Supervised Nonlinear Dimensionality Reduction for Visualization and Classification

Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou, *Member, IEEE*

***Abstract*—When performing visualization and classification, people often confront the problem of dimensionality reduction. Isomap is one of the most promising nonlinear dimensionality reduction techniques. However, when Isomap is applied to real-world data, it shows some limitations, such as being sensitive to the noise. In this paper, an improved version of Isomap, namely S-Isomap, is proposed. S-Isomap utilizes class information to guide the procedure of nonlinear dimensionality reduction. Such a kind of procedure is called supervised nonlinear dimensionality reduction. In S-Isomap, the neighborhood graph of the input data is constructed according to a certain kind of dissimilarity between data points, which is specially designed to integrate the class information. The dissimilarity has several good properties which help to discover the true neighborhood of the data, and thus makes S-Isomap a robust technique for both visualization and classification, especially for real-world problems. In the visualization experiments, S-Isomap is compared with Isomap, LLE and WeightedIso. The results show that S-Isomap performs the best. In the classification experiments, S-Isomap is used as a preprocess of classification and compared with Isomap, WeightedIso, as well as some other well-established classification methods including *K* nearest neighbor classifier, BP neural network, J4.8 decision tree and SVM. The results reveal that S-Isomap excels Isomap and WeightedIso in classification and is highly competitive with those well-known classification methods.**

***Index Terms*—Supervised learning, Dimensionality reduction, Manifold learning, Visualization, Classification**

I. Introduction

With the wide usage of information technology in almost all aspects of daily lives, huge amounts of data, such as climate patterns, gene distributions and commercial records, have been accumulated in various databases and data warehouses. Most of these data have many attributes, i.e. they are distributed in high dimensional spaces. People working with them regularly confront the problem of dimensionality reduction, which is a procedure of finding intrinsic low dimensional structures hidden in the high dimensional observations. This may be a crucial step for the tasks of data visualization or classification.

Dimensionality reduction can be performed by keeping only the most important dimensions, i.e. the ones that hold the most useful information for the task at hand, or by projecting the original data into a lower dimensional space that is most expressive for the task. For visualization, the goal of dimensionality reduction is to map a set of observations into a (two or three dimensional) space that preserves as much as possible the intrinsic structure. For classification, the goal is to map the input data into a feature space in which the members from different classes are clearly separated.

Many approaches have been proposed for dimensionality reduction, such as the well-known methods of principal component analysis (PCA) [5], independent component analysis (ICA) [2] and multidimensional scaling (MDS) [3]. In PCA, the main idea is to find the projection that restores the largest possible variance in the original data. ICA is similar to PCA except that the components are designed to be independent. Finally in MDS, efforts are taken to find the low dimensional embeddings that best preserve the pair wise distances between the original data points. All of these methods are easy to implement. At the same time, their optimizations are well understood and efficient. Because of these advantages, they have been widely used in visualization and classification. Unfortunately, they have a common inherent limitation: they are all linear methods while the distributions of most real-world data are nonlinear.

Recently, two novel methods have been proposed to tackle the nonlinear dimensionality reduction problem, namely Isomap [13] and LLE [9]. Both of these methods attempt to preserve as well as possible the local neighborhood of each object while trying to obtain highly nonlinear embeddings. So they are categorized as a new kind of dimensionality reduction techniques called Local Embeddings [16]. The central idea of Local Embeddings is using the locally linear fitting to solve the globally nonlinear problems, which is based on the assumption that data lying on a nonlinear manifold can be viewed as linear in local areas. Both Isomap and LLE have been used in visualization [9], [11], [13], [16] and classification [11], [16]. Encouraging results have been reported when the test data contain little noise and are well sampled. But as can been seen in the following sections of this paper, they are not so powerful when confronted with noisy data, which is often the case for

Manuscript received February 12, 2004. This work was supported by the National Outstanding Youth Foundation of China under the Grant No. 60325207.

Xin Geng is with the National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. (e-mail: gengx@lamda.nju.edu.cn).

De-Chuan Zhan is with the National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. (e-mail: zhandc@lamda.nju.edu.cn).

Zhi-Hua Zhou is with the National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. (phone: +86-25-8368-6268; fax: +86-25-8368-6268; e-mail: zhouzh@nju.edu.cn).

real-world problems. In this paper, a robust method based on the idea of Isomap, namely S-Isomap, is proposed to deal with such situation. Unlike the unsupervised learning scheme of Isomap, S-Isomap follows the supervised learning scheme, i.e. it uses the class labels of the input data to guide the manifold learning.

The rest of this paper is organized as follows. In section II, Isomap and the usage of it in visualization and classification are briefly introduced. In section III, S-Isomap is proposed, and the usage of it in visualization and classification is also discussed. In section IV, experiments are reported. Finally in section V, conclusions are drawn and several issues for the future work are indicated.

II. ISOMAP FOR VISUALIZATION AND CLASSIFICATION

For data lying on a nonlinear manifold, the "true distance" between two data points is the geodesic distance on the manifold, i.e. the distance along the surface of the manifold, rather than the straight-line Euclidean distance. The main purpose of Isomap is to find the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The approximation of geodesic distance is divided into two cases. In case of neighboring points, Euclidean distance in the input space provides a good approximation to geodesic distance. In case of faraway points, geodesic distance can be approximated by adding up a sequence of "short hops" between neighboring points. Isomap shares some advantages with PCA, LDA and MDS, such as computational efficiency and asymptotic convergence guarantees, but with more flexibility to learn a broad class of nonlinear manifolds. The Isomap algorithm takes as input the distances $d(\mathbf{x}_i, \mathbf{x}_j)$ between all pairs $\mathbf{x}_i$ and $\mathbf{x}_j$ from N data points in the high-dimensional input space $\mathbb{R}^q$. The algorithm outputs coordinate vectors $\mathbf{y}_i$ in a d-dimensional Euclidean space $\mathbb{R}^d$ that best represent the intrinsic geometry of the data. The detailed steps of Isomap are listed as follows:

Step 1. Construct neighborhood graph: Define the graph G over all data points by connecting points $\mathbf{x}_i$ and $\mathbf{x}_j$ if they are closer than a certain distance ε, or if $\mathbf{x}_i$ is one of the K nearest neighbors of $\mathbf{x}_j$. Set edge lengths equal to $d(\mathbf{x}_i, \mathbf{x}_j)$.

Step 2. Compute shortest paths: Initialize $d_G(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are linked by an edge; $d_G(\mathbf{x}_i, \mathbf{x}_j) = +\infty$ otherwise. Then for each value of k=1, 2, …, N in turn, replace all entries $d_G(\mathbf{x}_i, \mathbf{x}_j)$ by $\min\{ d_G(\mathbf{x}_i, \mathbf{x}_j), d_G(\mathbf{x}_i, \mathbf{x}_k) + d_G(\mathbf{x}_k, \mathbf{x}_j)\}$. The matrix of final values $\mathbf{D}_G = \{d_G(\mathbf{x}_i, \mathbf{x}_j)\}$ will contain the shortest path distances between all pairs of points in G (This procedure is known as Floyd's algorithm).

Step 3. Construct d-dimensional embedding: Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(\mathbf{D}_G)$ (The operator τ is defined by $\tau(\mathbf{D}) = -\mathbf{HSH}/2$, where $\mathbf{S}$ is the matrix of squared distances $\{\mathbf{S}_{ij} = \mathbf{D}_{ij}^2\}$, and $\mathbf{H}$ is the "centering matrix" $\{\mathbf{H}_{ij} = \delta_{ij} - 1/N\}$, δ_{ij} is the Kronecker delta function. [7]), and $\mathbf{v}_p^i$ be the i-th component of the p-th eigenvector. Then set the p-th component of the d-dimensional coordinate vector $\mathbf{y}_i$ equal to $\sqrt{\lambda_p}\mathbf{v}_p^i$ (This is actually a procedure of applying classical MDS to the matrix of graph distances $\mathbf{D}_G$).

Note that the only free parameter of Isomap, ε or K, appears in *Step 1*. In this paper, only the parameter K is used in Isomap.

Isomap can be easily applied to visualization. In this case, two or three dimensional embeddings of higher dimensional data are constructed using Isomap and then depicted in a single global coordinate system. Isomap's global coordinates provide a simple way to analyze and manipulate high-dimensional observations in terms of their intrinsic nonlinear degrees of freedom. Isomap has been successfully used to detect the true underlying factors of some high-dimensional data sets, such as synthetic face images, hand gesture images and handwritten digits [13]. However, as can be seen in the following sections, when the input data are more complex and noisy, such as a set of face images captured by a web camera, Isomap often fails to nicely visualize them. The reason is that the local neighborhood structure determined in the first step of Isomap is critically distorted by the noise.

As for the classification tasks, Isomap can be viewed as a preprocess. When the dimensionality of the input data is relatively high, most classification methods, such as K nearest neighbor classifier [4], [6], will suffer from the curse-of-dimensionality and get highly biased estimates. Fortunately, high dimensional data often represent phenomena that are intrinsically low dimensional. Thus the problem of high dimensional data classification can be solved by first mapping the original data into a lower dimensional space by Isomap (which can be viewed as a preprocess) and then applying K-NN classification to the images. Since the mapping function is not explicitly given by Isomap, it should be learned by some nonlinear interpolation techniques, such as Generalized Regression Networks [17]. Suppose that the data in $\mathbb{R}^q$ are mapped into $\mathbb{R}^d$ ($d < q$) by Isomap. The mapping function $f: \mathbb{R}^q \to \mathbb{R}^d$ can be learned by Generalized Regression Networks, using the corresponding data pairs in $\mathbb{R}^q$ and $\mathbb{R}^d$ as the training set. A given query $\mathbf{x}_0$ is first mapped into $\mathbb{R}^d$ to get its lower dimensional image $f(\mathbf{x}_0)$. Then its class label is given as the most frequent one occurring in the K neighbors of $f(\mathbf{x}_0)$ in $\mathbb{R}^d$. Unfortunately, this scheme seems not to work very well compared with those widely used classification methods (according to the experiments in this paper), such as BP network [10], decision tree [8] and SVM [14], [15]. There may be two reasons. First, the real-world data are often noisy, which can weaken the mapping procedure of Isomap. Second, the goal of the mapping in classification is different from that in visualization. In visualization, the goal is to faithfully preserve the intrinsic structure as well as possible, while in classification, the goal is to transform the original data into a feature space that can make classification easier, by stretching or constricting the original metric if necessary. Both reasons indicate that some

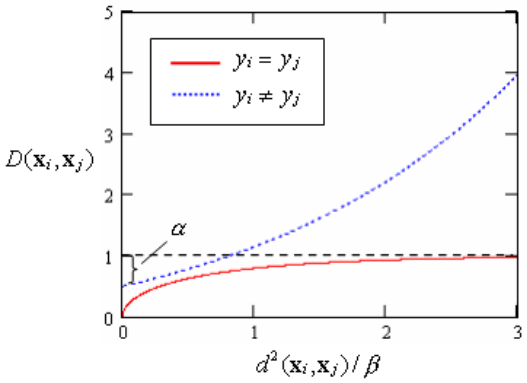


Fig. 1. Typical plot of $D(\mathbf{x}_i, \mathbf{x}_j)$ as a function of $d^2(\mathbf{x}_i, \mathbf{x}_j)/\beta$

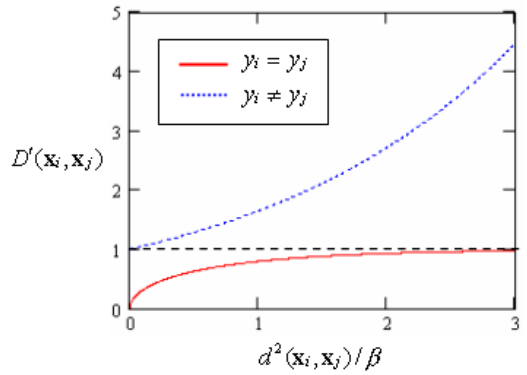


Fig. 2. Typical plot of $D'(\mathbf{x}_i, \mathbf{x}_j)$ as a function of $d^2(\mathbf{x}_i, \mathbf{x}_j)/\beta$

modification should be made on Isomap for the tasks of classification.

III. Supervised Isomap: S-Isomap

In some visualization tasks, data are from multiple classes and the class labels are known. In classification tasks, the class labels of all training data must be known. The information provided by these class labels may be used to guide the procedure of dimensionality reduction. This can be called supervised dimensionality reduction, in contrast to the unsupervised scheme of most dimensionality reduction methods. Some preliminary efforts have already been taken toward supervised dimensionality reduction, such as the WeightedIso method [16], which changes the first step of Isomap by rescaling the Euclidean distance between two data points with a constant factor λ ($\lambda < 1$) if their class labels are the same. The basic idea behind WeightedIso is to make the two points closer to each other in the feature space if they belong to the same class. It is believed that this can make classification in the feature space easier. However, WeightedIso is not suitable for visualization because it forcefully distorts the original structure of the input data no matter whether there is noise in the data or not and how much noise is in the data. Even in classification, the factor λ must be very carefully tuned to get a satisfying result (this has been well experienced in the following experiments). To make the algorithm more robust for both visualization and classification, a more sophisticated method is proposed in this section.

Suppose the given observations are $(\mathbf{x}_i, y_i)$, $i = 1...N$, where $\mathbf{x}_i \in \mathbb{R}^q$ and y_i is the class label of $\mathbf{x}_i$. Define the dissimilarity between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ as

$$D(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \sqrt{1 - e^{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\beta}}} & y_i = y_j \\ \sqrt{e^{\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{\beta}}} - \alpha & y_i \neq y_j \end{cases}, \quad (1)$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ denotes the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. A typical plot of $D(\mathbf{x}_i, \mathbf{x}_j)$ as a function of $d^2(\mathbf{x}_i, \mathbf{x}_j)/\beta$ is shown in Fig. 1. Since the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j)$ is in the exponent, the parameter β is used to prevent $D(\mathbf{x}_i, \mathbf{x}_j)$ to increase too fast when $d(\mathbf{x}_i, \mathbf{x}_j)$ is relatively large. Thus the value of β should depend on the "density" of the data set. Usually, β is set to be the average Euclidean distance between all pairs of data points. The parameter α gives a certain chance to the points in different classes to be "more similar", i.e. to have a smaller value of dissimilarity, than those in the same class. For a better understanding of α, it may be helpful to look at Fig. 2, which is the typical plot of $D'(\mathbf{x}_i, \mathbf{x}_j)$ defined in (2).

$$D'(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \sqrt{1 - e^{\frac{-d^2(\mathbf{x}_i, \mathbf{x}_j)}{\beta}}} & y_i = y_j \\ \sqrt{e^{\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{\beta}}} & y_i \neq y_j \end{cases} \quad (2)$$

In Fig. 2, the dissimilarity between two points is equal to or larger than 1 if their class labels are different and is less than 1 if otherwise. Thus the inter-class dissimilarity is definitely larger than the intra-class dissimilarity, which is a very good property for classification. However, this can also make the algorithm apt to overfit the training set. Moreover, this can often make the neighborhood graph of the input data disconnected, which is a situation that Isomap cannot handle. So α is used to lose the restriction and give the intra-class dissimilarity a certain probability to exceed the inter-class dissimilarity. The value of α should be greater than 0 and less than the value that makes the two curves tangent (when the two curves are tangent to each other, α is about 0.65 and the value of $d^2(\mathbf{x}_i, \mathbf{x}_j)/\beta$ at which the two curves touch is about 0.38). It is worth mentioning that the function of α can also be achieved through subtracting a constant from the squared dissimilarity $D^2(\mathbf{x}_i, \mathbf{x}_j)$, it does not matter much.

As a point of comparison, the typical plot of the dissimilarity used in WeightedIso, namely $WD(\mathbf{x}_i, \mathbf{x}_j)$, as a function of $d(\mathbf{x}_i, \mathbf{x}_j)$ is shown in Fig. 3 ($\lambda = 0.1$). When $y_i \neq y_j$, $WD(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_i, \mathbf{x}_j)$, otherwise, $WD(\mathbf{x}_i, \mathbf{x}_j) = \lambda\, d(\mathbf{x}_i, \mathbf{x}_j)$. Thus the intra-class dissimilarity is linearly reduced while the inter-class dissimilarity keeps unchanged. This does offer some help to classification, but with limited ability to control the noise in the data. In detail, the range of $WD(\mathbf{x}_i, \mathbf{x}_j)$, whether $y_i = y_j$ or $y_i \neq y_j$, is $[0, +\infty]$. This means the noise, theoretically speaking, can

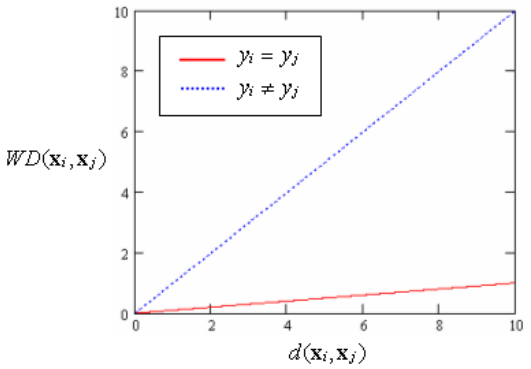


Fig. 3. Typical plot of $WD(\mathbf{x}_i, \mathbf{x}_j)$ as a function of $d(\mathbf{x}_i, \mathbf{x}_j)$

change the original dissimilarity to any value in $[0,+\infty]$. Consequently, so long as the noise is strong enough, the neighborhood relationship among the data points can be completely destroyed.

As for the dissimilarity $D(\mathbf{x}_i, \mathbf{x}_j)$, its properties and the corresponding advantages can be summarized as follows:

Property 1: When the Euclidean distance is equal, the inter-class dissimilarity is larger than the intra-class dissimilarity. This is similar to $WD(\mathbf{x}_i, \mathbf{x}_j)$ and makes $D(\mathbf{x}_i, \mathbf{x}_j)$ suitable for classification tasks.

Property 2: The inter-class dissimilarity is equal to or larger than $1-\alpha$ while the intra-class dissimilarity is less than 1. Thus no matter how strong the noise is, the inter-class and intra-class dissimilarity can be controlled in certain ranges respectively. This makes $D(\mathbf{x}_i, \mathbf{x}_j)$ suitable to apply to noisy data.

Property 3: Each dissimilarity function is monotone increasing with respect to the Euclidean distance. This ensures that the main geometric structure of the original data set, which is embodied by the Euclidean distances among data points, can be preserved.

Property 4: With the increasing of the Euclidean distance, the inter-class dissimilarity increases faster while the intra-class dissimilarity increases slower. This endows $D(\mathbf{x}_i, \mathbf{x}_j)$ with certain ability to "recognize" noise in the data. On the one hand, the intra-class distance is usually small. So the larger it is, the more possible the noise exists, and the slower $D(\mathbf{x}_i, \mathbf{x}_j)$ increases. On the other hand, the inter-class distance is usually large. So the smaller it is, the more possible the noise exists, and the slower $D(\mathbf{x}_i, \mathbf{x}_j)$ decreases. Both aspects indicate that $D(\mathbf{x}_i, \mathbf{x}_j)$ can gradually strengthen the power of noise suppression with the increase of noise-existing possibility.

Because of these good properties, the dissimilarity $D(\mathbf{x}_i, \mathbf{x}_j)$ can be used in the procedure of Isomap to address the robustness problem in visualization and classification. Since $D(\mathbf{x}_i, \mathbf{x}_j)$ integrates the class information, this algorithm is called supervised Isomap, denoted by S-Isomap. There are also three steps in S-Isomap. In the first step, neighborhood graph of the input data is constructed according to the dissimilarity between data points. The neighborhood can be defined as the K most similar points or the points whose dissimilarity is less than a certain value ε. In this paper, the neighborhood is defined as the K most similar points. If two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors, then connect them with an edge and assign $D(\mathbf{x}_i, \mathbf{x}_j)$ to the edge as a weight. The second step is similar to that of Isomap, but the shortest path between each pair of points is computed according to the weight of the edge rather than the Euclidean distance between the points. However, for convenience of discussion, the word "distance" is still used to indicate the sum of the weights along the shortest path. Finally, the third step of S-Isomap is the same as that of Isomap.

A. S-Isomap for Visualization

For visualization, the goal is to map the original data set into a (two or three dimensional) space that preserves as much as possible the intrinsic structure. Isomap can do this well when the input data are well sampled and have little noise. As for noisy data, which is common in real world, Isomap often fails to nicely visualize them. In this situation, the class labels of the data, if known, can be used to relieve the negative effect of noise. It is well known that points belonging to the same class are often closer to each other than those belonging to different classes. Under this assumption, S-Isomap can be used to recover the true manifold of the noisy data. In the first step of S-Isomap, $D(\mathbf{x}_i, \mathbf{x}_j)$ pulls points belonging to the same class closer and propels those belonging to different classes further away (*Property 1*). Recall that $D(\mathbf{x}_i, \mathbf{x}_j)$ has certain ability to "recognize" noise (*Property 4*), so when the data are noisy, this procedure can counteract the effect of noise and help to find the true neighborhood, and when the data are not noisy, this procedure hardly affects the neighborhood constructing. In both cases, $D(\mathbf{x}_i, \mathbf{x}_j)$ ensures to preserve the intrinsic structure of the data set (*Property 3*) and bounds the effect of noise in the data (*Property 2*). Thus S-Isomap is suitable to visualize the real-world data, whether noisy or not.

B. S-Isomap for Classification

For classification, the goal is to map the data into a feature space in which the members from different classes are clearly separated. S-Isomap can map the data into such a space where points belonging to the same class are close to each other while those belonging to different classes are far away from each other (*Property 1*). At the same time, the main structure of the original data can be preserved (*Property 3*). It is obvious that performing classification in such a space is much easier. When the data are noisy, S-Isomap can detect the existence of noise (*Property 4*) and limit the effect of noisy (*Property 2*). Thus S-Isomap can be used to design a robust classification method for real-world data. The procedure is similar to that of using Isomap in classification, which has been described in section II. To summarize, the classification has three steps as follows:

1. Map the data into a lower dimensional space using S-Isomap.
2. Construct Generalized Regression Network to approximate the mapping.
3. Map the given query using the Generalized Regression Network and then predict its class label using K-NN.

IV. Experiments

A. Visualization

1) Methodology

In many previous works on visualization, the results are mainly compared through examining the figures to point out which "looks" better. To compare the results more impersonally, some numerical criteria should be designed. When the distances between all pairs of data points are simultaneously changed by a linear transformation, the relationship of the points will not change, in other words, the intrinsic structure will not change. Recall that the goal of visualization is to faithfully represent the intrinsic structure of the input data. Thus the correlation coefficient between the distance vectors, i.e. the vectors that comprises the distances between all pairs of points, of the true structure and that of the recovered structure provides a good measurement of the validity of the visualization procedure. Suppose the distance vector of the true structure is $\mathbf{DV}$ and that of the recovered structure is $\mathbf{DV}'$, then the correlation coefficient between $\mathbf{DV}$ and $\mathbf{DV}'$ is computed by

$$\rho(\mathbf{DV},\mathbf{DV}') = \frac{\langle \mathbf{DVDV}' \rangle - \langle \mathbf{DV} \rangle \langle \mathbf{DV}' \rangle}{\sigma(\mathbf{DV})\sigma(\mathbf{DV}')}, \qquad (3)$$

where $\langle\ \rangle$ is the average operator, $\mathbf{DVDV}'$ represents the element-by-element product and σ is the standard deviation of the vector's elements. The larger the value of $\rho(\mathbf{DV},\mathbf{DV}')$, the better the performance of the visualization is.

The experiment data sets include two artificial ones. First, a two-dimensional structure with 50 classes is constructed as shown in Fig. 4 (a), where different colors denote different classes. Then 1000 points are randomly sampled from the structure as shown in Fig. 4 (b). After that, the points are separately embedded onto two nonlinear manifolds "S-curve" and "Swiss roll". At last, random Gaussian noise is added into the data. The mean of the noise is 0 and the standard deviation of the noise on each dimension is 3% of the largest distance on that dimension among the data points. The final two data sets are shown in Fig. 4 (c) and (d). Now the target of visualization is to discover the intrinsic structure (b) from one of the two three-dimensional data sets (c) and (d).

Isomap, LLE, WeightedIso and S-Isomap are all used to visualize the data sets. Their results are compared through the figures as well as the correlation values between the distance vectors. When computing the correlation values, not only the distance vector of all the points, but also that of the class centers is considered. They are denoted by $corr_{global}$ and $corr_{class}$ respectively. The value of $corr_{global}$ indicates the ability to discover the global structure while the value of $corr_{class}$ indicates the ability to discover the class distribution.

Isomap and S-Isomap are also compared on a real-world data set. The data set includes 505 gray-level face images, which are captured by a web camera from the same person with different poses. The face images have a resolution of 60×50 pixels, i.e. the data are 3000-dimensional vectors. The images contain some natural noise due to the limitations of the camera. Some typical images from the data set are shown in Fig. 5. The images are divided into 9 classes according to the face poses, i.e. Frontal (0), Left (1), Down-left (2), Down (3), Down-right (4), Right (5), Up-right (6), Up (7), Up-left (8). The indexes of the 9 classes are superimposed upon the face images in Fig. 5. Since the only difference among these images is face poses, the degrees of freedom should be two. Thus the visualization task is to show the input images in a two-dimensional space and reveal the different poses of the face.

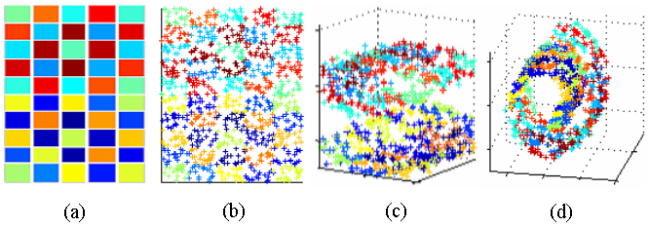


Fig. 4. Artificial data sets: (a) Class structure. (b) Random samples. (c) Samples imbedded on S-curve. (d) Samples imbedded on Swiss roll.

Fig. 5. Face images with 9 different poses. The class indexes are superimposed upon the typical face images.

In the following experiments, if not explicitly stated, the number of neighbors K is set to 10 (including the parameter K in Isomap, LLE, WeightedIso and S-Isomap), the parameter λ in WeightedIso is set to 0.1, the parameter α in S-Isomap is set to 0.5, and the parameter β in S-Isomap is set to be the average Euclidean distance between all pairs of data points.

2) Results

Fig. 6 and Fig. 7 show the visualization results of Isomap, LLE, WeightedIso and S-Isomap applied to the data sets "S-curve" and "Swiss roll" respectively. The corresponding correlation values between the results and the original structure are shown in Table I and Table II. To compare the different kinds of distortion generated by the four compared algorithms in the absence of nonlinearity and noise, these algorithms are also applied to the original two-dimensional data set shown in Fig. 4 (b). The results are shown in Fig. 8.

Fig. 6 and Fig. 7 reveal that both Isomap and LLE fail to nicely visualize the data, which is consistent with their relatively low correlation values in Table I and Table II. As mentioned in section II, the poor performances of Isomap and LLE are due to that the noise in the data sets disturbs the

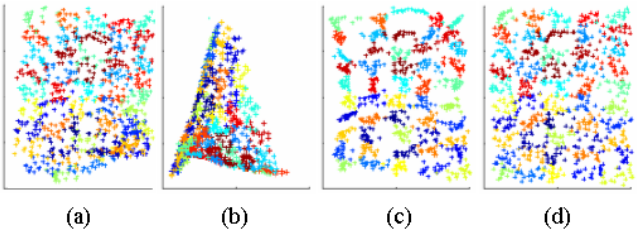


Fig. 6. Visualization of the "S-curve" data set: (a) Isomap. (b) LLE. (c) WeightedIso. (d) S-Isomap.

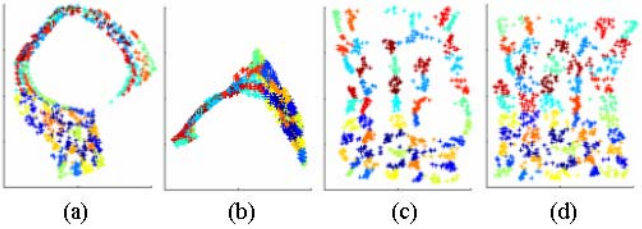


Fig. 7. Visualization of the "Swiss roll" data set: (a) Isomap. (b) LLE. (c) WeightedIso. (d) S-Isomap.

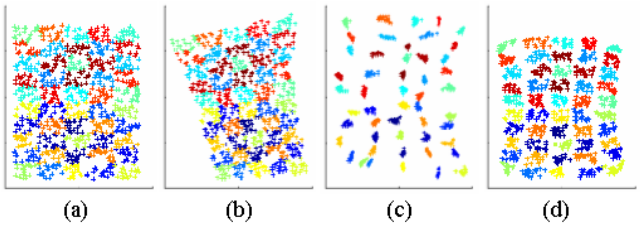


Fig. 8. Results of the compared algorithms applied to the original two-dimensional data set: (a) Isomap. (b) LLE. (c) WeightedIso. (d) S-Isomap.

TABLE I
CORRELATION VALUES OF THE "S-CURVE" DATA SET

	Isomap	LLE	WeightedIso	S-Isomap
$corr_{global}$	0.9277	0.5765	0.9855	0.9880
$corr_{class}$	0.9366	0.5960	0.9921	0.9945

TABLE II
CORRELATION VALUES OF THE "SWISS ROLL" DATA SET

	Isomap	LLE	WeightedIso	S-Isomap
$corr_{global}$	0.8082	0.7218	0.9775	0.9807
$corr_{class}$	0.8195	0.7411	0.9781	0.9811

neighborhood of the original structure. Fig. 6 and Fig. 7 also reveal that both WeightedIso and S-Isomap are able to catch the main structure of the data. However, WeightedIso forcefully distorts the original structure and makes the points in the same class tend to shrink to the center. On the other hand, S-Isomap can reproduce the original structure more faithfully because of the good properties of $D(\mathbf{x}_i, \mathbf{x}_j)$. This is also supported by the correlation values of S-Isomap in Table I and Table II, which are the highest ones of both $corr_{global}$ and $corr_{class}$ on both data sets. All in all, both the figures and the correlation values indicate that S-Isomap is the most powerful method to visualize the intrinsic structure of the noisy data among the four methods.

It can be seen from Fig. 8 that in the absence of nonlinearity and noise, Isomap almost perfectly reproduces the original structure, LLE and S-Isomap can also obtain satisfying results, but WeightedIso fails to recover the original structure.

TABLE III
COMPARISON OF ISOMAP AND S-ISOMAP WITH DIFFERENT VALUES OF K

K	Isomap		S-Isomap	
	$corr_{global}$	$corr_{class}$	$corr_{global}$	$corr_{class}$
6	0.9837	0.9923	0.9858	0.9927
8	0.9284	0.9377	0.9880	0.9944
10	0.9277	0.9366	0.9880	0.9945
12	0.9236	0.9325	0.9874	0.9937
14	0.9243	0.9334	0.9863	0.9925
16	0.9252	0.9344	0.9875	0.9937
18	0.9163	0.9265	0.9875	0.9942
20	0.8858	0.8984	0.9891	0.9958
Avg.	0.9268	0.9365	0.9874	0.9939
Std.	0.0269	0.0259	0.0010	0.0010

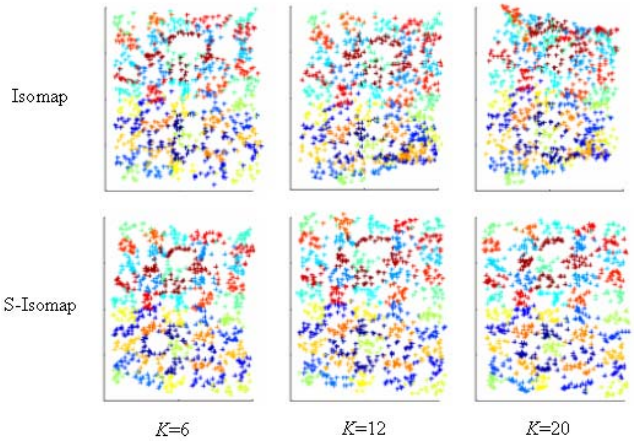


Fig. 9. The visualization results of Isomap and S-Isomap corresponding to K=6, 12 and 20. The rows correspond to different algorithm, and the columns correspond to different values of K.

Although both WeightedIso and S-Isomap change the original distances among the data points according to their class labels, S-Isomap exceeds WeightedIso in visualization because of its ability to "recognize" the noise and to work properly in the absence of nonlinearity.

When applying Isomap, the number of neighbors K must be carefully adjusted to get a reasonable result. To find out whether S-Isomap is also sensitive to the value of K, it is tested on the "S-curve" data set while K increases from 6 to 20 with 2 as the interval. Then the results are compared with those of Isomap. The correlation values are tabulated in Table III, where "Avg." means average and "Std." means standard deviation. The visualization results corresponding to K=6, 12 and 20 are shown in Fig. 9.

Table III reveals that the average correlation values of S-Isomap are significantly larger than those of Isomap while the standard deviations of S-Isomap are significantly smaller than that of Isomap. Fig. 9 reveals that as K increases, the results of Isomap become significantly worse, but those of S-Isomap do not change much. Both Table III and Fig. 8 indicate that S-Isomap is more accurate and less sensitive to K than Isomap. Thus S-Isomap can be easily applied to real-world data without much effort on parameter tuning.

The visualization results of Isomap and S-Isomap on the face images are shown in Fig. 10 and Fig. 11 respectively. In the

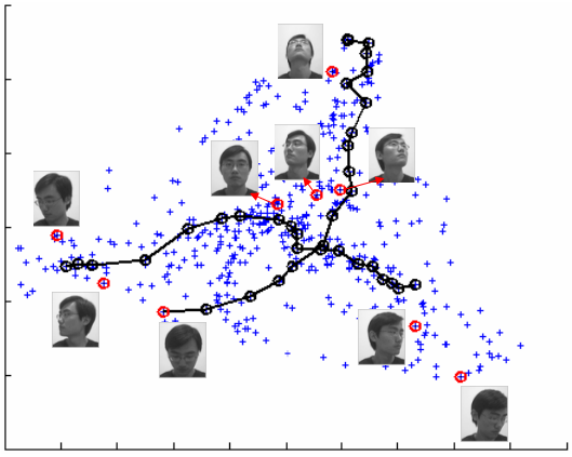


Fig. 10. Visualization result of Isomap on the face images

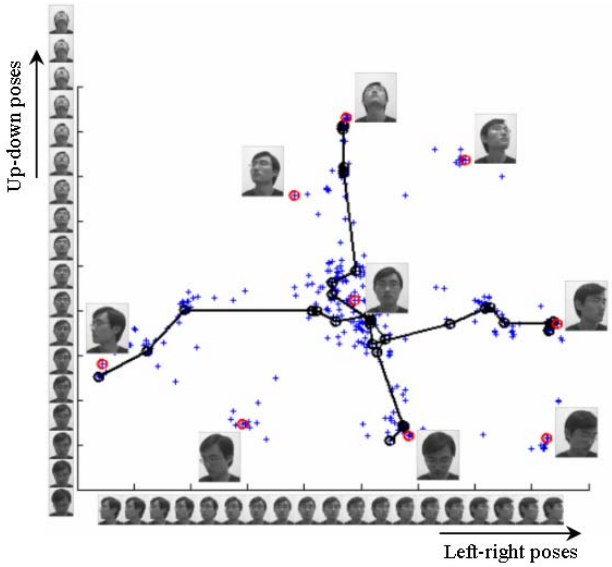


Fig. 11. Visualization result of S-Isomap on the face images

figures, the corresponding points of the successive images from up to down and from left to right are marked by black circles and linked by lines. The nine critical face samples shown in Fig. 5 are marked by gray circles and shown near the corresponding points.

It can be seen from Fig. 10 that Isomap can hardly reveal the different face poses. Part of the up-down line is almost parallel to the left-right line. And the arrangement of the nine face samples is tanglesome. On the other hand, in Fig. 11, the up-down line is approximately perpendicular to the left-right line. Thus the horizontal axis represents the left-right poses, and the vertical axis represents the up-down poses. Moreover, the nine face samples are mapped to the approximately right positions corresponding to the face poses. All these indicate that S-Isomap is able to approximately visualize the different poses of the face images. It is worth mentioning that the face images are only roughly divided into 9 classes, so the points in Fig. 11 tend to congregate in 9 groups corresponding to the 9 classes. If the face images can be more detailedly divided (of course there should be more than one image in one class, otherwise there will not be any "intra-class" distances), the visualization result will be more accurate.

B. Classification

1) Methodology

In this section, S-Isomap is compared with Isomap and WeightedIso in classification. Some other well-established classification methods including *K* nearest neighbor classifier [4], [6], BP neural network [10], J4.8 decision tree [18] and SVM [14], [15] are also compared.

The method of using S-Isomap in classification has been described in section III.B. For convenience of discussion, "S-Isomap" is still used to indicate the three-step classification method that uses S-Isomap as the first step in this experiment. The meanings of "S-Isomap" can be easily distinguished from the context. Isomap and WeightedIso are used in classification in the similar way except that in the first step, S-Isomap is replaced by Isomap and WeightedIso respectively, and the corresponding classification methods are still denoted by "Isomap" and "WeightedIso". In the dimensionality reduction procedure, the dimensionality of the data is reduced to half of the original.

The parameters for most of the methods are determined empirically through ten-fold cross validation. That is, for each parameter, several values are tested through ten-fold cross validation and the best one is selected. For S-Isomap, different values of α between 0.25 and 0.6 are tested. For WeightedIso, different values of λ between 0.05 and 0.5 are tested. When applying *K*-NN algorithm, several values of *K* from 10 to 40 are tested. The BP neural networks and the J4.8 decision trees are constructed using WEKA [18] with the default settings. For SVM, both linear kernel and nonlinear kernel (radial basis function with bias 1.0) are tested and the best result is reported.

The data sets include two artificial ones and thirteen real-world ones. The two artificial data sets are "S-curve" and "Swiss roll" which have been used in the visualization experiments. But now the task changes to classification. The thirteen real-world data sets are all from UCI machine learning repository [1]. Instances that have missing values are removed. Information of all the data sets is summarized in Table IV.

It can be seen from Table IV that some data sets have categorical attributes. The previous discussion on S-Isomap is only about continuous data. As for categorical attributes, the distance is computed through VDM proposed by Stanfill and Waltz [12].

On each data set, ten times ten-fold cross validation is run. That is, in each time, the original data set is randomly divided into ten equal-sized subsets while keeping the proportion of the instances in different classes. Then, in each fold, one subset is used as testing set and the union of the remaining ones is used as training set. After ten folds, each subset has been used as testing set once. The average result of these ten folds is recorded. This procedure is repeated ten times and gets ten results for each compared algorithm. After that, the pair wise one-tailed *t*-test is performed on the results of S-Isomap paired with every other algorithm at the significance level 0.025. It is worth mentioning that the ten times ten-fold cross validation is

TABLE IV
DATA SETS USED IN CLASSIFICATION

Data Set		Size	Classes	Attributes		
Abbr.	Name			Total	Categorical	Continuous
bal	Balance-scale	625	3	4	0	4
bre	Breast-cancer	277	2	9	9	0
dia	Diabetes	768	2	8	0	8
gla	Glass	214	7	9	0	9
hea	Heart-Cleveland	296	2	13	8	5
iri	Iris	150	3	4	0	4
let	Letter-recognition	2000	26	16	0	16
liv	Liver-disorders	345	2	6	0	6
lym	Lymphography	148	4	18	15	3
scu	S-curve	1000	50	3	0	3
son	Sonar	208	2	60	0	60
soy	Soybean	562	19	35	35	0
swi	Swiss roll	1000	50	3	0	3
vow	Vowel	990	11	10	0	10
wav	Waveform	2500	3	40	0	40

TABLE V
MEAN PRECISIONS OF THE COMPARED CLASSIFICATION METHODS

Data Set	S-Isomap	Isomap	WeightedIso	*K*-NN	BP	J4.8	SVM
bal	0.8953	0.7282	0.8742	0.8867	0.9075	0.7831	**0.9090**
bre	**0.7769**	0.7553	0.7711	0.7593	0.7382	0.7358	0.7650
dia	0.7525	0.7042	0.7435	0.7474	0.7508	0.7455	**0.7737**
gla	**0.7141**	0.6163	0.7027	0.6897	0.6751	0.6796	0.6309
hea	0.8326	0.7857	0.8215	0.8120	0.8083	0.7689	**0.8328**
iri	0.9600	0.9293	0.9593	0.9627	**0.9653**	0.9447	0.9627
let	**0.8408**	0.5909	0.8057	0.8138	0.7731	0.7051	0.7681
liv	0.6306	0.5690	0.6254	0.6470	**0.6935**	0.6656	0.5750
lym	**0.8590**	0.8067	0.8310	0.8509	0.8194	0.8037	0.8328
scu	**0.7687**	0.6681	0.6811	0.7437	0.7202	0.7125	0.7524
son	0.8740	0.6519	0.8616	0.8275	0.8240	0.7316	**0.8813**
soy	**0.9082**	0.8015	0.9081	0.8767	0.8944	0.9069	0.9058
swi	0.8318	0.5339	0.6223	0.8268	0.8252	0.7987	**0.8376**
vow	**0.9855**	0.6576	0.9776	0.9477	0.7951	0.7971	0.8669
wav	0.8279	0.8221	0.8378	0.8037	0.8359	0.7397	**0.8584**
Avg.	**0.8305**	0.7081	0.8015	0.8130	0.8017	0.7679	0.8102

completely separate with those used in parameter tuning, i.e. once the parameters for a certain method on a certain data set are determined, the data set is re-divided ten times and tested.

2) Results

The mean precisions, i.e. the average correct rates of the ten times ten-fold cross validation, of S-Isomap, Isomap, WeightedIso, *K*-NN, BP neural network (denoted by BP), J4.8 decision tree (denoted by J4.8) and SVM on the 15 data sets are tabulated in Table V. On each data set, the best performance is emphasized by bold. The average performance of each method on all data sets is also given in the last row of the table.

Table V reveals that S-Isomap gives the best performance on 7 data sets, and is close to the best method on the remaining 8 data sets. SVM performs best on 6 data sets and BP on 2. According to the average performance, the seven methods can be sorted as: S-Isomap > *K*-NN > SVM > BP > WeightedIso > J4.8 > Isomap. Here *K*-NN is in the second place. Recall that *K*-NN is also used in the last step of S-Isomap. However, *K*-NN in S-Isomap is performed in a space whose dimensionality is much lower. Thus S-Isomap is less computation and storage consuming than pure *K*-NN when the dimensionality of the input data is relatively high.

To compare the robustness of these methods, i.e. how well the particular method *m* performs in different situations, a criteria is defined similar to the one used by Vlachos et al. [16]. In detail, the relative performance of *m* on a particular data set is represented by the ratio b_m of its mean precision p_m and the highest mean precision among all the compared methods:

$$b_m = \frac{p_m}{\max_k p_k} \quad (4)$$

The best method m^* on that data set has $b_{m^*} = 1$, and all the other methods have $b_m \leqslant 1$. The larger the value of b_m, the better the performance of the method m is in relation to the best performance on that data set. Thus the sum of b_m over all data sets provides a good measurement of the robustness of the method m. A large value of the sum indicates good robustness.

Fig. 12 shows the distribution of b_m of each method over the 15 data sets. For each method, the 15 values of b_m are stacked and the sum is given on top of the stack. Fig. 12 reveals that S-Isomap has the highest sum value. In fact, the b_m values of S-Isomap are equal or very close to 1 on all the data sets, which means S-Isomap performs very well in different situations. Thus S-Isomap is the most robust method among the compared methods.

The results of pair wise one-tailed *t*-test performed on S-Isomap paired with every other algorithm are tabulated in Table VI, where the results "significantly better", "significantly worse" and "not significantly different" are denoted by 1, –1 and 0 respectively.

In Table VI, the average of each column is larger than 0.00, which means the average performances of S-Isomap over the data sets are better than all the other methods. Specially, the average of the "Isomap" column is 1.00, which means S-Isomap performs significantly better than Isomap on all data sets. Table VI also reveals that the average of each row is larger than 0.00 except for that of the "wav" row, which means the performances of S-Isomap are better than most other methods on almost all data sets. Specially, the average of the row "bre", "gla", "let", "lym", "scu" and "vow" are all equal to 1.00, which means on these 6 data sets, S-Isomap performs significantly better than all the other methods. As for the data set "wav", the average value is 0.00. This is likely due to that the number of the samples is large while the number of the classes is small, so the information given by each class label is relatively little. Since such information is used to guide the dimensionality reduction in S-Isomap, when it is relatively limited, S-Isomap may be not significantly superior to other methods.

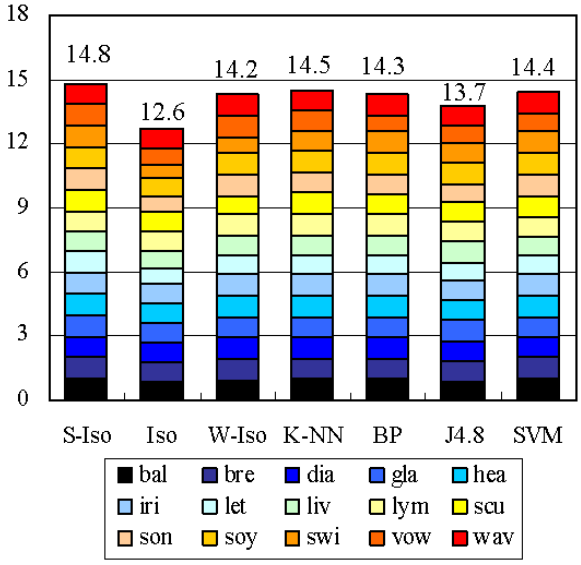


Fig. 12. Robustness of the compared methods

V. Conclusions

In this paper, an improved version of Isomap, namely S-Isomap, is proposed for robust visualization and classification. S-Isomap uses the class information of the given data to guide the manifold learning. The procedure of using S-Isomap to reduce dimensionality is called supervised nonlinear dimensionality reduction. The utilization of the class information helps to deal with the noise in the data and thus

TABLE VI
Pair wise one-tailed *t*-test on S-Isomap versus the other six methods

Data Set	Isomap	WeightedIso	*K*-NN	BP	C4.5	SVM	Avg.
bal	1	1	1	-1	1	-1	0.33
bre	1	1	1	1	1	1	1.00
dia	1	1	0	0	0	-1	0.17
gla	1	1	1	1	1	1	1.00
hea	1	1	1	1	1	0	0.83
iri	1	0	0	0	1	0	0.33
let	1	1	1	1	1	1	1.00
liv	1	1	0	-1	-1	1	0.17
lym	1	1	1	1	1	1	1.00
scu	1	1	1	1	1	1	1.00
son	1	1	1	1	1	0	0.83
soy	1	0	1	1	0	0	0.50
swi	1	1	0	0	1	-1	0.33
vow	1	1	1	1	1	1	1.00
wav	1	-1	1	-1	1	-1	0.00
Avg.	1.00	0.73	0.73	0.40	0.73	0.20	

makes S-Isomap more robust in both visualization and classification. In the visualization experiments, S-Isomap is compared with Isomap, LLE and WeightedIso. Both the figures and the correlation values between the recovered structures and the intrinsic structure indicate that S-Isomap is more powerful than the other three methods in visualization. In the classification experiments, S-Isomap is compared with Isomap and WeightedIso in classification on both artificial and real-world data sets. Some other well-established classification methods including *K*-NN, BP network, J4.8 decision tree and SVM are also compared. The results show that S-Isomap is also an accurate and robust technique for classification.

When the given data are scattered in faraway clusters, the neighborhood graph of them may be disconnected. Unfortunately, neither Isomap nor S-Isomap can deal with such kind of data. In the future work, more efforts should be taken to tackle this problem.

When S-Isomap is used for classification, the explicit mapping function from the original data space to the feature space is learned by some other nonlinear interpolation techniques. Thus the final generalization error is brought by not only S-Isomap, but also the interpolation method. If a more natural way for S-Isomap to map the query into the feature space can be found, the generalization ability of S-Isomap will be further improved. This is also worth to be further studied in the future.

VI. Acknowledgments

The comments and suggestions from the anonymous reviewers greatly improved this paper.

References

[1] C. Blake, E. Keogh, and C. J. Merz, "UCI repository of machine learning databases" [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998.

[2] P. Comon, "Independent component analysis: a new concept?" *Signal Processing*, vol.36, no.3, pp.287-314, 1994.

[3] T. Cox and M. Cox, *Multidimensional Scaling*, London: Chapman & Hall, 1994.

[4] T. K. Ho, "Nearest Neighbors in random subspaces," in *Lecture notes in computer Science: Advances in Pattern Recognition*, Berlin: Springer, pp.640-948, 1998.

[5] I. T. Jolliffe, *Principal Component Analysis*, New York: Springer, 1986.

[6] D. Lowe, "Similarity metric learning for a variable-kernel classifier," *Neural Computation*, vol.7, no.1, pp.72-85, 1995.

[7] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*, London: Academic Press, 1979.

[8] J. Quinlan, *C4.5: Programs for Machine Learning*, San Francisco, CA: Morgan-Kaufmann, 1993.

[9] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol.290, no.5500, pp.2323-2326, 2000.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature*, vol.323, no.9 pp.318-362, 1986.

[11] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research*, vol.4, pp.119-155, 2003.

[12] C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Communications of the ACM*, vol.29, no.12, pp.1213-1228, 1986.

[13] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol.290, no.5500, pp.2319-2323, 2000.

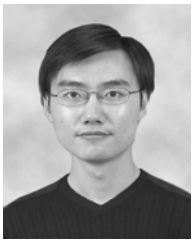
[14] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 1995.

[15] V. Vapnik, *Statistical Learning Theory*, New York: John Wiley and Sons, 1998.

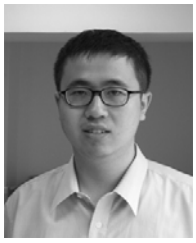
[16] M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios, and N. Koudas, "Non-linear dimensionality reduction techniques for classification and visualization," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, pp.645-651, 2002.

[17] P. D. Wasserman, *Advanced Methods in Neural Computing*, New York: Van Nostrand Reinhold, 1993.

[18] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools With Java Implementations*, San Francisco, CA: Morgan Kaufmann, 1999.



Xin Geng received the BSc and MSc degrees in computer science from Nanjing University, China, in 2001 and 2004 respectively. Currently he is a research and teaching assistant at the Department of Computer Science & Technology of Nanjing University, and a member of the LAMDA group. His research interests are in machine learning, pattern recognition and computer vision. He has been a reviewer for several international conferences.



De-Chuan Zhan received the BSc degree in computer science from Nanjing University, China, in 2004. He got some awards such as peoples' scholarship for outstanding undergraduate. Currently he is a first year MSc student at the Department of Computer Science & Technology of Nanjing University, supervised by Prof. Zhi-Hua Zhou. He is also a member of the LAMDA group. His research interests are in machine learning and pattern recognition.

Zhi-Hua Zhou (S'00-M'01) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honor. He joined the Department of Computer Science & Technology of Nanjing University as a lecturer in 2001, and is a professor and head of the LAMDA group at present. His research interests are in artificial intelligence, machine learning, data mining, pattern recognition, information retrieval, neural computing, and evolutionary computing. In these areas he has published over 40 technical papers in refereed international journals or conference proceedings. He has won the Microsoft Fellowship Award (1999), the National Excellent Doctoral Dissertation Award of China (2003), and the Award of National Outstanding Youth Foundation of China (2004). He is an associate editor of *Knowledge and Information Systems* (Springer), and on the editorial boards of *Artificial Intelligence in Medicine* (Elsevier) and *International Journal of Data Warehousing and Mining* (Idea Group). He served as the organising chair of the 7th Chinese Workshop on Machine Learning (2000), program co-chair of the 9th Chinese Conference on Machine Learning (2004), and program committee member for numerous international conferences. He is a senior member of China Computer Federation (CCF) and the vice chair of CCF Artificial Intelligence & Pattern Recognition Society, a councilor of Chinese Association of Artificial Intelligence (CAAI), the vice chair and chief secretary of CAAI Machine Learning Society, and a member of IEEE and IEEE Computer Society.