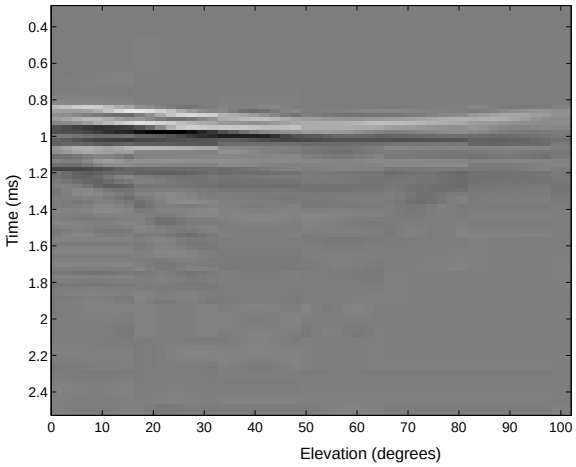
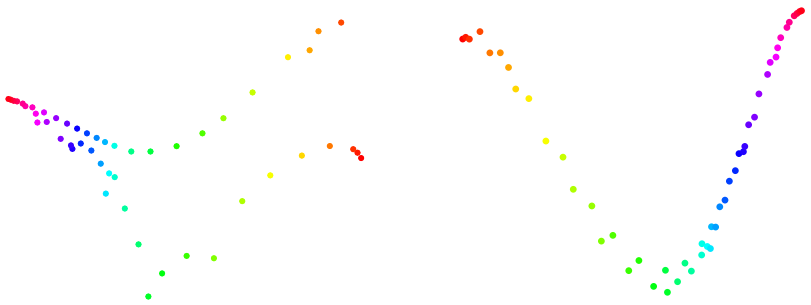


Fig. 7. A HRIR of the right ear for the case of zero azimuth.

Two-dimensional manifolds of HRIRs (with different elevation angles) are shown in Fig. 8 for Isomap and kernel Isomap, where the kernel Isomap finds a smooth low-dimensional manifold which encodes the perceptual information related to different elevation angles (location of sound), in contrast to the Isomap. The one-dimensional manifold computed by the kernel Isomap is shown in Fig. 9, where points projected onto the largest eigenvector of the geodesic kernel matrix $\widetilde{\boldsymbol{K}}$ used in the kernel Isomap, are plotted with respect to elevation angles.



(a) (b)

Fig. 8. Two-dimensional manifolds of HRIRs: (a) Isomap; (b) Kernel Isomap. The kernel Isomap finds a smooth low-dimensional manifold of HRIRs, in contrast to the Isomap.

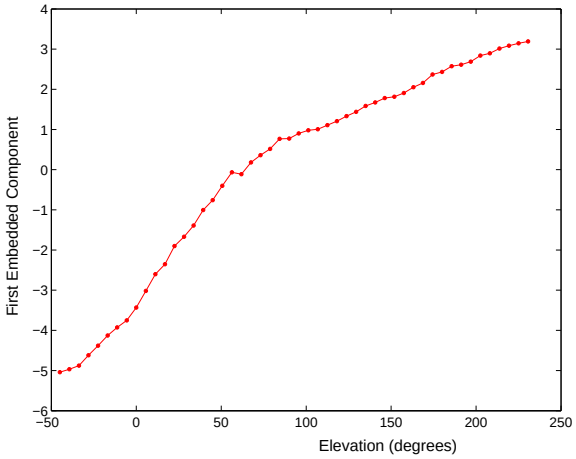


Fig. 9. The one-dimensional manifold computed by the kernel Isomap.

4.3 Experiment 3: Spoken letters data

In Experiment 3, we applied Isomap and kernel Isomap to 'Isolet Spoken Letters Data' which is available from UCI Repository [17], that was also used recently in [18, 19]. We used a portion of Isolet DB, which contains utterances of 30 subjects who spoke the name of each letter of English alphabet twice. Thus the number of data points are $1560 = 26 \times 2 \times 30$. Attributes (features) are 617, including spectral coefficients, contour features, sonorant features, pre-sonorant features, and post-sonorant features. These utterances are high-dimensional data, however, it is expected that distinctive phonetic dimensions are few. Two-dimensional manifolds of Isolet data found by Isomap and robust kernel Isomap, are shown in Fig. 10, where one can see that the robust kernel Isomap shows slightly better cluster structure, compared to the Isomap. Even though the robust kernel Isomap are not able to discriminate every English letters clearly in a two-dimensional manifold, it still shows better performance over the Isomap. The network flow evaluation for the robust kernel Isomap, is

shown in Fig. 11, where 7 data points whose total flow values are above the half of the largest values, were identified as outliers.

5 Conclusions

We have presented the kernel Isomap algorithm where we constructed a geodesic Mercer kernel matrix through a constant-shifting method. The kernel Isomap was explicitly related to the kernel PCA, providing the generalization property such that test data points were able to be embedded in the associated low-dimensional space by a geodesic kernel mapping. We have also presented a method for eliminating critical outliers that could cause short-circuit edges. The detection of such outliers was carried out by evaluating the network flow. These two contributions (Mercer kernelized Isomap and network flow-based preprocessing) led to the robust kernel Isomap. Numerical experiments with several data sets such as noisy Swiss roll data, HRIR, and spoken letters, verified the useful behavior of our proposed method.

Acknowledgments

This work was supported by ITEP Brain Neuroinformatics program, KOSEF Basic Research Program (grant R01-2006-000-11142-0), and Korea MIC under ITRC support program supervised by the IITA (IITA-2005-C1090-0501-0018). Heeyoul Choi was also supported by KOSEF Grant-D00115.

References

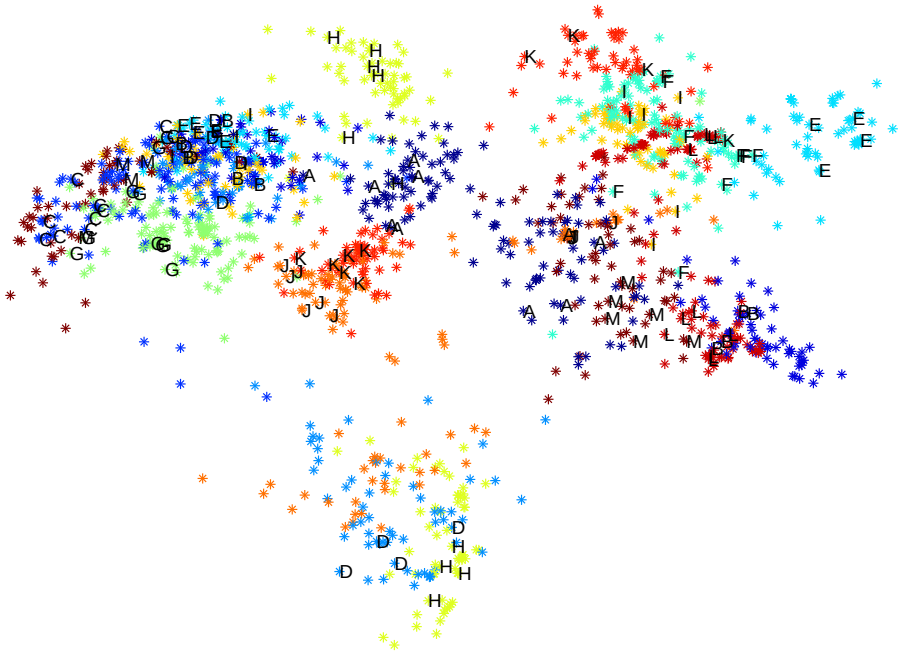
[1] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[2] L. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, June 2003.

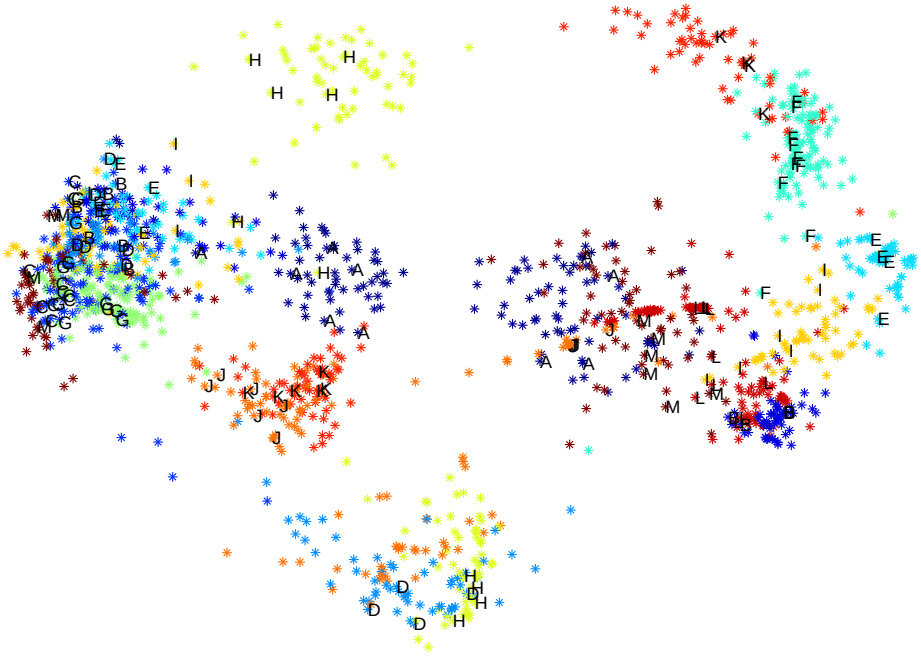
[3] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

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

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



(a) Isomap



(b) Robust kernel Isomap

Fig. 10. Two-dimensional manifolds of spoken letters data.

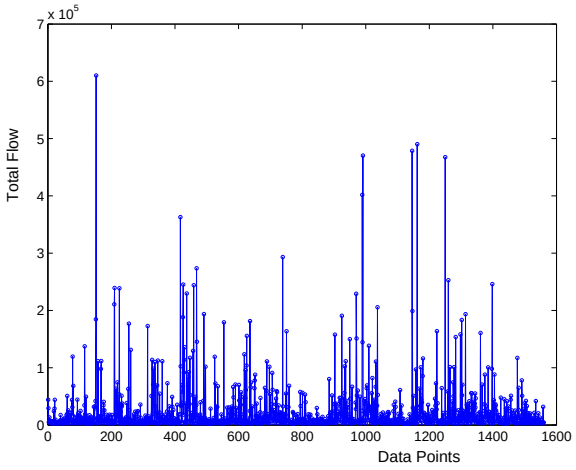


Fig. 11. Total flow value of each node for the case of spoken letters data.

[6] 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*, pages 369–376, Banff, Canada, 2004.

[7] B. Schölkopf, A. J. Smola, and K. -R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

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

[9] 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*, 25(12):1540–1551, 2003.

[10] F. Cailliez. The analytical solution of the additive constant problem. *Psychometrika*, 48(2):305–308, 1983.

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

[12] 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*, volume 16, Cambridge, MA, 2004. MIT Press.

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

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

[15] R. Duraiswami and V. C. Raykar. The manifolds of spatial hearing. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pages 285–288, 2005.

[16] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano. The CIPIC HRTF database. In *Proc. 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 99–102, 2001.

[17] S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

[18] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

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