M-Isomap: Orthogonal Constrained Marginal Isomap for Nonlinear Dimensionality Reduction

Zhao Zhang, *Student Member, IEEE*, Tommy W. S. Chow, *Senior Member, IEEE*, and Mingbo Zhao, *Student Member, IEEE*

***Abstract*—Isomap is a well-known nonlinear dimensionality reduction (DR) method, aiming at preserving geodesic distances of all similarity pairs for delivering highly nonlinear manifolds. Isomap is efficient in visualizing synthetic data sets, but it usually delivers unsatisfactory results in benchmark cases. This paper incorporates the pairwise constraints into Isomap and proposes a *marginal Isomap* (M-Isomap) for manifold learning. The pairwise Cannot-Link and Must-Link constraints are used to specify the types of neighborhoods. M-Isomap computes the shortest path distances over constrained neighborhood graphs and guides the nonlinear DR through separating the interclass neighbors. As a result, large margins between both inter- and intraclass clusters are delivered and enhanced compactness of intracluster points is achieved at the same time. The validity of M-Isomap is examined by extensive simulations over synthetic, University of California, Irvine, and benchmark real Olivetti Research Library, YALE, and CMU Pose, Illumination, and Expression databases. The data visualization and clustering power of M-Isomap are compared with those of six related DR methods. The visualization results show that M-Isomap is able to deliver more separate clusters. Clustering evaluations also demonstrate that M-Isomap delivers comparable or even better results than some state-of-the-art DR algorithms.**

***Index Terms*—Isomap, manifold learning, nonlinear dimensionality reduction (DR), pairwise constraints (PCs), visualization.**

I. Introduction

HIGH-DIMENSIONAL data analysis has been attracting considerable attention, as most of emerging applications are related with the high-dimensional attributes, such as gene expressions and face recognition. However, high-dimensional attributes usually contain redundant information. Therefore, extracting the informative attributes that hold the required information is important. Note that human eyes are impossible to visually perceive the high-dimensional representations of samples, so reducing the dimensionality of data to two or three and visualizing embedded data have become increasingly useful for multivariate analysis. Also, visualization via dimensionality reduction (DR) plays an important role in revealing the intrinsic characteristics, e.g., local or nonlinear structures, of high-dimensional data. DR aims to compute a set of optimal or close-optimal projections for mining the low-dimensional structures of high-dimensional observations [1], [2].

In the last decades, many unsupervised and discriminant DR algorithms have been proposed, such as *principal component analysis* (PCA) [4], *multidimensional scaling* (MDS) [5], and *linear discriminant analysis* (LDA) [4]. PCA, LDA, and MDS are linear methods widely applied in data visualization and classification, making them appealing in revealing the linear relations between two sets of features. However, nonlinear structures are rather common in real data. Obviously, it is advantageous to apply the nonlinear DR or data visualization techniques to real data [6]. *Laplacian eigenmaps* [2], *locally linear embedding* (LLE) [1], and *Isomap* [7] are three representative manifold learning algorithms proposed for nonlinear DR and data visualization. These methods are efficient at visualizing artificial data sets and are powerful to handle nonlinear data. However, they are unsupervised methods, so they cannot make use of any supervised prior information for discrimination. Also, they fail to identity the types (inter- or intraclass) of neighborhoods. To address these issues, *supervised Isomap* (S-Isomap) [11] and *supervised LLE* (SLLE) [21] have been recently proposed by enabling the inclusion of class labels directly. SLLE guides the discriminant learning by increasing the preobtained distances artificially between interclass points and leaving the distances unchanged for those intraclass points. S-Isomap drives the discriminant learning through defining a new distance metric to enhance interclass dissimilarity over intraclass similarity. The same idea of SLLE and S-Isomap is to pick the neighbors of each point from the same class and then separate interclass points through improving intraclass compactness. Satisfactory results are reported if data sets are well sampled with relatively convex intrinsic geometry. However, as can be observed from the benchmark simulations of this paper, SLLE and S-Isomap are not so powerful for handing multiple-class real cases.

In this work, we incorporate the pairwise Cannot-Link (CL) and Must-Link (ML) constraints [13]–[15] induced from the neighborhood graph into the Isomap to guide the discriminant manifold learning. We then propose a pairwise-constrained marginal Isomap (*M-Isomap*) for visualization and nonlinear DR. M-Isomap considers both discriminant information and geometrical information of data. Note that pairwise constraints (PCs) can sometimes be achieved with minimal human effort and can provide more supervision information compared with the class labels. More importantly, PC sets are flexible in regulating the supervised information. In other words, we can

Manuscript received October 18, 2011; revised March 20, 2012; accepted May 10, 2012. This work was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU8/CRF/09). This paper was recommended by Editor H. Qiao.

The authors are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: itzzhang@ee.cityu.edu.hk; eerchow@cityu.edu.hk; mzhao4@student.cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2202901

employ all available constraints or partial constraints for the simulations. Apparently, it is a great advantage to apply PC for discriminant manifold learning. Different from the ideas of S-Isomap and SLLE, M-Isomap handles the inter- and intraclass neighbors independently. Also, M-Isomap approximates the geodesic distances by the shortest path distances based on the constrained neighborhood graphs. Specifically, M-Isomap aims to preserve the pairwise geodesic distances between intraclass similarity pairs and aims to separate the interclass neighbors in the reduced embedding space. Based on the extracted informative features, large margins between inter- and intraclass clusters are organized, delivering a strong interclass discrimination power. A related PC-guided DR method is the *constraint margin maximization* (CMM) criterion [14], [15]. CMM maximizes the constraint margin of data sets, but it only focuses on interclass discrimination without considering the data geometry. Another related algorithm is called *average neighborhood margin maximization* (ANMM) [3] which also considers the types of neighborhoods, but the ANMM framework is inherently different from ours in two aspects. First, ANMM directly drives the discrimination by class labels, so it will be rigid in regulating supervised information. Second, our algorithm is formulated on the classical Isomap learning framework, so their objectives are different. By visualizing synthetic and benchmark data sets, the results demonstrate that our M-Isomap method is powerful for handling multiple-class real problems.

This paper is outlined as follows. Section II briefly reviews Isomap and S-Isomap. Section III details M-Isomap. Section IV describes the simulations and evaluates the M-Isomap using synthetic, UCI, and real ORL, YALE, and CMU PIE face databases. Finally, this paper is concluded in Section V.

II. Preliminaries

A. Classical Isomap Algorithm

For a given set of data points $x_1, x_2, \ldots, x_N$ in n-dimensional input space $\Box^n$, Isomap aims at seeking an optimal subspace that best preserves the geodesic distances between points, i.e., the geodesic distance between paired points x_i and x_j is as close to the Euclidean distance between their low-dimensional representations y_i and y_j as possible. Isomap proceeds DR in three steps. First, Isomap finds the neighbors of points on the manifold $\mathcal{M}$. The ε-neighborhood [2] and k-neighborhood [2] can be used here. Then, a weighted undirected neighborhood graph $G = (V, E)$, where node $v_i \in V$ corresponds to point x_i, is constructed based on the Euclidean distances. An edge $e(x_i, x_j) \in E$ with weight $d_X(x_i, x_j)$ is put between v_i and v_j if x_i and x_j are mutually neighbors, where $d_X(x_i, x_j)$ is the Euclidean distance between points x_i and x_j. Second, Isomap estimates the geodesic distances $d_M(x_i, x_j)$ between all pairs of points on the manifold through computing all the shortest path distances $d_G(x_i, x_j)$ in G. To find the shortest paths, Dijkstra's algorithm [16] and Floyd's algorithm [17] can be applied. After computing the shortest path distances, Isomap initializes $d_G(x_i, x_j) = d_X(x_i, x_j)$ if x_i and x_j are connected by an edge and $d_G(x_i, x_j) = \infty$ if otherwise. Then, for each value of $p = 1, 2, \ldots, N$, in turn, replace all the entries $d_G(x_i, x_j)$ by $\min\{d_G(x_i, x_j), d_G(x_i, x_p) + d_G(x_p, x_j)\}$. The matrix of final values $D_G = \{d_G(x_i, x_j)\}$ contains the shortest path distances between all pairs of points on the manifold. At last, MDS is used to the matrix of distances $D_G = \{d_G(x_i, x_j)\}$ to construct the low-dimensional coordinates y_i of Y in a d-dimensional space $\Box^d$ $(d \le n)$. The Isomap criterion is defined as

$$J(Y) = \min_{Y} \|\tau(D_G) - \tau(D_Y)\|_F \tag{1}$$

where D_Y denotes the matrix of Euclidean distances $\{d_Y(y_i, y_j) = \|y_i - y_j\|\}$ in $\Box^d$, $\tau(D_Y)$ is the corresponding Euclidean inner product matrix, $\tau(D_G)$ is the shortest path inner product matrix, $\|\Box\|$ is the l^2-norm, and $\|\Box\|_F$ is the Frobenius matrix norm. In a least square sense, Isomap expects $Y^{\mathrm{T}}Y$ to be close to $\tau(D_G)$ [10], where $^{\mathrm{T}}$ denotes the transpose of a matrix. According to Tenenbaum *et al.* [7] and Mardia *et al.* [8], the global minimum of (1) can be obtained by setting the coordinates y_i to the top d eigenvectors of $\tau(D_G)$. Let $S_{i,j} = (D_G)^2_{i,j}$ and the centering matrix $H = I - (1/N)ee^{\mathrm{T}}$ [9], where I is an identity matrix and e is a vector of all ones; then, $\tau(D_G) = -HSH/2$. Finally, the embedding Y is obtained as $[\sqrt{\lambda_1}\tau_1, \sqrt{\lambda_2}\tau_2, \ldots, \sqrt{\lambda_d}\tau_d]^{\mathrm{T}}$, where $\{\tau_i\}_{i=1}^{d}$ denotes the eigenvectors according to the first d leading eigenvalues of the matrix $\tau(D_G)$.

B. S-Isomap Algorithm

By considering the class label information of data, a supervised version of Isomap, namely, *S-Isomap*, has been recently proposed. Let $l(x_i) \in \{1, 2, \ldots, c\}$, with $i = 1, 2, \ldots, N$, be the class labels of points $\{x_i\}_{i=1}^{N}$; S-Isomap proceeds the supervised nonlinear DR using similar steps as Isomap. S-Isomap optimizes the same problem as Isomap, i.e., S-Isomap also seeks $Y^{\mathrm{T}}Y$ to be as close to $\tau(D_G)$ as possible. The major difference between Isomap and S-Isomap lies in the step of constructing the neighborhood graph and setting the weights. To improve the compactness of intraclass points and pull interclass points away, S-Isomap defines the following dissimilarity between samples x_i and x_j according to their associated class labels $l(x_i)$ and $l(x_j)$:

$$\widehat{d_X}(x_i, x_j) = \begin{cases} \sqrt{1 - \exp\left(\frac{-d_X^2(x_i, x_j)}{\beta}\right)}, & \text{if } l(x_i) = l(x_j) \\ \sqrt{\exp\left(\frac{d_X^2(x_i, x_j)}{\beta}\right)} - \alpha, & \text{if } l(x_i) \ne l(x_j) \end{cases} \tag{2}$$

where parameter β is added to prevent $\widehat{d_X}(x_i, x_j)$ from increasing too fast when $\widehat{d_X}(x_i, x_j)$ is relatively large. The parameter α is added to ensure that a smaller value of dissimilarity is imposed for interclass points than for intraclass points [11]. Different from the first step of Isomap, the neighborhood relationships of sample points and the neighborhood graph $\widetilde{G}$ of input data in S-Isomap are determined based on the dissimilarity $\widehat{d_X}(x_i, x_j)$ between the points. The neighborhood of x_i is defined as the k most close points or the points whose dissimilarity is less than ε. When defining the weights, $\widehat{d_X}(x_i, x_j)$ is set to the

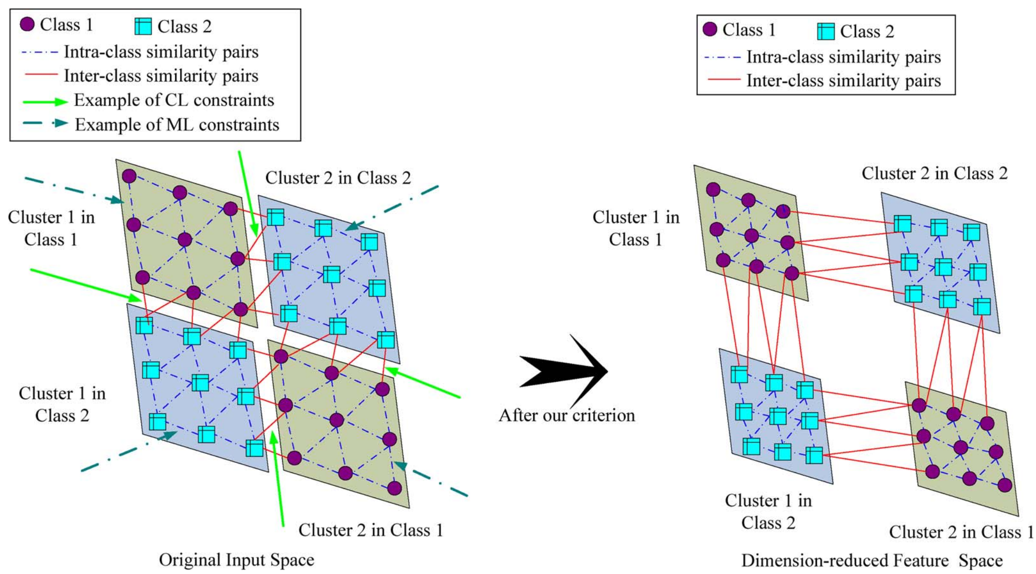


Fig. 1. Geometrical interpretation of our proposed pairwise-constrained M-Isomap criterion.

edges linking v_i and v_j on graph $\widetilde{G}$ if x_i and x_j are neighbors. The latter two steps of S-Isomap are similar in spirit to those of Isomap except that S-Isomap computes the shortest path between each pair of points according to the edge weights rather than the Euclidean distances. Based on the weights $\widehat{d_X}(x_i, x_j)$, the interclass dissimilarity is larger than intraclass dissimilarity, delivering a high discriminative power on S-Isomap. It is noted that, except the k number in the neighborhood definition, another two key parameters α and β, which may significantly influence the neighborhoods of samples and final embeddings, are involved. However, one must notice that estimating such tuning parameters will never be easy and straightforward.

III. ORTHOGONAL M-ISOMAP

A. *Pairwise-Constrained Neighborhood Graphs*

This work models the M-Isomap algorithm from a pairwise-constrained perspective. For data set $X = [x_1, x_2, \ldots, x_N]$ with the class labels $l(x_i) \in \{1, 2, \ldots, c\}$, with $i = 1, 2, \ldots, N$, we conduct the k-nearest neighbor search (NNS) to find the neighbors of each point based on the Euclidean distances between the points and then construct a neighborhood graph $G = (V, E)$. Denote by $N_+^{(x_i)}$ the k neighbor set of $v_i \in V$. We put weight $\in \{0, 1, -1\}$ to the edge $e(x_i, x_j) \in E$ linking x_i and x_j to distinguish the types (inter- or intraclass) of the neighboring pairs. More specifically, $e(x_i, x_j) = 1$, when $x_j \in N_+^{(x_i)}$ or $x_i \in N_+^{(x_j)}$, if $l(x_j) = l(x_i)$; $e(x_i, x_j) = -1$, when $x_j \in N_+^{(x_i)}$ or $x_i \in N_+^{(x_j)}$, if $l(x_j) \neq l(x_i)$; and $e(x_i, x_j) = 0$, if $x_j \notin N_+^{(x_i)}$ and $x_i \notin N_+^{(x_j)}$. Based on the aforementioned definitions, the ML and CL constraint sets for M-Isomap are defined as

$$S_{ML} = \{(x_i, x_j) | e(x_i, x_j) = 1, v_i \in V, v_j \in V, l(x_j) = l(x_i)\} \tag{3}$$

$$S_{CL} = \{(x_i, x_j) | e(x_i, x_j) = -1, v_i \in V, v_j \in V, l(x_j) \neq l(x_i)\}. \tag{4}$$

By removing edges $e(x_i, x_j)$ with negative weights from the graph G, we obtain an ML-constrained neighborhood graph $G_{ML} = (V, E_{ML})$ that has the same vertices as G. Similarly, by removing edges $e(x_i, x_j)$ with positive weights from G, we can obtain a CL-constrained neighborhood graph $G_{CL} = (V, E_{CL})$ with the same vertices as graph G.

Here, we consider a binary-class case in the left of Fig. 1, in which each class has two separate clusters, i.e., *multimodal*. Based on our definitions, we divide the similarity neighboring pairs to inter- and intraclass. Note that we show some typical examples of the ML and CL constraints by arrows. For efficient DR and feature extraction, it is desired that the compactness of neighboring pairs constrained by ML can be enhanced, while high separation between neighboring pairs constrained by CL is achieved, because we want to separate them in the reduced space. This paper will focus on addressing this issue. After performing our M-Isomap for nonlinear DR, the embeddings can be geometrically illustrated in the right of Fig. 1, from which we find that large margins between inter- and intraclass clusters are delivered. More importantly, natural clusters within each class, i.e., multimodal structure, can be preserved, since intraclusters will not be projected into a single cluster.

Since our technique integrates the PCs and exhibits large margins between different clusters, this method is referred to as constrained *M-Isomap*. We elaborate in the next section the definition and objective function of the presented M-Isomap algorithm.

B. *Objective Function*

The main objective of M-Isomap is to compute the embeddings from the pairwise-constrained geodesic space of the nonlinear data manifold. Like Isomap and S-Isomap, M-Isomap proceeds the nonlinear DR in three steps. The first step is performed by determining the k neighbors or by choosing all points within a fixed radius ε on the manifold. Accordingly, the ML and CL constraint sets can be computed. Based on the constrained neighborhood graphs G_{ML} and G_{CL}, $d_X^{(ML)}(x_i, x_j) = \|x_i - x_j\|$ is reset on the edges $e(x_i, x_j) \in E_{ML}$ linking pair $(x_i, x_j) \in S_{ML}$ over graph G_{ML}. Similarly, a weight $d_X^{(CL)}(x_i, x_j) = \|x_i - x_j\|$ will be reset on edges $e(x_i, x_j) \in E_{CL}$ connecting pair $(x_i, x_j) \in S_{CL}$ over the graph G_{CL}. The geodesic distances $d_M^{ML}(x_i, x_j)$ and $d_M^{CL}(x_i, x_j)$ between all pairs of constrained points on the manifold are estimated in the second step, where $d_M^{ML}(x_i, x_j)$ and $d_M^{CL}(x_i, x_j)$ denote the geodesic distances between pairs $(x_i, x_j) \in S_{ML}$ and S_{CL}, respectively. To estimate $d_M^{ML}(x_i, x_j)$ and $d_M^{CL}(x_i, x_j)$, M-Isomap approximates $d_M^{ML}(x_i, x_j)$ and $d_M^{CL}(x_i, x_j)$ with the shortest path distances $d_G^{ML}(x_i, x_j)$ and $d_G^{CL}(x_i, x_j)$ between all vertex pairs over graphs G_{ML} and G_{CL}, respectively. To improve the compactness of intraclass similarity pairs, we normalize the edge weights $d_X^{(ML)}(x_i, x_j)$ to $\widetilde{d}_X^{(ML)}(x_i, x_j) = d_X^{(ML)}(x_i, x_j) / \max(d_X^{(ML)})$, where $\max(d_X^{(ML)})$ is the biggest edge weight in $d_X^{(ML)}(x_i, x_j)$. It is worth noting that, based on the normalized edge weights $\widetilde{d}_X^{(ML)}(x_i, x_j)$ over G_{ML}, the shortest paths keep consistent with those over $d_X^{(ML)}(x_i, x_j)$, but the shortest path distances $d_G^{ML}(x_i, x_j)$ are shortened by computing from $\widetilde{d}_X^{(ML)}(x_i, x_j)$. The procedures of computing the shortest path distances are similar to those of Isomap. Finally, M-Isomap uses the trace ratio (TR) optimization [18]–[20] to the matrices $D_G^{ML} = \{\widetilde{d}_G^{ML}(x_i, x_j)\}$ and $D_G^{CL} = \{d_G^{CL}(x_i, x_j)\}$ for computing the embeddings of the original samples in a reduced d-dimensional Euclidean space.

Recall that S-Isomap and Isomap seek the matrix $\tau(D_Y)$ over all points to be as close to $\tau(D_G)$ as possible. As a result, for some complex distributed real data sets, the interclass neighbors are likely to be congregated in the reduced output space. Note that S-Isomap optimizes the same problem as Isomap and achieves interclass discrimination via enhancing the interclass dissimilarity over the intraclass dissimilarity, so S-Isomap is incapable of delivering large margins between inter- and intraclass clusters, although a new distance metric has been defined to replace the original Euclidean distances. Unlike S-Isomap and Isomap, to deliver large margins for interclass discrimination, M-Isomap handles the ML- and CL-constrained points independently. For the ML constraint set, M-Isomap optimizes the following criterion:

$$J_{ML}(Y) = \min_Y \sum_{(x_i, x_j) \in ML} \left\| d_G^{ML}(x_i, x_j) - d_Y^{ML}(y_i, y_j) \right\|^2 \quad (5)$$

where y_i is the low-dimensional representation of x_i and $d_Y^{ML}(y_i, y_j) = \|y_i - y_j\|$ is the Euclidean distance between y_i and y_j in the reduced space. That is, M-Isomap preserves the neighborhood relationship via seeking the distance metric $d_Y^{ML}(y_i, y_j)$ to be as close to the shortest path distance $d_G^{ML}(x_i, x_j)$ as possible when data pair $(x_i, x_j) \in S_{ML}$. As a result, the compactness of data pairs $(x_i, x_j) \in S_{ML}$ can be effectively enhanced. By extending (5) to all data points of the data set, we have the following matrix form:

$$J_{ML}(Y) = \min_Y \left\| \tau\left(D_G^{ML}\right) - \tau(D_Y) \right\|_F^2 \quad (6)$$

where $\tau(D_G^{ML})$ denotes the shortest path inner product matrix constrained by ML. Similarly, M-Isomap aims at optimizing the following criterion for points in the CL set:

$$J_{CL}(Y) = \max_Y \sum_{(x_i, x_j) \in CL} \left\| d_Y^{CL}(y_i, y_j) - d_G^{CL}(x_i, x_j) \right\|^2 \quad (7)$$

where $d_Y^{CL}(y_i, y_j) = \|y_i - y_j\|$ denotes the Euclidean distance between the low-dimensional representations of the corresponding pair $(x_i, x_j) \in S_{CL}$. That is, M-Isomap aims at separating x_i from point x_j by seeking the distance $d_Y^{CL}(y_i, y_j)$ to be as far to the shortest path distance $d_G^{CL}(x_i, x_j)$ as possible. As a result, more separated embeddings of pairs $(x_i, x_j) \in S_{CL}$ are obtained, delivering large inter- and intracluster margins. With all the points considered, the aforementioned criterion can be written as

$$J_{CL}(Y) = \max_Y \left\| \tau(D_Y) - \tau\left(D_G^{CL}\right) \right\|_F^2 \quad (8)$$

where $\tau(D_G^{CL})$ denotes the shortest path inner product matrix based on the CL constraint set.

Fig. 2 shows the geometric interpretation of the M-Isomap criteria in (5) and (7). We take the neighbors (x_a, x_b) as an example to show the effect of the PCs on the embeddings, where $d_Y(x_a, x_b)$ is the Euclidean distance between x_a and x_b and $d_G^{ML}(y_a, y_b)$, $d_G^{CL}(y_a, y_b)$, and $d_G(x_a, x_b)$ are the shortest path distances. The paths directed by arrows are the shortest paths. After minimizing the criterion in (5), the difference between the distances $d_G^{ML}(y_a, y_b)$ and $d_Y^{ML}(y_a, y_b)$ is as small as possible, since enhanced compactness of the data pairs $(x_i, x_j) \in S_{ML}$ can be gotten. Resembling Isomap, minimizing $\|\tau(D_G^{ML}) - \tau(D_Y)\|_F^2$ is equivalent to maximizing $Y\tau(D_G^{ML})Y^{\mathrm{T}}$. In contrast, if neighboring pair $(x_a, x_b) \in S_{CL}$, i.e., interclass neighbors, after maximizing the criterion in (7), $d_Y^{CL}(y_a, y_b)$ will be as large as possible by comparing with the shortest path distance $d_G^{CL}(y_a, y_b)$, since we aim at separating x_a from x_b under this case. We similarly minimize $Y\tau(D_G^{CL})Y^{\mathrm{T}}$ instead of maximizing $\|\tau(D_Y) - \tau(D_G^{CL})\|_F^2$. As a result, the embedded Euclidean distance $d_Y^{CL}(y_a, y_b)$ is far larger than $d_Y^{ML}(y_a, y_b)$ in the reduced space. Resembling [7] and [9], let $Q_{i,j}^{ML} = (D_G^{ML})_{i,j}^2$, $Q_{i,j}^{CL} = (D_G^{CL})_{i,j}^2$, and matrix $H = I - (1/N)ee^{\mathrm{T}}$; we similarly have $\tau(D_G^{ML}) = -HQ^{ML}H/2$ and $\tau(D_G^{CL}) = -HQ^{CL}H/2$.

By combining (6) and (8), we equivalently optimize the following TR problem [18]–[20]:

$$Y^* = \arg\max_{YY^{\mathrm{T}}=I} \frac{\mathrm{tr}\left(Y\tau\left(D_G^{ML}\right)Y^{\mathrm{T}}\right)}{\mathrm{tr}\left(Y\tau\left(D_G^{CL}\right)Y^{\mathrm{T}}\right)}. \quad (9)$$

From (9), the low-dimensional embedding coordinates in Y can be achieved subject to the orthogonal constraint $YY^{\mathrm{T}} = I$. The detailed implementation procedures of M-Isomap are summarized in Table I.

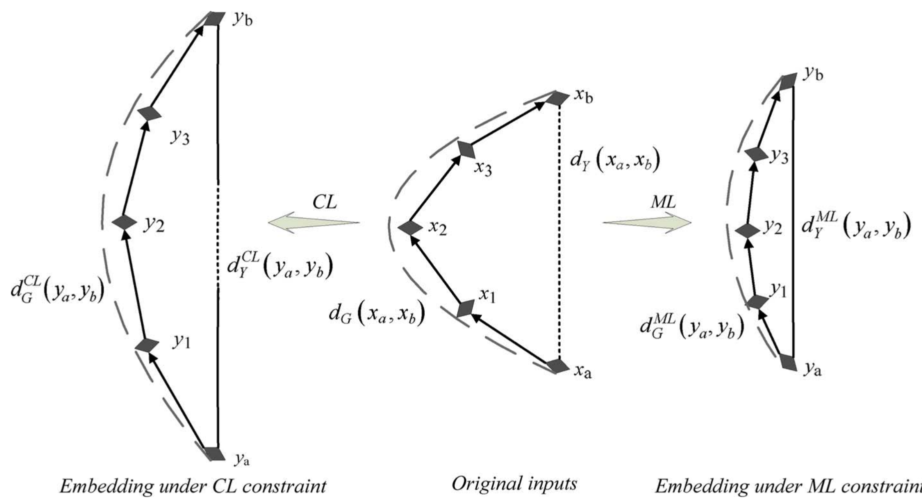


Fig. 2. Geometric interpretation of the maximization and minimization criteria of M-Isomap.

TABLE I
M-ISOMAP ALGORITHM

Input: Data matrix $X = [x_1, x_2, \ldots x_N]$ and k value in NNS
Output: Low-dimensional coordinates y_i in Y
1. Determine the neighbors of each point and construct the *ML* and *CL* constraint sets S_{ML} and S_{CL};
2. Construct the *k*-neighborhood graph (or the ball of radius ε) with the defined weights {-1,1} for each pair of neighbors. Then, the two constrained neighborhood graphs G_{ML} and G_{CL} can be achieved;
3. Compute the shortest path distances $d_G^{ML}(x_i, x_j)$ and $d_G^{CL}(x_i, x_j)$ of each pair $(x_i, x_j) \in S_{ML}$, S_{CL} based on graphs G_{ML} and G_{CL} by using Dijkstra's or Floyd's algorithm. Store the squares of these distances in matrices D_G^{ML} and D_G^{CL} accordingly;
4. Return the embedding coordinates Y by maximizing $Tr\left(Y\tau\left(D_G^{ML}\right)Y^{\mathrm{T}}\right) / Tr\left(Y\tau\left(D_G^{CL}\right)Y^{\mathrm{T}}\right)$ with respect to the orthogonal constraint $YY^{\mathrm{T}} = I$ using the TR optimization procedures in Table 2.

TABLE II
TR OPTIMIZATION FOR SOLVING M-ISOMAP

Input: Two symmetric matrices $\tau\left(D_G^{ML}\right)$ and $\tau\left(D_G^{CL}\right)$
Output: The optimum TR value λ^* and embedding Y^*
1. Initialize $\lambda^v = 0$, step $v = 0$;
2. Solve the eigenvalue problem $\left(\tau\left(D_G^{ML}\right) - \lambda^v \tau\left(D_G^{CL}\right)\right)\psi = \tau^v \psi$ and calculate the eigenvectors $\left\{\psi_\delta^v\right\}_{\delta=1}^d$ of $\left(\tau\left(D_G^{ML}\right) - \lambda^v \tau\left(D_G^{CL}\right)\right)$ by conducting eigen-decomposition;
3. Low-dimensional embedding Y^v is gained as the transpose of the eigenvectors $\psi_1^v, \psi_2^v, \ldots, \psi_d^v$ according to the *d* leading eigenvalues $\left\{\tau_\delta^v\right\}_{\delta=1}^d$ of the matrix $\left(\tau\left(D_G^{ML}\right) - \lambda^v \tau\left(D_G^{CL}\right)\right)$;
4. Renew λ^{v+1} by $Tr\left(Y^v \tau\left(D_G^{ML}\right)\left(Y^v\right)^{\mathrm{T}}\right) / Tr\left(Y^v \tau\left(D_G^{CL}\right)\left(Y^v\right)^{\mathrm{T}}\right)$;
5. If $\left|\lambda^{v+1} - \lambda^v\right| < \varepsilon$, go to step 6; else $v = v + 1$, steps 2-4 repeat;
6. Output the optimal trace ratio value $\lambda^* = \lambda^v$, and optimal matrix $Y^* = \arg\max_{YY^{\mathrm{T}}=I} Tr\left(Y\left(\tau\left(D_G^{ML}\right) - \lambda^* \tau\left(D_G^{CL}\right)\right)Y^{\mathrm{T}}\right)$.

C. Effective Solution With TR Optimization

In this section, we will show how to compute the TR problem $\max_{YY^{\mathrm{T}}=I} \mathrm{tr}(Y\tau(D_G^{ML})Y^{\mathrm{T}})/\mathrm{tr}(Y\tau(D_G^{CL})Y^{\mathrm{T}})$ of M-Isomap. For given two symmetric matrices $\tau(D_G^{ML}) = -HQ^{ML}H/2$ and $\tau(D_G^{CL}) = -HQ^{CL}H/2$, the iterative trace ratio (ITR) algorithm [19], [20] can be applied to solve this TR problem. The ITR algorithm tackles the TR problem by directly optimizing the objective $\mathrm{tr}(Y^v\tau(D_G^{ML})(Y^v)^{\mathrm{T}})/\mathrm{tr}(Y^v\tau(D_G^{CL})(Y^v)^{\mathrm{T}})$ if the row vectors of Y^v are orthogonal together. Given λ^v at each iteration v, the optimum matrix Y^v can be obtained from the following trace difference problem [19]:

$$Y^v = \arg\max_{YY^{\mathrm{T}}=I} \mathrm{tr}\left(Y\tau\left(D_G^{ML}\right)Y^{\mathrm{T}} - \lambda^v Y\tau\left(D_G^{CL}\right)Y^{\mathrm{T}}\right). \quad (10)$$

Then, the ITR method renews λ^{v+1} as the TR value given by Y^v: $\lambda^{v+1} = \mathrm{tr}(Y^v\tau(D_G^{ML})(Y^v)^{\mathrm{T}})/\mathrm{tr}(Y^v\tau(D_G^{CL})(Y^v)^{\mathrm{T}})$ until convergence of the algorithm. Theoretical analysis shows that the ITR algorithm delivers specific solutions and converges to the global optimum [19], [20]. Note that ITR initializes Y^0 to be an arbitrary orthogonal matrix; thus, ITR may be unstable because of the randomness. Most importantly, the orthogonal initialized Y^0 is difficult to be constructed, and a bad initialization may greatly increase the number of iterations in the optimizations. In this work, we initialize $\lambda^0 = 0$ instead of initializing Y^0 to be the rowly orthogonal matrix. The algorithmic procedures of using ITR to solve our M-Isomap problem are described in Table II. Note that, under TR criterion, the low-dimensional coordinate vectors are orthogonal together, so in the linearized cases, the similarity can be effectively preserved if it is based on the Euclidian distance [20].

IV. SIMULATION RESULTS AND ANALYSIS

In this section, we examine the proposed M-Isomap algorithm by visualizing synthetic, UCI, and real data sets. The visual and clustering performances are compared with those of the unsupervised Isomap, LLE, Hessian LLE (HLLE) [12], and discriminant approaches including S-Isomap, SLLE, CMM [14], [15], and ANMM [3]. For each method, k-neighborhood is used for finding the neighbors. For S-Isomap, parameters α and β are defined the same as in [11]. The parameter in the objective function of CMM is set to 0.5. In this study, three benchmark real face databases are tested. The first one is the *ORL* database (available at http://www.uk.research.att.com/facedatabase.html), the second one is the *YALE* database (available at http://cvc.yale.edu/projects/yalefaces/yalefaces.html), and the third one is the *CMU PIE* database [23]. In the simulations, original images were resized to 32 × 32 pixels due to computational consideration. Therefore, each face is denoted by a 1024-dimensional vector in the image space. In our study, we aim at visualizing the whole data sets, i.e., PC sets are created based on the class labels of all points. To numerically evaluate the results, the following evaluation metric is applied. All the simulations were performed on a personal computer with Intel(R) Core i5-650 CPU at 3.20 GHz (3.19 GHz 4G).

A. Similarity Evaluation Metric

Clustering performance is evaluated by comparing the obtained cluster label of each datum with that provided by the data corpus. The accuracy rates and normalized mutual information (MI) metric [22], [24] are used to measure the clustering. Given a point x_i, let r_i and f_i be the obtained cluster label and the provided class label. The clustering accuracy is defined as

$$AC = \frac{\sum_{i=1}^{N} \delta\left(f_i, \mathrm{Map}(r_i)\right)}{N} \tag{11}$$

where N is the total amount of data, $\delta(p, q)$ is the delta function which equals one if $p = q$ and equals zero if otherwise, and $\mathrm{Map}(r_i)$ is the permutation mapping function, mapping each r_i to the equivalent label from the data corpus. Let C denote the set of clusters obtained from the ground truth and C' denote the set of clusters obtained from our method. Their MI metric $MI(C, C')$ is defined by

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} \Pr\left(c_i, c'_j\right) \cdot \log_2 \frac{\Pr\left(c_i, c'_j\right)}{\Pr(c_i) \cdot \Pr\left(c'_j\right)} \tag{12}$$

where $\Pr(c_i)$ and $\Pr(c'_j)$ are the probabilities that a point randomly selected from the data corpus belongs to the clusters c_i and c'_j, respectively, and $\Pr(c_i, c'_j)$ is the joint probability that the arbitrarily selected point belongs to clusters c_i and c'_j. $MI(C, C')$ takes values between zero and $\max(H(C), H(C'))$ as inputs, where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. To simplify comparisons between different pairs of cluster sets, the following normalized MI is commonly used:

$$\overline{MI}(C, C') = \frac{MI(C, C')}{\max\left(H(C), H(C')\right)}. \tag{13}$$

It is easy to check that $\overline{MI}(C, C')$ ranges from zero to one, i.e., $\overline{MI}$ is equal to one if the two sets of clusters are identical and zero if the two sets are completely independent.

B. Visualizing Synthetic Data Set

This section evaluates M-Isomap using a challenging synthetic 3-D "four moons" data set. This data set has four clusters. Each cluster denoted as a single class has 150 points and looks like a moon. The original distribution is shown in Fig. 3(a), in which each symbol plus a color denotes a class. We see clearly that interclass points of the data set are linearly inseparable, so it is a challenging problem for DR. This data set is mainly used to evaluate the intraclass compactness and interclass separation. The between-class separation is directly reflected by the produced margins between interclusters.

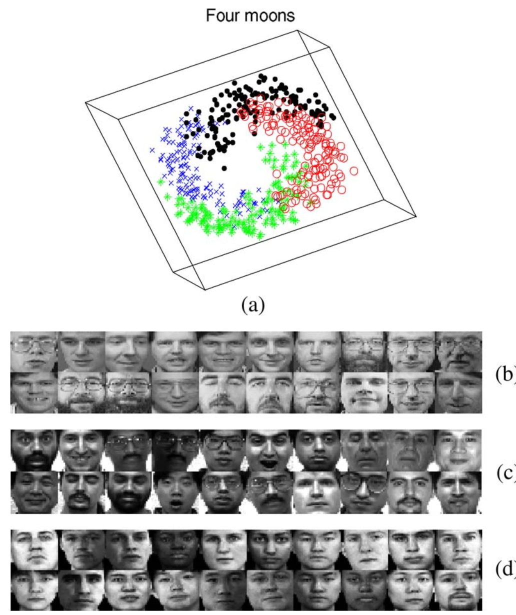


Fig. 3. Original distribution and typical images of the data sets. (a) Synthetic. (b) ORL. (c) YALE. (d) CMU PIE.

We show the 2-D embeddings of the "four moons" data set in Fig. 4. For each method, the k value in k-neighborhood is set to 45. For CMM and M-Isomap, 60% ML constraints plus 60% CL constraints, which are randomly selected from the constraint sets, are used in all simulations if without special remarks. Note that CMM is a semisupervised method. In our simulations, we apply all points to construct both the supervised and unsupervised parts. Observing the results, we find the following.

1) The embeddings of the clusters are still overlapped in the results of unsupervised Isomap, LLE, HLLE, and discriminant ANMM, CMM, S-Isomap, and SLLE, although they can reveal the intrinsic structures of the set. Thus, these methods are likely to produce relatively high clustering errors.
2) Compared with other methods, our M-Isomap provides a clear separation on the embeddings of the clusters. It is interesting to note that the clusters are linearly separable in the M-Isomap embedding space. We also observe that large inter- and intraclass margins are produced by M-Isomap and enhanced intraclass compactness is delivered. At the same time, our M-Isomap method clearly keeps the neighborhood relations between the intraclass and intracluster points.

We then apply the clustering evaluation measure induced by the k-means algorithm to numerically compare the clustering performances of each method. In our simulations, the clustering measurement process is performed as follows. First, points are embedded into a low-dimensional output space; then, k-means

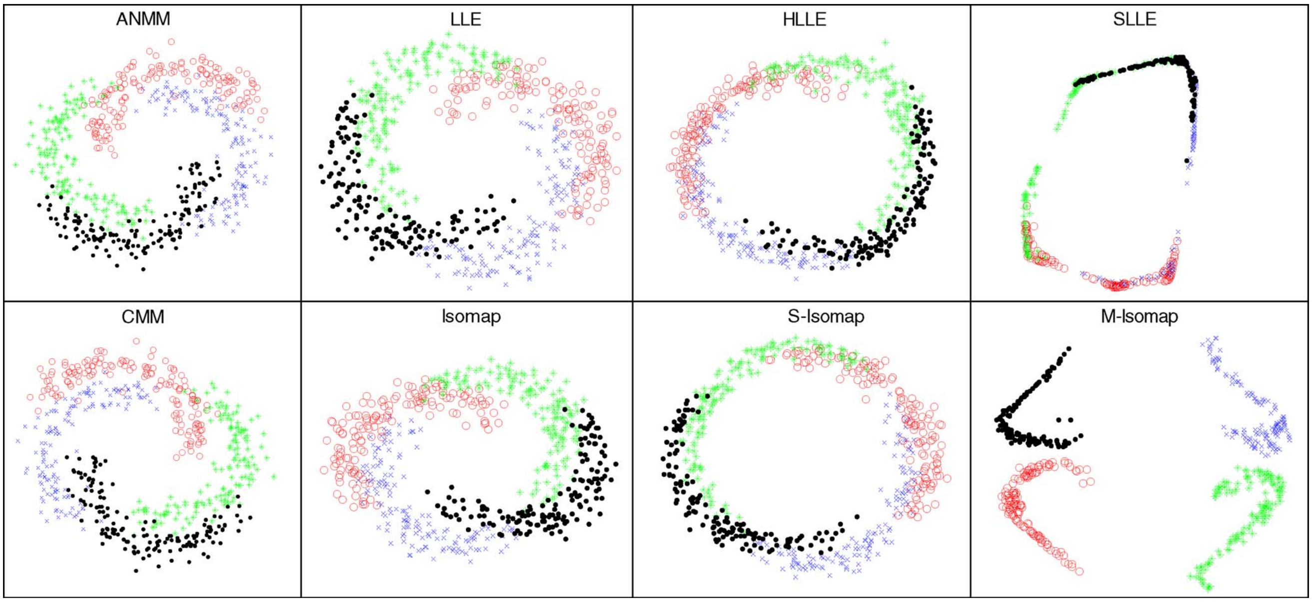


Fig. 4. Two-dimensional embedding result of the "four moons" data set by each method.

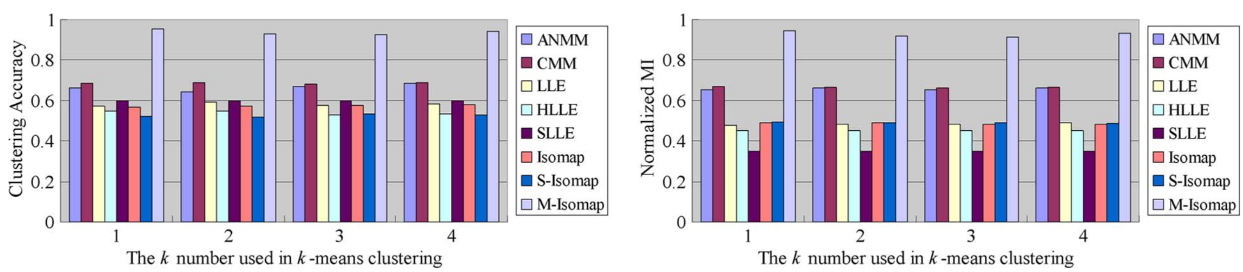


Fig. 5. Clustering evaluation result of each method on the "four moons" data set.

clustering is performed. For each k value used in the k-means algorithm, the k-means clustering is applied 200 times with different initializations. The averaged clustering accuracy and normalized MI over different k values are recorded in Fig. 5. We have the following observations.

1) The clustering results of Isomap, LLE, HLLE, S-Isomap, and SLLE are comparable in most cases. ANMM and CMM are slightly better than Isomap, LLE, HLLE, S-Isomap, and SLLE in terms of clustering.
2) The clustering performance is greatly improved with our M-Isomap, compared with the other methods.

C. *Visualizing UCI Data Set*

In this study, we test M-Isomap by visualizing the Synthetic Control Chart Time Series data set or simply Control-Chart (available at http://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series) from the UCI ML repository. This data set contains 600 examples of synthetically generated control charts and 60 attributes for each example. This data set has six different classes of control charts, so each class has 100 data points. We show the 2-D embeddings of the data set in Fig. 6. For each method, the k number used in k-neighborhood is set to 65. Observing Fig. 6, we see that ANMM, CMM, Isomap, LLE, HLLE, S-Isomap, and SLLE deliver unsatisfactory results in achieving enhanced interclass separation and intraclass compactness. Compared with the other evaluated methods, our M-Isomap algorithm is able to deliver more separated manifolds and can organize enhanced compactness within each natural cluster.

We record the numerical clustering results in Fig. 7. The setting for the k-means clustering process is the same as above. The clustering accuracy and normalized MI are also averaged over 200 initializations. From Fig. 7, we find the following.

1) For each algorithm, the normalized MI is slightly higher than the corresponding clustering accuracy.
2) The clustering results of ANMM, CMM, Isomap, LLE, HLLE, S-Isomap, and SLLE are comparable together. More specifically, their clustering accuracies and normalized MI are around 0.6.
3) We once again observe that M-Isomap gains the best results in each case, implying that applying M-Isomap to extract the features and visualize the embedded data is promising.

D. *Face Manifold Visualization on ORL Database*

In this study, the ORL face database is tested. The database contains 40 distinct persons with ten images per person. These images are taken at different time instances, with varying lighting conditions, facial expressions, and details. There are certain typical images with different expressions, and the details

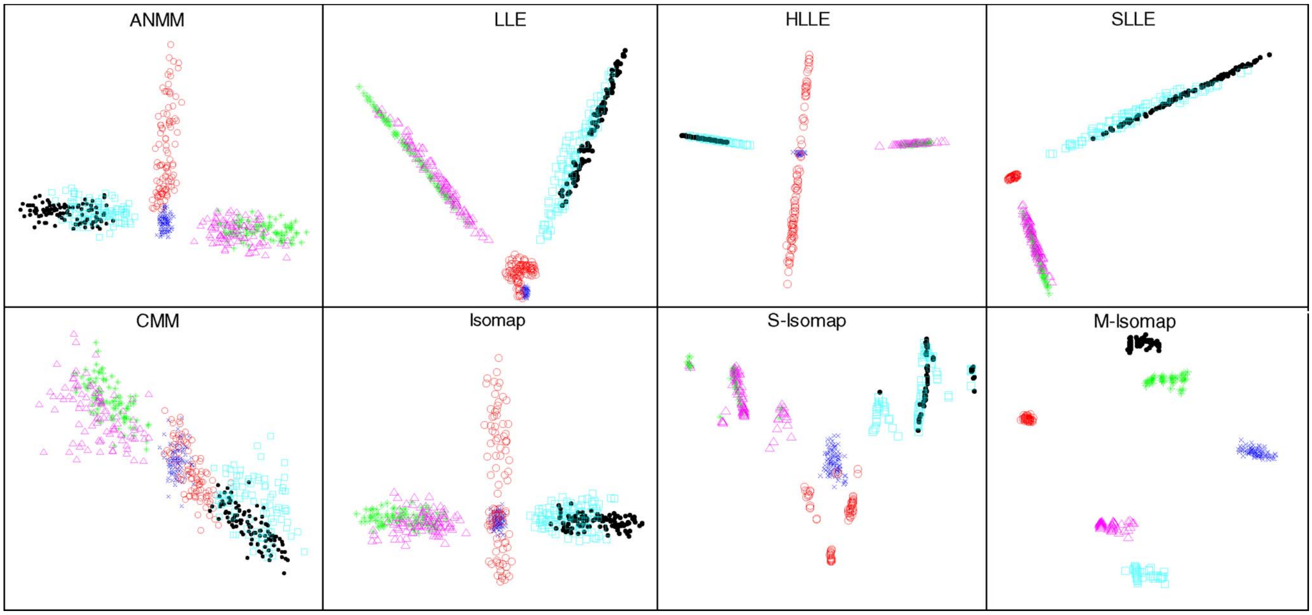


Fig. 6. Two-dimensional embedding result of the Control-Chart data set by each method.

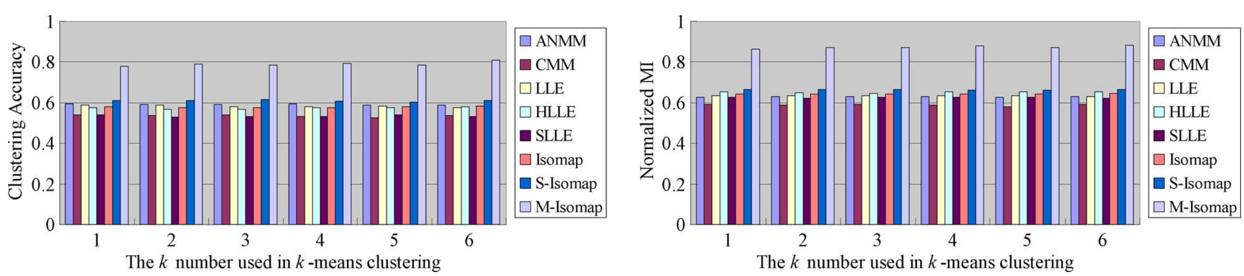


Fig. 7. Clustering evaluation result of each method on the Control-Chart data set.

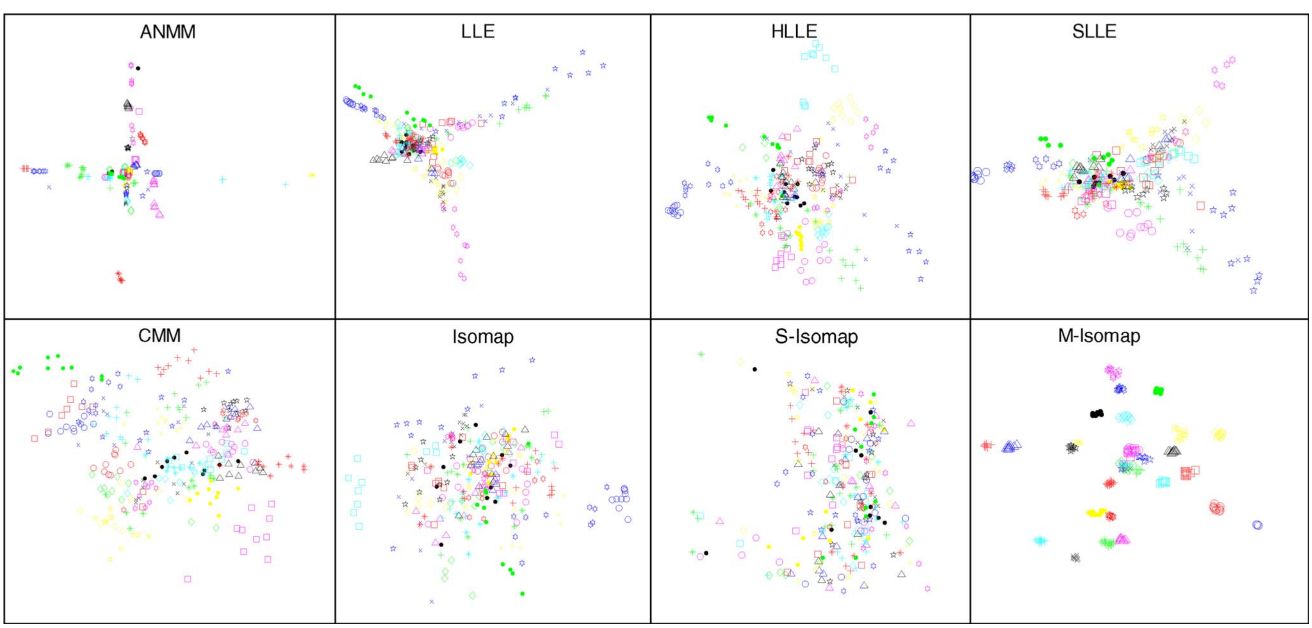


Fig. 8. Two-dimensional embedding result of the ORL face data set by each method.

of the database are shown in Fig. 3(b). In this simulation, we visualize the face images of the first 30 persons. That is, a 30-class problem is created. We show the 2-D embeddings of the faces in Fig. 8. The k number in k-neighborhood is set to 15 for each method. From Fig. 8, the following observations can be obtained.

1) Although the intrinsic manifolds can be preserved by CMM, Isomap, LLE, HLLE, S-Isomap, and SLLE to

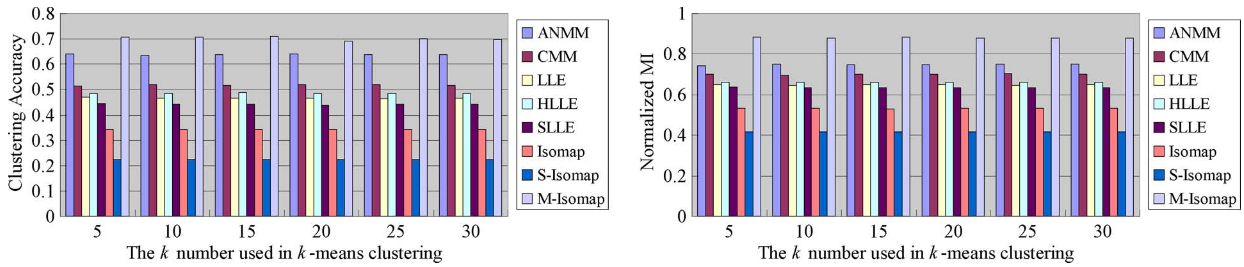


Fig. 9. Clustering evaluation result of each method on the ORL face data set.

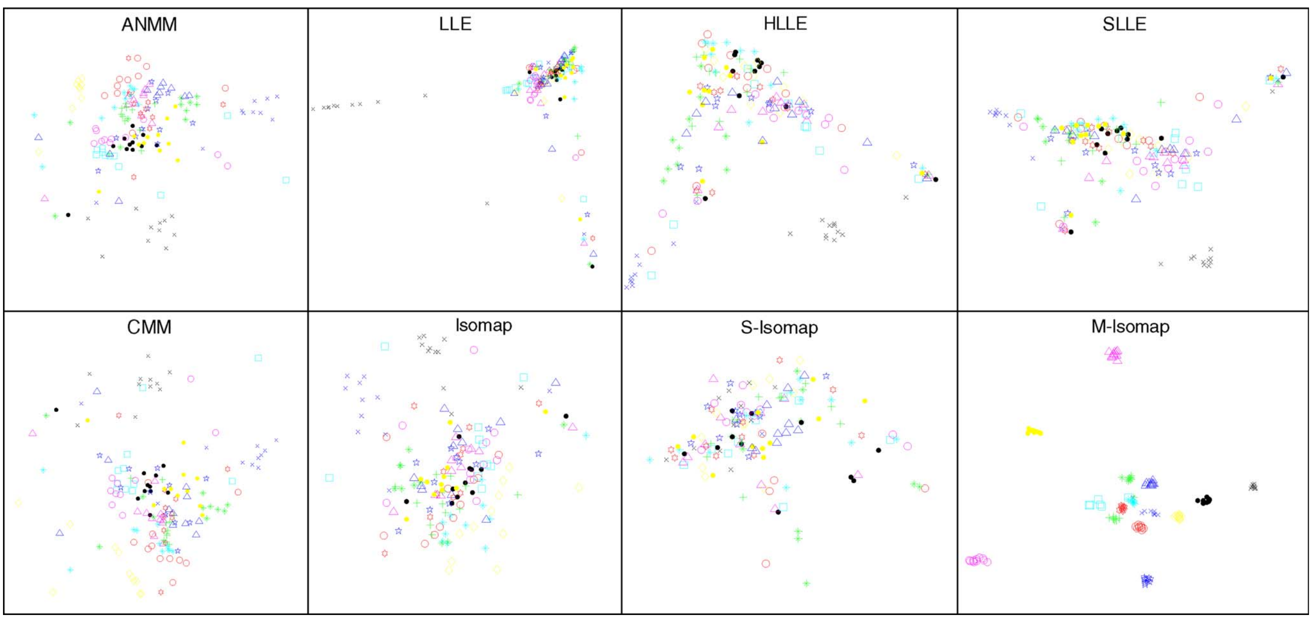


Fig. 10. Two-dimensional embedding result of the YALE face data set by each method.

some extent, however, these methods are unable to organize more separate embeddings of the face manifolds. Isomap, LLE, HLLE, and SLLE embed the faces of several persons well, but most faces of different persons are still congregated in their embedding spaces.

2) From the face visualization results, ANMM and our M-Isomap implicitly emphasize the natural clusters of the faces and exhibit separate clusters between dissimilar faces. They make similar face of the same individual lie in the vicinity of the face image space and make dissimilar faces from different individuals appear far away in their reduced embedding spaces.

To numerically evaluate the visualization results, we apply the same k-means clustering setting for evaluations. We show the averaged clustering results in Fig. 9. We can similarly observe the following.

1) For each method, the delivered normalized MI is higher than the clustering accuracy in each case.
2) For this data set, the results of the S-Isomap are the worst compared with the other methods in each case. It is interesting to find that Isomap performs slightly better than S-Isomap in this case. CMM, LLE, HLLE, and SLLE deliver comparable results, and they are better than S-Isomap and Isomap.
3) The numerical results of ANMM and M-Isomap keep consistent with the visual results in Fig. 8. Specifically, the clustering results obtained by ANMM and our M-Isomap are higher than those by the other methods in this data set.

E. Face Manifold Visualization on YALE Database

In this experiment, we address a face manifold visualization by using the YALE database. This database is collected at the Yale Center for Computational Vision and Control, which consists of 165 grayscale images in graphics interchange format of 15 individuals. The images demonstrate variations in lighting conditions (left light, center light, and right light), facial expressions (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Some typical face images under are shown in Fig. 3(c). In this simulation, we aim at visualizing images of all individuals. The k number in k-neighborhood is set to 15 for each NNS-type method. The 2-D embeddings are shown in Fig. 10. We have the following: 1) Due to the complex distributions and relations between features, ANMM, CMM, Isomap, S-Isomap, LLE, HLLE, and SLLE work poorly in interface separation, because they cannot embed the related low-dimensional face manifolds respectably, and 2) compared with the other methods, M-Isomap can construct the optimal face image subspace in which higher separation between

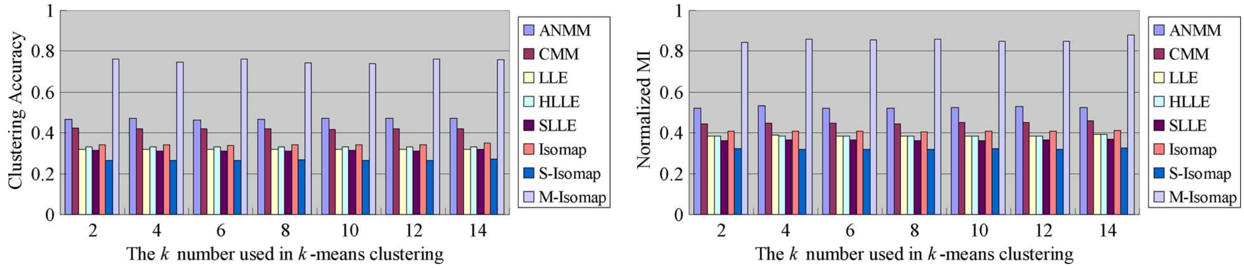


Fig. 11. Clustering evaluation result of each method on the YALE face data set.

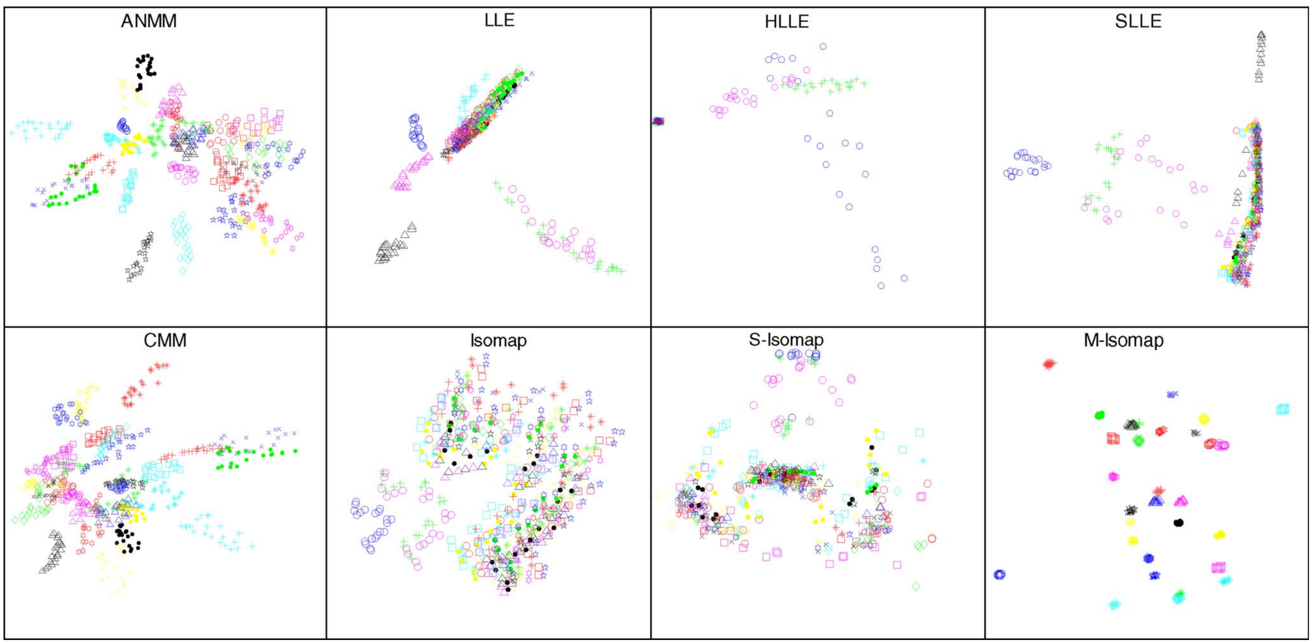


Fig. 12. Two-dimensional embedding result of the CMU PIE face data set by each method.

interface images and enhanced intraface compactness are produced at the same time.

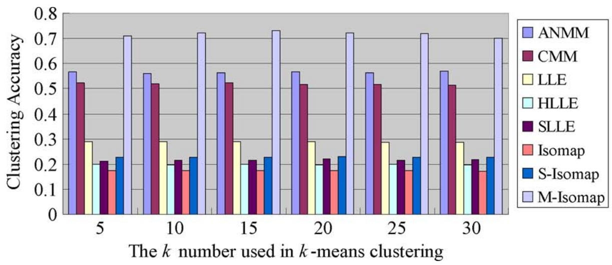
We show the averaged clustering accuracy and normalized MI computed by the k-means algorithm in Fig. 11. We observe from Fig. 11 the following.

1) The clustering performance of each method is also stable with the increasing k number in the k-means clustering. We once again experimentally observe that the normalized MI obtained by each method is higher than the corresponding clustering accuracy.
2) Because ANMM, CMM, Isomap, LLE, HLLE, S-Isomap, and SLLE cannot separate faces of different persons visually, their clustering results, which are comparable, are worse in this data set.
3) Recalling that M-Isomap embeds the faces appropriately, thus, it exhibits promising clustering accuracy and normalized MI in this case.

F. Face Visualization on CMU PIE Database

In this section, we test our method by visualizing the CMU PIE database which contains 68 individuals with 41 368 face images as a whole. The images were captured under varying poses, illuminations, and expressions. Some typical face images are shown in Fig. 3(d). In our simulation, we employ a sampled subset tested in [24] from the CMU PIE database. In this subset, the poses and expressions are fixed. Thus, finally, 21 images per person (a total of 1428 images for the 68 individuals) under different lighting conditions are sampled. In order to make the results clear, we only visualize the first 30 individuals. Before DR, we preprocess the data set through applying PCA to reduce the dimensionality of the data to 200. For each NNS-type method, the k value in k-neighborhood is set to 15. We show the 2-D embedding result of each method in Fig. 12. We see the following.

1) Due to the complex distributions of the data set, Isomap and S-Isomap perform particularly poor in revealing the intrinsic characteristics of the faces and separating the faces from different individuals. Through comparing with Isomap and S-Isomap, LLE, HLLE, and SLLE implicitly emphasize the natural clusters of some similar faces, but most dissimilar faces of different persons are still overlapped in their embedding spaces.
2) Clearly, the marginal ANMM and CMM algorithms and our M-Isomap algorithm work well in revealing the intrinsic structures of the data set and delivering enhanced interface separation. However, by comparing with ANMM and CMM, our M-Isomap approach is able to organize compact clusters of faces belonging to the same individual by delivering larger margins between interface clusters.



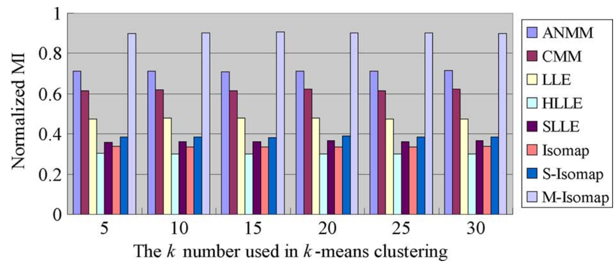


Fig. 13. Clustering evaluation result of each method on the CMU PIE face data set.

We apply the k-means clustering to the embedded data for evaluating the clustering performance. Similarly, we average the results over 200 initializations for each k number in the k-means clustering. The corresponding clustering results are shown in Fig. 13. Similar observations can be found as follows.

1) The normalized MI delivered by each method is higher than the corresponding clustering accuracy.
2) Our M-Isomap method achieves the highest clustering accuracy and normalized MI in each case. The ANMM and CMM algorithms deliver the highest clustering results among the remaining methods. The results of the LLE, Isomap, HLLE, S-Isomap, and SLLE algorithms are comparative with each other in virtually all the cases.

V. Concluding Remarks

In this paper, we have discussed the pairwise-constrained discriminant nonlinear DR problem. By incorporating the pairwise-constrained neighborhood graphs into the Isomap framework, an effective marginal Isomap algorithm named M-Isomap has been proposed for nonlinear DR and data visualization. M-Isomap uses the PCs derived from the given data to guide the discriminant manifold learning. As a result, M-Isomap will be flexible in regulating supervised information. In extracting the informative features, M-Isomap aims at separating interclass neighboring points in addition to preserving the neighborhood relations of intraclass points. Thus, large margins between inter- and intraclass clusters are delivered. To compute the embeddings, a TR optimization approach has been employed delivering more specific solutions and orthogonal basis vectors.

The validity of M-Isomap has been examined by a synthetic, a UCI, and three real data sets. From all investigated cases, the M-Isomap algorithm is capable of producing clear separation on the manifold embedding of multiple classes or objects. Because of the stronger constraints brought by the PCs, the margins of both inter- and intraclass clusters are significantly enlarged in the embedding space of M-Isomap. These margins are significantly larger than those produced by the discriminant ANMM, CMM, SLLE, and S-Isomap and the unsupervised LLE, HLLE, and Isomap. The clustering evaluation has also verified the efficiency of M-Isomap. We also observe from our simulations that SLLE and S-Isomap cannot embed the real data sets of multiple classes appropriately. Note that, for NNS-type methods, the selection of the k number in k-neighborhood is an open problem. Therefore, investigating the optimal determination of k for manifold learning is required. Exploring selecting an optimal constraint subset and extending M-Isomap to the semisupervised settings are also worth studying.

References

[1] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[2] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[3] F. Wang and C. S. Zhang, "Feature extraction by maximizing the average neighborhood margin," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Minneapolis, MN, 2007, pp. 1–8.

[4] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

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

[6] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. New York: Springer-Verlag, 2007.

[7] 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, Dec. 2000.

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

[9] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[10] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.

[11] X. Geng, D. Zhan, and Z. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1098–1107, Dec. 2005.

[12] D. L. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," *Proc. Nat. Acad. Sci. USA*, pp. 5591–5596, 2003.

[13] D. Sun and D. Q. Zhang, "Bagging constraint score for feature selection with pairwise constraints," *Pattern Recognit.*, vol. 43, no. 6, pp. 2106–2118, Jun. 2010.

[14] F. Wang, "Semi-supervised metric learning by maximizing constraint margin," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 931–939, Aug. 2011.

[15] F. Wang, S. C. Chen, T. Li, and C. S. Zhang, "Semi-supervised metric learning by maximizing constraint margin," in *Proc. ACM Conf. Inf. Knowl. Manag.*, Napa Valley, CA, 2008, pp. 1457–1458.

[16] R. Rivest, T. Cormen, C. Leiserson, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

[17] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing*. Redwood City, CA: Benjamin Cummings, 1994.

[18] Y. Guo, S. Li, J. Yang, T. Shu, and L. Wu, "A generalized Foley–Sammon transform based on generalized Fisher discriminant criterion and its application to face recognition," *Pattern Recognit. Lett.*, vol. 24, no. 1–3, pp. 147–158, Jan. 2003.

[19] Y. Jia, F. Nie, and C. Zhang, "Trace ratio problem revisited," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 729–735, Apr. 2009.

[20] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, 2007, pp. 1–8.

[21] D. Ridder, O. Kouropteva, O. Okun, M. Pietikäinen, and R. P. W. Duin, "Supervised locally linear embedding," in *Proc. Int. Conf. Artif. Neural Netw.*, Istanbul, Turkey, 2003, pp. 333–341.
[22] D. Cai, X. He, and J. W. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[23] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.
[24] D. Cai, X. He, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2005.

Zhao Zhang (S'11) received the B.Eng. (First Hons.) and M.Sc. degrees from the Department of Computer Science and Technology, Nanjing Forestry University, Nanjing, China, in 2008 and 2010, respectively. He is currently working toward the Ph.D. degree in the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong.

He was a Visiting Research Engineer with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore, in 2012. His current research interests include machine learning, pattern recognition, and applications.

Tommy W. S. Chow (M'93–SM'03) received the B.Sc. (First Hons.) and Ph.D. degrees from the University of Sunderland, Sunderland, U.K.

He is currently a Professor with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. He has authored or coauthored over 140 technical papers in international journals, five book chapters, and over 60 technical papers in conference proceedings. He is serving as an Associate Editor of *Pattern Analysis and Applications* and the *International Journal of Information Technology*.

Mingbo Zhao (S'11) received the B.Sc. and M.Sc. degrees from the Department of Electronic Engineering, Shanxi University, Taiyuan, China, in 2005 and 2008, respectively. He is currently working toward the Ph.D. degree in the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong.

His current interests include machine learning, data mining, pattern recognition, and applications.