Pattern Recognition 43 (2010) 2982–2992

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Non-linear metric learning using pairwise similarity and dissimilarity constraints and the geometrical structure of data

Mahdieh Soleymani Baghshah [a,*], Saeed Bagheri Shouraki [b]

[a] Computer Engineering Department, Sharif University of Technology (SUT), Azadi St., PO Box: 1458889694, Tehran, Iran
[b] Electrical Engineering Department, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 18 December 2008
Received in revised form
18 January 2010
Accepted 26 February 2010

Keywords:
Metric learning
Positive and negative constraints
Semi-supervised clustering
Optimization problem
Non-linear
Topological structure
Kernel

ABSTRACT

The problem of clustering with side information has received much recent attention and metric learning has been considered as a powerful approach to this problem. Until now, various metric learning methods have been proposed for semi-supervised clustering. Although some of the existing methods can use both positive (must-link) and negative (cannot-link) constraints, they are usually limited to learning a linear transformation (i.e., finding a global Mahalanobis metric). In this paper, we propose a framework for learning linear and non-linear transformations efficiently. We use both positive and negative constraints and also the intrinsic topological structure of data. We formulate our metric learning method as an appropriate optimization problem and find the global optimum of this problem. The proposed non-linear method can be considered as an efficient kernel learning method that yields an explicit non-linear transformation and thus shows out-of-sample generalization ability. Experimental results on synthetic and real-world data sets show the effectiveness of our metric learning method for semi-supervised clustering tasks.

1. Introduction

Distance metrics are a key issue in many machine learning algorithms [1]. Over the past few years, there has been considerable research on distance metric learning [2]. Many of the earlier studies optimize the metric with class labels for classification tasks [3–8]. More recently, researchers have given much attention to distance learning for semi-supervised clustering tasks. As class label information is not generally available for clustering tasks, constraints are used as more natural supervisory information for these tasks. Pairwise similarity (positive) and dissimilarity (negative) constraints are the most popular kind of side information that has been used for semi-supervised clustering. However, other kinds of side information like relative comparisons have also been considered in some studies.

Over the last few years, the problem of clustering with side information (semi-supervised clustering) has received increasing attention [9,10] and distance learning has been considered as a powerful approach for this problem. The two most frequently used approaches that include side information in the clustering algorithms are *constraint-based* [11–21] and *distance function learning* [22–34] approaches [24]. In the former approach, the clustering algorithm itself is modified to use the available labels or constraints to bias the search for an appropriate data clustering. However, in the latter approach, the algorithm learns a distance function prior to clustering. The learned distance function tries to put similar points close together and dissimilar points far away from each other. This approach is more flexible in the choice of distance function [33]. Additionally, it has received considerable attention in recent studies [1,25,28–31,33,34] and we also use this approach.

Distance learning based on constraints has been studied by many researchers [22–34]. Klein et al. [22] introduced a metric adaptation method for semi-supervised clustering. This method finds a distance measure according to the shortest path in a version of the similarity graph that has been altered by positive constraints. However, negative constraints have been employed after the metric adaptation phase during the complete-link clustering. Some latter studies [1,23,25,28,34] have considered a more popular approach that learns a global Mahalanobis metric from pairwise constraints. Xing et al. [23] proposed a convex optimization problem to learn a global Mahalanobis metric according to pairwise constraints. Bar-Hillel et al. [25] devised a more efficient, non-iterative algorithm called *relevant component analysis* (RCA) for learning a Mahalanobis metric. This method can only incorporate positive constraints. An extension of the RCA method that can consider both positive and negative constraints has also been introduced by Yeung and Chang [28].

* Corresponding author. Tel.: +98 21 6616 4642; fax: +98 21 6601 9246.
E-mail addresses: soleyman@ce.sharif.edu, mahdiesoleymani@yahoo.com (M. Soleymani Baghshah), bagheri-s@sharif.edu (S. Bagheri Shouraki).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.02.022

More recently, some non-linear metric learning methods for semi-supervised clustering have been introduced. Chang and Yeung [29] proposed a locally linear metric learning method that considers only positive constraints. The objective function of this method has many local optima and the topology cannot be preserved well during this approach [30]. Chang and Yeung [31] proposed also a metric adaptation method. This method adjusts the location of data points iteratively, so that similar points tend to get closer and dissimilar points tend to move away from each other. As this method lacks an explicit transformation map, it cannot project new data points onto the transformed space straightforwardly [31]. Additionally, the movement of data points in this method may interfuse the structure of the data. In [30], two kernel-based metric learning methods have been presented that do have some limitations [30]. These kernel-based methods can use only positive constraints.

Among the existing metric learning methods, some of them [1,23,28,34,39,40] can incorporate both positive and negative constraints. However, most of these methods [1,23,28,34] learn only a linear transformation that corresponds to a Mahalanobis metric. Although some recent studies [39,40] have been introduced for kernel learning from positive and negative constraints, they are based on learning non-parametric kernel matrices. These methods can only find distances of the seen data. Additionally, the optimization problems in these methods are usually difficult to solve [40] and the degree of freedom of the corresponding models is very high (i.e., n^2 where n denotes the number of data points). In this paper, we propose an efficient non-linear metric learning method that considers both positive and negative constraints and also the topological structure of the data. We formulate the proposed method as a constrained trace ratio optimization problem that can be solved efficiently using algorithms introduced for this purpose (e.g., Xiang et al.'s method [1]). The proposed non-linear method can be considered as an efficient kernel learning method that does not need to learn all items of an $n \times n$ matrix. Our method yields an explicit transformation that can project new data points onto the transformed space.

The rest of this paper is organized as follows: Section 2 presents a brief review of related works. In Section 3, first the general form of the proposed optimization problems that incorporate both positive and negative constraints and also the topological structure of the data is introduced. Then, we present special problems that can be solved efficiently for learning linear and non-linear transformations. Finally, we present a kernel-based method and show the relation between the proposed non-linear method and a special form of this kernel-based method. Section 4 presents some experimental results on synthetic and real-world data sets. Concluding remarks are given in the last section.

2. Related works

In this section, we review those methods that can consider both positive and negative constraints to learn a transformation. A positive constraint denotes a pair of data points that must be in the same cluster while a negative constraint denotes two data points that must be in two different clusters [1]. Most of the existing methods that can use both positive and negative constraints learn a Mahalanobis metric $\mathbf{A}$ (where $\mathbf{A}$ is a positive semi-definite matrix) or, equivalently, find a transformation matrix $\mathbf{W}$ ($\mathbf{y}=\mathbf{W}^T\mathbf{x}$). Learning the transformation matrix $\mathbf{W}$ can yield the Mahalanobis metric $\mathbf{A}=\mathbf{W}\mathbf{W}^T$ according to:

$$\begin{aligned}\|\mathbf{y}_i-\mathbf{y}_j\|^2 &= (\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{W}\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\\ &= (\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{A}(\mathbf{x}_i-\mathbf{x}_j)=\|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2.\end{aligned} \tag{1}$$

Xing et al. introduced the first metric learning method using both positive and negative constraints [23]. They presented the following objective function:

$$\begin{aligned}&\min_{\mathbf{A}} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2,\\ &\text{s.t.} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D} \|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2 \geq 1,\\ &\mathbf{A}\geq\mathbf{0},\end{aligned} \tag{2}$$

where P is the set of positive constraints and D is the set of negative constraints. Xing et al. [23] used the gradient descent and the idea of iterative projection to solve this problem. Although the above optimization problem is convex, it is a hard problem to solve and the introduced solution in [23] is slow and somewhat unstable [25].

Chang et al. [28] introduced an extended version of the RCA [25] method. They proposed the transformation matrix $\mathbf{W}=(\mathbf{S}_b)^{1/2}(\mathbf{S}_w)^{1/2}$ where $\mathbf{S}_b$ denotes the inter-class (cluster) covariance matrix computed from negative constraints and $\mathbf{S}_w$ shows the intra-class (cluster) covariance matrix computed from positive constraints. Although this transformation can be found easily, the singularity problem may occur during the calculation of $\mathbf{S}_w^{-1/2}$. Additionally, it has not been obtained as a solution of an optimization problem.

Hoi et al. [34] proposed the *discriminative component analysis* (DCA) method using the ratio of determinants as the objective function:

$$\mathbf{W}^*=\arg\max_{\mathbf{W}}\frac{|\mathbf{W}^T\widehat{\mathbf{C}}_b\mathbf{W}|}{|\mathbf{W}^T\widehat{\mathbf{C}}_w\mathbf{W}|}, \tag{3}$$

where $\widehat{\mathbf{C}}_b$ shows the covariance between data of discriminative chunklets (cannot-links) and $\widehat{\mathbf{C}}_w$ shows the total covariance of data within the same chunklet (must-links) [34]. This problem can be solved analytically by the eigenvalue decomposition of $\widehat{\mathbf{C}}_w^{-1}\widehat{\mathbf{C}}_b$ [1]. However, the singularity problem may occur during the calculation of $\widehat{\mathbf{C}}_w^{-1}$ [34]. To avoid the singularity problem, DCA diagonalizes the covariance matrices $\widehat{\mathbf{C}}_b$ and $\widehat{\mathbf{C}}_w$ simultaneously and discards the eigenvectors corresponding to the zero eigenvalue [1].

Recently, Xiang et al. [1] introduced the trace ratio objective function (with the constraint $\mathbf{W}^T\mathbf{W}=\mathbf{I}$) as a more appropriate objective function:

$$\mathbf{W}^*=\arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}}\frac{tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})}, \tag{4}$$

where $\mathbf{S}_w$ is the covariance matrix computed from positive constraints and $\mathbf{S}_b$ is the covariance matrix obtained from negative constraints. The constraint $\mathbf{W}^T\mathbf{W}=\mathbf{I}$ has been introduced to avoid degenerate solutions [1]. The optimization problem in (4) is similar to the problem introduced by Guo et al. [35] as the *generalized Foley–Sammon transform* (GFST). It seeks a transformation matrix in the global sense instead of learning individual transformation vectors for different dimensions like Fisher criterion. To solve the above optimization problem, Xiang et al. [1] have developed an iterative method exploring the optimum in way of binary search. Additionally, they have found a lower bound and an upper bound including the optimum to speed up the search. Their proposed method provides a heuristic search to solve the problem presented in (4) [1]. In this paper, we propose a generalized form of the objective function presented in (4) that can learn a non-linear transformation and also considers the topological structure of data.

3. Proposed approach

In this section, we first propose a general framework for learning an appropriate transformation from positive and negative constraints. Based on this framework, we propose problems (that can be solved efficiently) for learning linear and non-linear transformations. Finally, we introduced our kernel-based method and show the relation between a special form of this method and the proposed non-linear metric learning method.

Here, we introduce some notations used in this section. $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ represents the set of n data points. $P=\{(\mathbf{x}_i,\mathbf{x}_j)|\mathbf{x}_i$ and $\mathbf{x}_j$ must be in the same class$\}$ shows the set of positive constraints and $D=\{(\mathbf{x}_i,\mathbf{x}_j)|\mathbf{x}_i$ and $\mathbf{x}_j$ must be in two different classes$\}$ denotes the set of negative constraints. $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n]$ represents the matrix of data points.

3.1. General form of the proposed optimization problems

In this section, we introduce a general framework for the proposed optimization problems that incorporate both positive and negative constraints and also the manifold structure of data to learn a transformation. We propose the following optimization problem for learning the transformation $\mathbf{y}=f(\mathbf{x})$:

$$f^*=\arg\max_f \frac{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D}\|f(\mathbf{x}_i)-f(\mathbf{x}_j)\|^2}{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P}\|f(\mathbf{x}_i)-f(\mathbf{x}_j)\|^2+\alpha J_{LLE}(f,\mathbf{X})}. \tag{5}$$

Here, $J_{LLE}(f,\mathbf{X})$ shows a regularizer (penalty) term that tries to preserve the topological structure of the data in the transformed space and $\alpha\geq 0$ balances between distances of similar pairs and the regularizer term. In Problem (5), a transformation is sought that makes distances of point pairs in D as large as possible while making a combination of distances between point pairs in P and the regularizer term as small as possible. In this problem, we have used all the data points along with the pairwise constraints via the term $J_{LLE}(f,\mathbf{X})$.

To preserve the topological structure of data through the term $J_{LLE}(f,\mathbf{X})$ in the objective function, we use the idea of *locally linear embedding* (LLE) [37] method. Indeed, we try to preserve the geometrical structure of data by retaining locally linear relationships between close data points. Given the set of data points, a k-nearest neighbor graph can be used to model the relation between close data points. The optimal weight matrix $\mathbf{S}^*=[s_{ij}^*]$ providing minimal error for linear reconstruction of data points from their neighbors is obtained according to the following problem [37]:

$$\mathbf{S}^*=\min_{\mathbf{S}=[s_{ij}]}\sum_{i=1}^{n}\Big\|\mathbf{x}_i-\sum_{\mathbf{x}_j\in N_k(\mathbf{x}_i)}s_{ij}\mathbf{x}_j\Big\|^2,\quad \text{s.t.}\quad \forall i,\ \sum_{x_j\in N_k(x_i)}s_{ij}=1, \tag{6}$$

where $N_k(\mathbf{x}_i)$ shows the set of k nearest neighbors of $\mathbf{x}_i$. This problem can be solved as a constrained least-squares problem [37]. After finding the optimal weight matrix $\mathbf{S}^*$, we can define the penalty term $J_{LLE}(f,\mathbf{X})$ as:

$$\begin{aligned}J_{LLE}(f,\mathbf{X})&\equiv\sum_{i=1}^{n}\Big\|f(\mathbf{x}_i)-\sum_{\mathbf{x}_j\in N_k(\mathbf{x}_i)}s_{ij}^*f(\mathbf{x}_j)\Big\|^2=tr(\mathbf{Y}(\mathbf{I}-\mathbf{S}^*)^T(\mathbf{I}-\mathbf{S}^*)\mathbf{Y}^T)\\&=tr(\mathbf{YEY}^T),\end{aligned} \tag{7}$$

where $\mathbf{E}=(\mathbf{I}-\mathbf{S}^*)^T(\mathbf{I}-\mathbf{S}^*)$ and $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_n]$. In (7), $J_{LLE}(f,\mathbf{X})$ denotes the locally linear reconstruction error of the transformed data points from their transformed neighbors according to the weight matrix $\mathbf{S}^*$. Indeed, it reflects the inconsistency between the transformed data and the matrix $\mathbf{S}^*$ that represents the geometrical structure of data in the input space.

We can rewrite the problem in (5) (using (7)) as:

$$f^*=\arg\max_{\mathbf{y}=f(\mathbf{x})}\frac{tr(\mathbf{YU}_D\mathbf{Y}^T)}{tr(\mathbf{YU}_P\mathbf{Y}^T)+\alpha tr(\mathbf{YEY}^T)}, \tag{8}$$

where

$$\mathbf{U}_P=\sum_{(i,j)\in P}(\mathbf{e}_i-\mathbf{e}_j)(\mathbf{e}_i-\mathbf{e}_j)^T, \tag{9}$$

$$\mathbf{U}_D=\sum_{(i,j)\in D}(\mathbf{e}_i-\mathbf{e}_j)(\mathbf{e}_i-\mathbf{e}_j)^T, \tag{10}$$

and $\mathbf{e}_i$ shows the i-th column of the $n\times n$ identity matrix. In the next sections, we propose special forms of the problem in (8) that can be solved efficiently.

3.2. Linear metric learning method

In this section, we introduce a special form of the problem in (8) that seeks a linear transformation $f(\mathbf{x})=\mathbf{W}^T\mathbf{x}$. For learning the transformation matrix $\mathbf{W}$, we substitute $\mathbf{Y}=\mathbf{W}^T\mathbf{X}$ in (8) and add the orthogonal constraint $\mathbf{W}^T\mathbf{W}=\mathbf{I}$ to prevent improper solutions [1,34–36]. Thus, the following optimization problem is obtained:

$$\mathbf{W}^*=\arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}}\frac{tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{tr(\mathbf{W}^T(\mathbf{S}_w+\alpha\mathbf{XEX}^T)\mathbf{W})}. \tag{11}$$

Here, $\mathbf{S}_w$ and $\mathbf{S}_b$ are defined as:

$$\mathbf{S}_w=\mathbf{XU}_P\mathbf{X}^T=\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^T. \tag{12}$$

$$\mathbf{S}_b=\mathbf{XU}_D\mathbf{X}^T=\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}(\mathbf{x}_k-\mathbf{x}_l)(\mathbf{x}_k-\mathbf{x}_l)^T. \tag{13}$$

If we set $\alpha=0$ in (11), the problem becomes similar to the optimization problem in (4), which has been introduced by Xiang et al. [1]. In (11), $\mathbf{W}\in R^{d\times d'}(d'\leq d)$ shows the transformation matrix where d and d' denote the dimensionality of the input and the transformed space respectively. When $d=d'$, we have $\mathbf{WW}^T=\mathbf{W}^T\mathbf{W}=\mathbf{I}$ which generates the Euclidean metric [1] and thus we consider $d'<d$ in our method. The optimization problem in (11) is equal to the following problem seeking a metric $\mathbf{A}=\mathbf{WW}^T$:

$$\mathbf{A}^*=\arg\max_{\substack{\mathbf{A}=\mathbf{WW}^T\\ \mathbf{W}^T\mathbf{W}=\mathbf{I}}}\frac{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D}\|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2}{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P}\|\mathbf{x}_i-\mathbf{x}_j\|_{\mathbf{A}}^2+\alpha tr(\mathbf{AXEX}^T)}. \tag{14}$$

The proposed problem in (11) cannot be solved by eigenvalue decomposition approaches. However, we can use the search algorithm introduced by Xiang et al. [1] (for optimizing the constrained trace ratio problem) to solve this problem. Since the matrix $\mathbf{E}$ and consequently $\mathbf{S}_w+\alpha\mathbf{XEX}^T$ are symmetric positive semi-definite matrices, the necessary condition of the introduced algorithm in [1] is satisfied. Thus, we can use the algorithm presented in Table 1 to solve the proposed optimization problem. Table 2 shows the steps of the proposed linear method. In the first step of this method, the matrix $\mathbf{E}$ that models the geometrical structure of the data (according to the locally linear relationship between close data points) is found. In the remaining steps, the proposed problem in (11) is obtained and solved. In Table 1, the introduced algorithm by Xiang et al. [1] for solving the constrained trace ratio problem $\mathbf{W}^*=\arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}}[tr(\mathbf{W}^T\mathbf{S}_1\mathbf{W})/tr(\mathbf{W}^T\mathbf{S}_2\mathbf{W})]$ has been shown. This algorithm considers two cases [1]:

- $d'\leq d-r$: If $\mathbf{W}$ is in the null space of $\mathbf{S}_2$, then $tr(\mathbf{W}^T\mathbf{S}_2\mathbf{W})=0$. Therefore, the numerator $tr(\mathbf{W}^T\mathbf{S}_1\mathbf{W})$ can be maximized after performing a null-space transformation $\mathbf{y}=\mathbf{Z}^T\mathbf{x}$ where

Table 1
Algorithm for solving the trace ratio optimization problem [1].

Input: $\mathbf{S}_1,\mathbf{S}_2 \in R^{d\times d}$, the lower dimensionality d', an error constant ε.
Output: A matrix $\mathbf{W}^* \in R^{d\times d'}$.
1. Calculate the rank r of the matrix $\mathbf{S}_2$.
2. if $d' > d-r$
   2.1. Find $\alpha_1,\ldots,\alpha_{d'}$ as the first d' largest eigenvalues of $\mathbf{S}_1$ and $\beta_1,\ldots,\beta_{d'}$ as the first d' smallest eigenvalues of $\mathbf{S}_2$.
   2.2. $\lambda_1 \leftarrow tr(\mathbf{S}_1)/tr(\mathbf{S}_2)$, $\lambda_2 \leftarrow \sum_{i=1}^{d'}\alpha_i/\sum_{i=1}^{d'}\beta_i$, $\lambda \leftarrow (\lambda_1+\lambda_2)/2$.
   2.3. while $\lambda_1-\lambda_2 > \varepsilon$ do
      2.3.1. Calculate $g(\lambda)$ as sum of the first d' largest eigenvalues of $\mathbf{S}_1-\lambda\mathbf{S}_2$.
      2.3.2. if $g(\lambda)>0$ then $\lambda_1 \leftarrow \lambda$ else $\lambda_2 \leftarrow \lambda$.
      2.3.3. $\lambda \leftarrow (\lambda_1+\lambda_2)/2$
   2.4. $\mathbf{W}^* = [\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_{d'}]$ where $\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_{d'}$ are the d' eigenvectors corresponding to the d' largest eigenvalues of $\mathbf{S}_1-\lambda\mathbf{S}_2$.
3. else
   3.1. $\mathbf{W}^* = \mathbf{Z}[\mathbf{v}_1,\ldots,\mathbf{v}_{d'}]$. Here $\mathbf{v}_1,\ldots,\mathbf{v}_{d'}$ are d' eigenvectors corresponding to the d' largest eigenvalues of $\mathbf{Z}^T\mathbf{S}_1\mathbf{Z}$ and $\mathbf{Z}=[\mathbf{Z}_1\ldots,\mathbf{Z}_{d-r}]$ are the eigenvectors corresponding to $d-r$ zero eigenvalues of $\mathbf{S}_2$.

Table 2
Our linear metric learning algorithm.

1 Find the weight matrix $\mathbf{S}^*$ as the solution to the constrained least-squares optimization problem presented in (6). $\mathbf{E}=(\mathbf{I}-\mathbf{S}^*)^T(\mathbf{I}-\mathbf{S}^*)$.
2 Find $\mathbf{S}_w$ according to (12) and $\mathbf{S}_b$ according to (13).
3 Obtain the new matrix $\mathbf{S}=\mathbf{S}_w+\alpha\mathbf{X}\mathbf{E}\mathbf{X}^T$.
4 Learn $\mathbf{W}^*$ according to the algorithm presented in Table 1 (Input:$\mathbf{S}_b$, $\widehat{\mathbf{S}}$, d', and ε).
5 Output the transformation matrix $\mathbf{W}^*$ or the metric $\mathbf{A}^*=\mathbf{W}^*(\mathbf{W}^*)^T$.

$\mathbf{Z}\in\mathbb{R}^{d\times(d-r)}$ is a matrix whose columns are the eigen-vectors corresponding to $d-r$ zero eigenvalues of $\mathbf{S}_2$. Thus, we find $\mathbf{W}^*=\mathbf{Z}\mathbf{V}^*$ where $\mathbf{V}^*=\arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}}[tr(\mathbf{V}^T(\mathbf{Z}^T\mathbf{S}_1\mathbf{Z})\mathbf{V})]$.

- When $d' > d-r$, a binary search (giving a lower bound and an upper bound) is used to find $\lambda^*=\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}}[tr(\mathbf{W}^T\mathbf{S}_1\mathbf{W})/tr(\mathbf{W}^T\mathbf{S}_2\mathbf{W})]$. Indeed, a function $g(\lambda)=\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} tr(\mathbf{W}^T(\mathbf{S}_1-\lambda\mathbf{S}_2)\mathbf{W})$ is introduced and a λ is sought such that $g(\lambda)=0$. The value of $g(\lambda)$ can be easily calculated as the sum of the first d' largest eigenvalues of $\mathbf{S}_1-\lambda\mathbf{S}_2$. The optimal $\mathbf{W}^*$ is finally obtained by performing the eigenvalue decomposition of $\mathbf{S}_1-\lambda^*\mathbf{S}_2$.

3.3. Non-linear metric learning method

In this section, we extend the metric learning method that we presented in the previous section to a non-linear method. Let $X'=\{\mathbf{x}_{l_1},\mathbf{x}_{l_2},\ldots,\mathbf{x}_{l_m}\}$ be the set of data points appearing in positive constraints (i.e., $\forall 1\le i\le m,\exists s|(\mathbf{x}_{l_i},\mathbf{x}_s)\in P\vee(\mathbf{x}_s,\mathbf{x}_{l_i})\in P$) and m be the number of unique data points involved in positive constraints. For each data point $\mathbf{x}_{l_r}\in X'$, a d' -dimensional vector $\mathbf{v}_r$ is considered. The data points appearing in X' are used as the most informative data points. Using an idea from [29], we define a non-linear transformation that is formed by effect of the data points appearing in X' on all data points via a neighborhood function. Every data point $\mathbf{x}_i$ $(1\le i\le n)$ is transformed to:

$$\mathbf{y}_i=\sum_{r=1}^{m}\pi_{l_r,i}\mathbf{v}_r, \tag{15}$$

where $\pi_{k,i}$ can be computed as:

$$\pi_{k,i}=\exp(-\|\mathbf{x}_i-\mathbf{x}_k\|/w). \tag{16}$$

Here, w shows the window parameter of the exponential neighborhood function. Note that (15) can be expressed as $\mathbf{y}_i=\mathbf{V}^T\boldsymbol{\pi}_i$ where $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_m]^T$ is an $m\times d'$ matrix and $\boldsymbol{\pi}_i=(\pi_{l_1,i},\pi_{l_2,i},\ldots,\pi_{l_m,i})^T$ is an m-dimensional vector. Thus, we have

$$\mathbf{Y}=\mathbf{V}^T\boldsymbol{\Pi}, \tag{17}$$

where $\boldsymbol{\Pi}=[\boldsymbol{\pi}_1,\boldsymbol{\pi}_2,\ldots,\boldsymbol{\pi}_n]$ is the $m\times n$ neighborhood matrix and $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_n]$ shows the transformed data points. Therefore, finding the optimal transformation matrix $\mathbf{V}^*$ and applying it on the neighborhood matrix $\boldsymbol{\Pi}$ yields a non-linear transformation. Using the general optimization problem in (8) and setting $\mathbf{Y}=\mathbf{V}^T\boldsymbol{\Pi}$, the optimal matrix can be obtained through the following optimization problem (the constraint $\mathbf{V}^T\mathbf{V}=\mathbf{I}$ has been added to prevent improper solutions):

$$\mathbf{V}^*=\arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}}\frac{tr(\mathbf{V}^T\mathbf{S}_b^{\pi}\mathbf{V})}{tr(\mathbf{V}^T(\mathbf{S}_w^{\pi}+\alpha\boldsymbol{\Pi}\mathbf{E}\boldsymbol{\Pi}^T)\mathbf{V})}, \tag{18}$$

where

$$\mathbf{S}_w^{\pi}=\boldsymbol{\Pi}\mathbf{U}_P\boldsymbol{\Pi}^T=\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P}(\boldsymbol{\pi}_i-\boldsymbol{\pi}_j)(\boldsymbol{\pi}_i-\boldsymbol{\pi}_j)^T, \tag{19}$$

$$\mathbf{S}_b^{\pi}=\boldsymbol{\Pi}\mathbf{U}_D\boldsymbol{\Pi}^T=\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D}(\boldsymbol{\pi}_i-\boldsymbol{\pi}_j)(\boldsymbol{\pi}_i-\boldsymbol{\pi}_j)^T. \tag{20}$$

A similar algorithm to the one used in the previous section can solve the optimization problem in (18). Table 3 shows the steps required for finding the non-linear transformation. In the first step of this method, the matrix $\mathbf{E}$ that models the geometrical structure of the data is found. In the second step, the neighborhood matrix $\boldsymbol{\Pi}$ is calculated. In steps (3)–(5), the proposed problem in (18) is obtained and solved (using the introduced algorithm in Table 1).

3.4. Kernel-based method and its relation to the non-linear method

In this section, we introduce a kernel-based method and show the relation between the nonlinear method proposed in Section 3.2 and a special form of the kernel-based method. To perform our linear method in *reproducing Kernel Hilbert space* (RKHS), we consider the problem in a feature space F induced by a nonlinear mapping $\phi: R^d\to F$ [38]. For a proper chosen ϕ, we can define an inner product on F using Mercer kernel:

$$\langle\phi(\mathbf{x}),\phi(\mathbf{y})\rangle=k(\mathbf{x},\mathbf{y}), \tag{21}$$

where $k(.,.)$ is a positive semi-definite kernel function.

In the kernel-based method, we apply the transformation matrix $\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_{d'}]$ consisting of orthonormal vectors $\{\mathbf{w}_i\in F|i=1,2,\ldots,d'\}(<\mathbf{w}_i,\mathbf{w}_j>=\delta_{i,j})$ on $\phi(\mathbf{x})\in F$:

$$\mathbf{y}=\mathbf{W}^T\phi(\mathbf{x}). \tag{22}$$

This transformation performs a mapping from R^d to $R^{d'}$ (d and d' show the dimensionality of the input and the transformed space

Table 3
Out non-linear metric learning algorithm.

1 Find the weight matrix $\mathbf{S}^*$ as the solution of the constrained least-squares problem presented in (6). $\mathbf{E}=(\mathbf{I}-\mathbf{S}^*)^T(\mathbf{I}-\mathbf{S}^*)$.
2 Find the neighborhood matrix $\boldsymbol{\Pi}=[\boldsymbol{\pi}_1,\boldsymbol{\pi}_2,\ldots,\boldsymbol{\pi}_n]$ where $\boldsymbol{\pi}_i=(\pi_{l_1,i},\pi_{l_2,i},\ldots,\pi_{l_m,i})^T$ shows the value of neighborhood functions of the m unique data points involved in positive constraints $\pi_{r,i}=\exp(-\|\mathbf{x}_i-\mathbf{x}_r\|/w)$ in the location of $\mathbf{x}_i$.
3 Find $\mathbf{S}_w^{\pi}$ according to (19) and $\mathbf{S}_b^{\pi}$ according to (20).
4 Obtain the matrix $\widehat{\mathbf{S}}^{\pi}=\mathbf{S}_w^{\pi}+\alpha\boldsymbol{\Pi}\mathbf{E}\boldsymbol{\Pi}^T$.
5 Learn $\mathbf{V}^*$ according to the algorithm in Table 1 (Input: $\mathbf{S}_b^{\pi}$, $\widehat{\mathbf{S}}^{\pi}$, d', and ε).
6 Output the transformation matrix $\mathbf{V}^*$ or the metric $\mathbf{A}^*=\mathbf{V}^*(\mathbf{V}^*)^T$.

respectively). Let $\mathbf{\Phi}=[\phi(\mathbf{x}_1),\phi(\mathbf{x}_2),\ldots,\phi(\mathbf{x}_n)]$ denotes the data points in the kernel space. Based on (8) and (22), we obtain the following optimization problem in RKHS:

$$\mathbf{W}^*=\arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}}\frac{tr(\mathbf{W}^T\mathbf{\Phi}\mathbf{U}_D\mathbf{\Phi}^T\mathbf{W})}{tr(\mathbf{W}^T\mathbf{\Phi}(\mathbf{U}_P+\alpha\mathbf{E})\mathbf{\Phi}^T\mathbf{W})} \tag{23}$$

We added the constraint $\mathbf{W}^T\mathbf{W}=\mathbf{I}$ in the above problem to reflect that $\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_{d'}$ have been considered as orthonormal vectors. Since the vectors $\{\mathbf{w}_i\in F|i=1,2,...,d'\}$ can be written as linear combinations of data points in the kernel space $\phi(\mathbf{x}_1),\phi(\mathbf{x}_2),\ldots,\phi(\mathbf{x}_n)$,[1] there exist a matrix $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_{d'}]$ such that:

$$\mathbf{W}=\mathbf{\Phi}\mathbf{V}. \tag{24}$$

Indeed, each vector $\mathbf{v}_i$ contains coefficients required for computing $\mathbf{w}_i$ from data points in the kernel space. According to (23) and (24), we propose the following problem:

$$\begin{aligned}\mathbf{V}^*&=\arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}}\frac{tr(\mathbf{V}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{U}_D\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{V})}{tr(\mathbf{V}^T\mathbf{\Phi}^T\mathbf{\Phi}(\mathbf{U}_P+\alpha\mathbf{E})\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{V})}\\&=\arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}}\frac{tr(\mathbf{V}^T\mathbf{K}\mathbf{U}_D\mathbf{K}\mathbf{V})}{tr(\mathbf{V}^T\mathbf{K}(\mathbf{U}_P+\alpha\mathbf{E})\mathbf{K}\mathbf{V})},\end{aligned} \tag{25}$$

where $\mathbf{K}=\mathbf{\Phi}^T\mathbf{\Phi}$ is the kernel matrix. We can solve this problem using the algorithm presented in Table 1 (by setting $\mathbf{S}_1=\mathbf{K}\mathbf{U}_D\mathbf{K}$ and $\mathbf{S}_2=\mathbf{K}(\mathbf{U}_P+\alpha\mathbf{E})\mathbf{K}$).

Though the above problem allows much flexibility for the transformation, it may lead to over-fitting. Below, we introduce a special form of the kernelized problem and show the relation between this special form and the proposed problem in (18) for non-linear metric learning. To this end, we limit columns of matrix $\mathbf{W}$ in (23) such that they are calculated only from positively constrained data points. Let $\{\mathbf{x}_{l_1},\mathbf{x}_{l_2},\ldots,\mathbf{x}_{l_m}\}$ be the set of data points appearing in positive constraints. If we assume that the columns of matrix $\mathbf{W}$ can be written as linear combinations of $\phi(\mathbf{x}_{l_1}),\phi(\mathbf{x}_{l_2}),\ldots,\phi(\mathbf{x}_{l_m})$, the transformation matrix $\mathbf{W}$ can be defined as:

$$\mathbf{W}=\mathbf{\Phi}'\mathbf{V}', \tag{26}$$

where $\mathbf{\Phi}'=[\phi(\mathbf{x}_{l_1}),\phi(\mathbf{x}_{l_2}),\ldots,\phi(\mathbf{x}_{l_m})]$. Thus, using (23) and (26), we obtain the following optimization problem:

$$\begin{aligned}\mathbf{V}'^*&=\operatorname*{argmax}_{\mathbf{V}'^T\mathbf{V}'=\mathbf{I}}\frac{tr(\mathbf{V}'^T\mathbf{\Phi}'^T\mathbf{\Phi}\mathbf{U}_D\mathbf{\Phi}^T\mathbf{\Phi}'\mathbf{V}')}{tr(\mathbf{V}'^T\mathbf{\Phi}'^T\mathbf{\Phi}(\mathbf{U}_P+\alpha\mathbf{E})\mathbf{\Phi}^T\mathbf{\Phi}'\mathbf{V}')}\\&=\operatorname*{argmax}_{\mathbf{V}'^T\mathbf{V}'=\mathbf{I}}\frac{tr(\mathbf{V}'^T\mathbf{K}'\mathbf{U}_D\mathbf{K}'^T\mathbf{V}')}{tr(\mathbf{V}'^T\mathbf{K}'(\mathbf{U}_P+\alpha\mathbf{E})\mathbf{K}'^T\mathbf{V}')},\end{aligned} \tag{27}$$

where $\mathbf{K}'=\mathbf{\Phi}'^T\mathbf{\Phi}$. In this problem, we intend to optimize an $m\times d'$ matrix $\mathbf{V}'$ while in (25), the $n\times d'$ matrix $\mathbf{V}$ must be optimized (usually $m\ll n$). According to (23) and (26), we have

$$\mathbf{Y}=\mathbf{W}^T\mathbf{\Phi}=(\mathbf{\Phi}'\mathbf{V}')^T\mathbf{\Phi}=\mathbf{V}'^T\mathbf{K}'. \tag{28}$$

If we use an exponential kernel, we have $\mathbf{K}'=[\mathbf{K}(.,l_1),\mathbf{K}(.,l_2),\ldots,\mathbf{K}(\mathbf{\Pi},l_m)]=$ and thus the optimization problem obtained in (27) will be equal to the one presented in (18). According to the efficiency of this special form of the kernel-based method, we consider it as the proposed non-linear metric learning method.

Our kernel-based method that has been formulated as an optimization in (25) can be interpreted as a kernel learning method that seeks an appropriate kernel $\mathbf{K}_\mathbf{V}=\mathbf{K}\mathbf{V}\mathbf{V}^T\mathbf{K}$ by learning a proper matrix $\mathbf{V}$ where $\mathbf{K}$ is an initial $n\times n$ kernel matrix (constructed form the input data). As opposed to the introduced methods in Refs. [30,39,40] that need to learn all entries of an $n\times n$ matrix, we need to learn only an $n\times d'$matrix $\mathbf{V}$. Additionally, in the special form introduced in (27), we learn $\mathbf{K}_{\mathbf{V}'}=\mathbf{K}'^T\mathbf{V}'\mathbf{V}'^T\mathbf{K}'$ by finding the optimum $m\times d'$ (m is usually much less than n) matrix $\mathbf{V}'$. Thus, the proposed method relieves the deficiency of the kernel learning methods [30,39,40] that require the optimization of an $n\times n$ matrix. Moreover, our method can yield an explicit non-linear transformation $\mathbf{Y}=\mathbf{V}'^T\mathbf{K}'$.

4. Experimental results

In this section, we explain experiments that we have conducted to compare our linear and non-linear metric learning methods with some existing methods. We measure the effectiveness of semi-supervised metric-learning algorithms by comparing clustering results obtained from using different metrics. We report results on both synthetic and real-world data sets.

4.1. Setup

We compare our linear and non-linear methods with the metric learning algorithms introduced in Refs. [1,28], as they are among the most effective linear methods considering both positive and negative constraints. We also include the LLMA [29] and the kernel-β [30] methods as non-linear metric learning methods in our evaluations.

Similar to the experiments presented in Refs. [23,25,28–31], we use the Euclidean distance (without metric learning) for baseline comparison and apply the k-means clustering algorithm on different distance metrics. Thus, the performance of the methods is evaluated by comparing the following algorithms (the short forms inside parentheses will be used subsequently):

(1). k-means without metric learning (Euclidean);
(2). k-means with the metric learning method introduced by Xiang et al. [1] (Xiang's);
(3). k-means with the extended RCA [28] method for metric learning (ERCA);
(4). k-means with the LLMA [29] method for metric learning (LLMA);
(5). kernel k-means with the kernel obtained by the kernel-β method [30] (Kernel-β);
(6). k-means with our linear metric learning method (Proposed-L);
(7). k-means with our non-linear metric learning method (Proposed-NL);

For the proposed methods, we set the number of nearest neighbors in the geometrical structure to $k=10$ and the regularization parameter to $\alpha=0.2$ (based on few pilot experiments). The reduced dimensionality of our linear method is set to $d'=d/2$ and the output dimensionality of our non-linear method is set to $d'=m/2$. The widow parameter of the neighborhood function in the proposed non-linear method (or, equivalently, the parameter of the exponential kernel used as an initial kernel in problem (27) of Section 3.4), is set to $w=\sum||\mathbf{x}_i-\mathbf{x}_j||_2^2/[0.5n(n-1)]$. However, our method is not such sensitive to the value of the window parameter.

Parameters of other methods are set to the specified values in the corresponding studies. Unfortunately, there is no discussion about the initial kernel used for the kernel-β method in Ref. [30]. Since this method is sensitive to the parameter of the RBF kernel (used as the initial kernel), we have tried to specify this parameter as well as possible. We set it to $\sigma^2=\theta\sum_{i<j}||\mathbf{x}_i-\mathbf{x}_j||_2^2/[n(n-1)]$ where $\theta=6$ is chosen as generally the most appropriate value for

[1] Using the search algorithm presented in Table 1 to solve the problem in (23), the columns of $\mathbf{W}$ in (23) are obtained as eigenvectors of $\mathbf{\Phi}(\mathbf{U}_D-\lambda(\mathbf{U}_P+\alpha\mathbf{E}))\mathbf{\Phi}^T$ and these eigenvectors are linear combinations of $\phi(\mathbf{x}_1),\phi(\mathbf{x}_2),\ldots,\phi(\mathbf{x}_n)$.

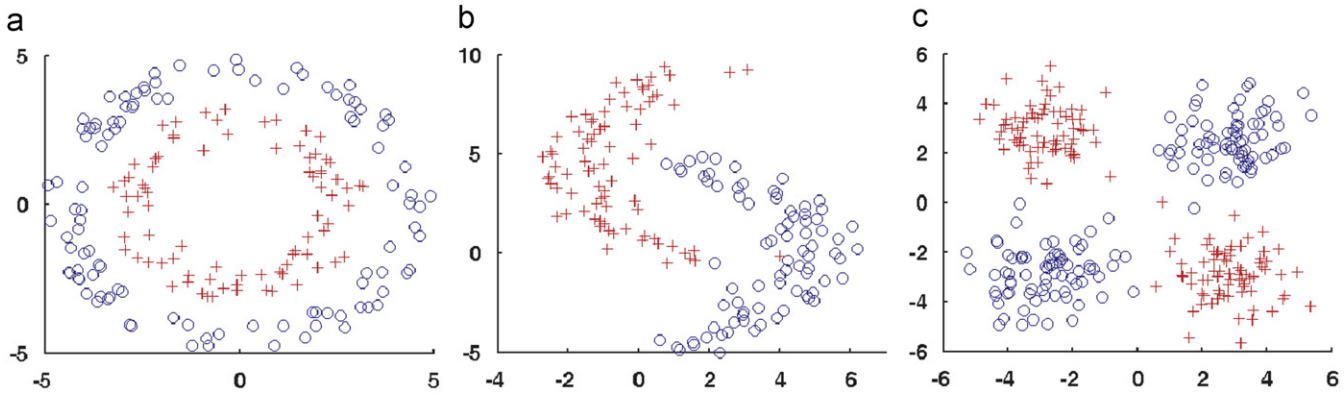

Fig. 1. Synthetic data sets. (a) Data set 1; (b) Data set 2; (c) Data set 3.

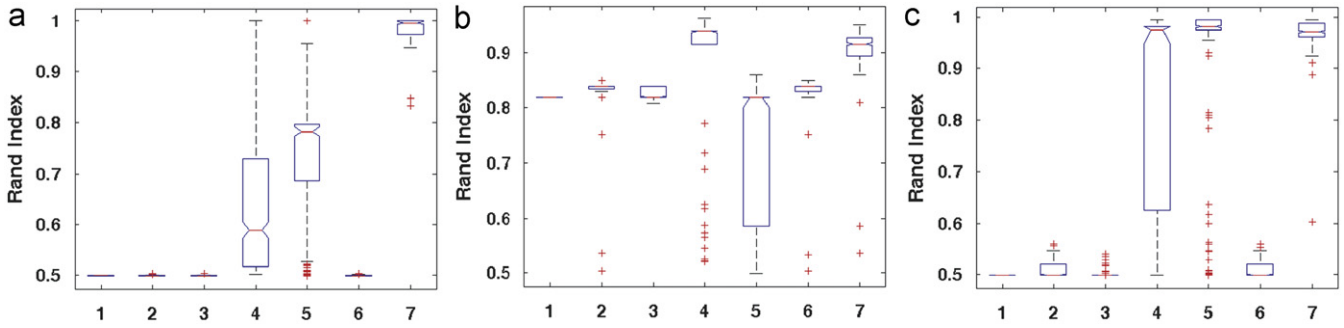

Fig. 2. Clustering results on three synthetic data sets using different metric learning methods. The seven algorithms (numbered in Section 4.1) are as follows: (1) Euclidean, (2) Xiang's, (3) ERCA, (4) LLMA, (5) Kernel-β, (6) Proposed-L, and (7) Proposed-NL. (a) data set 1, nc=20; (b) data set 2, nc=15; (c) data set 3, nc=10.

the data sets used in our experiments. For the parameter β ($w=\beta\bar{d}$) of the spectral LLMA method [29], Chang and Yeung have specified a range that the best value must be chosen from it for each data set individually. But, it seems that they fit this parameter for each data set manually. We set $\beta=0.5$ for the spectral LLMA method as an appropriate value (in the specified range in Ref. [29]) for all data sets.

Similar to experiments in [23,28], we set $nc=|P|=|D|$ for methods that use both positive and negative constraints. As the results depend on P and/or D sets, we generate 20 different P and/or D sets for each data set. Additionally, we run the k-means algorithm 20 times with different random initializations for each P and/or D set. All data sets are normalized before use in the clustering algorithms (each feature is normalized to zero mean and unit variance).

Table 4
UCI data sets used in our experiments.

Data set	n	c	d
Soybean	47	4	35
Protein	116	6	20
Wine	178	3	13
Sonar	208	2	60
Glasses	214	6	10
Ionosphere	351	2	34
Boston housing	506	3	13
Breast cancer	569	2	31
Balance	625	3	4
Diabetes	768	2	8
Transfusion	748	2	4
Segment	2309	7	19

4.2. Performance measure

To measure the performance of clustering in our experiments, we use the *Rand index* which shows how well the clustering results agree with the ground truth clusters [29]. Let n_s be the number of data pairs that are assigned to the same cluster, both in the ground truth and the resultant clustering (i.e., matched pairs) and n_d be the number of data pairs that are assigned to different clusters both in the ground truth and the resultant clustering (i.e., mismatched pairs). Also, n denotes the number of data points. The Rand index is defined as [23]:

$$RI=\frac{n_s+n_d}{n(n-1)/2}. \tag{29}$$

This index will favor assigning data points to different clusters when there are more than two clusters [23,29]. Thus, we modify the Rand index as in Refs. [23,29–31] such that the matched pairs and mismatched pairs are assigned weights to give them equal chances of occurrence (0.5) [29].

4.3. Experiments on synthetic data sets

At first, we conduct experiments on three synthetic data sets displayed in Fig. 1. In this figure, the data points that belong to the same class are shown with the same color and style. Fig. 2 shows the results of applying different algorithms on these data sets as box-plots ($nc=|P|=|D|$). According to this figure, the data sets cannot be clustered well using the standard k-means clustering algorithm. Additionally, the linear methods (Xiang's, ERCA, and Proposed-L) cannot yield proper results on these data sets. These methods are not able to learn a metric that fits to the

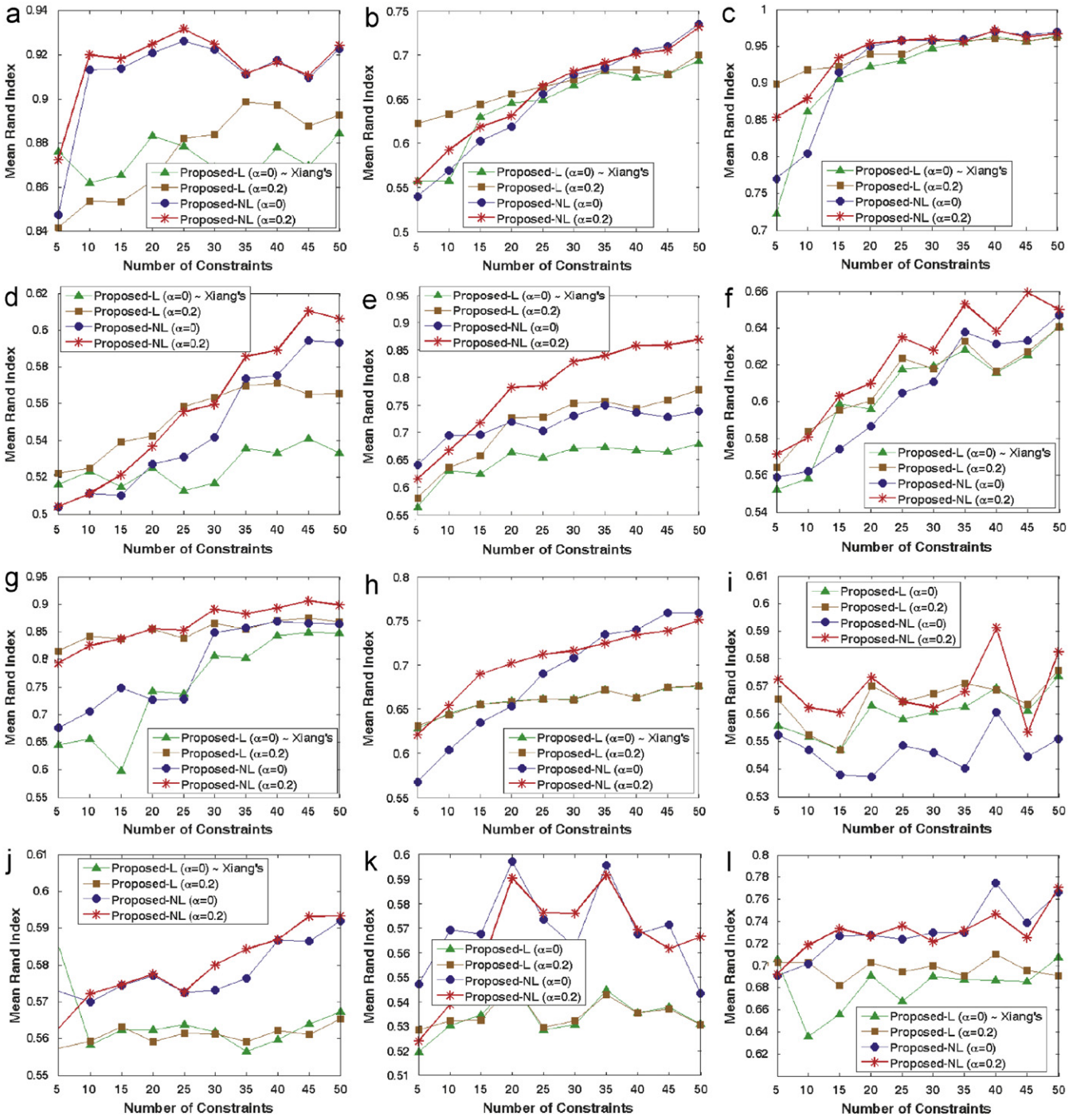

Fig. 3. Average Rand index curves of our linear and non-linear methods on the UCI data sets for both cases of considering the geometrical structure ($\alpha=0.2$) and ignoring it ($\alpha=0$). (a) Soybean; (b) Protein; (c) Wine; (d) Sonar; (e) Ionosphere; (f) Boston housing; (g) Breast cancer; (h) Balance; (i) Diabetes; (j) Glasses; (k) Transfusion; (l) Segment.

complex patterns in Fig. 1. According to the structure of the clusters in these data sets, we need to learn a non-linear transformation. LLMA and kernel-β as non-linear metric learning methods show good results on some of these data sets. However, the proposed non-linear method performs well on all of them.

4.4. Experiments on UCI data sets

In this section, we conduct experiments on some real-world data sets obtained from the Machine Learning Repository[2] of the University of California, Irvine (UCI). Table 4 shows the properties of these data sets. The three columns of the table show the number of data points n, the number of classes c, and the number of attributes d for each data set.

Before comparing the performance of different metric learning methods, we first evaluate the effect of the regularizer term (incorporating the geometrical structure) in the proposed linear and nonlinear methods. In Fig. 3, we show the average Rand index curve of our linear and non-linear methods for both cases of

[2] http://archive.ics.uci.edu/ml/

considering the geometrical structure, i.e., $\alpha=0.2$, and ignoring it, i.e., $\alpha=0$, (when we set $\alpha=0$, the proposed linear method will be equivalent to Xiang's). According to this figure, using the geometrical structure (via the regularizer term) usually improves results of both proposed linear and non-linear methods especially when the amount of supervisory information is low. Additionally, using the geometrical structure usually yields more improvement for higher dimensional data sets (e.g., Fig. 3(e) and (g)) particularly in the proposed linear method. However, it is not such helpful to use the regularizer term for low-dimensional data sets (e.g., Fig. 3(h) and (k)).

To evaluate the performance of different methods, we obtain results of these methods for different numbers of constraints on the UCI data sets. In Fig. 4, the average Rand index of each method (over different sets of constraints and different runs of the clustering method as explained in Section 4.1) vs. the number of constraints has been displayed. According to this figure, the proposed non-linear method generally outperforms all the other methods. Although the performance of the proposed linear method is not comparable to that of the proposed non-linear method, this method is the second best method in many cases.

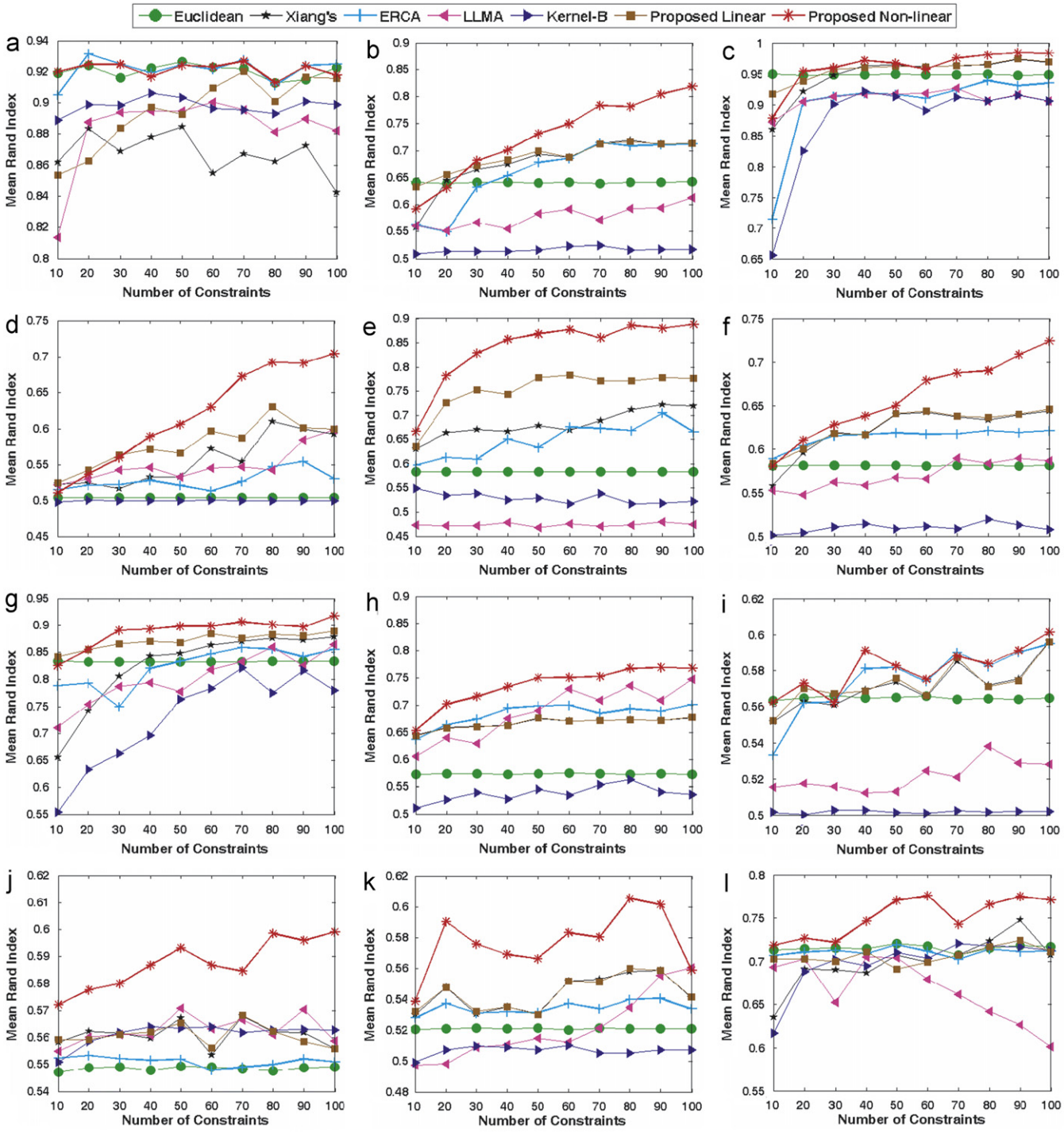

Fig. 4. Average Rand index curves of different methods on the UCI data sets. (a) Soybean ; (b) Protein; (c) Wine; (d) Sonar; (e) Ionosphere; (f) Boston housing; (g) Breast cancer; (h) Balance; (i) Diabetes; (j) Glasses; (k) Transfusion; (l) Segment.

In Fig. 5, we show the spread of the Rand index values (shown in Fig. 4) for a specified number of constraints, $nc=30$, on each data set. As we can see in Fig. 5, our non-linear method generally yields the best results. For more accurate comparison of the results shown in Fig. 5, we do a paired t-test with significance level 0.05 that have also been used in [30,31] to evaluate the results significantly. Table 5 summarizes the comparison of the results shown in Fig. 5 using a paired t-test. The relation $x < y$ indicates that the Rand index values of the latter method are significantly higher than those of the former one. Similarly, we use $x \sim y$ to denote that the results of two methods are not significantly different for the given confidence level. From the paired t-test results, we find with a 95% confidence level that our non-linear method generally outperforms all the other methods. Moreover, our linear method in most cases is the second best method. To compare the proposed linear method with that of Xiang et al. [1], the recent and powerful metric learning method, we find that our linear method is clearly better than Xiang's for 9 out of the 12 data sets and comparable with it for three data sets.

4.5. Experiments on MNIST digits

The last set of experiments is performed on the MNIST database containing handwritten digits. There are 60 000 digit images in the training set of this database. Digits have been centered and normalized to 28×28 gray-scale images. In our experiments, we choose 500 images randomly for each digit. We set the number of constraints to $nc = |P| = |D| = 30$ for all

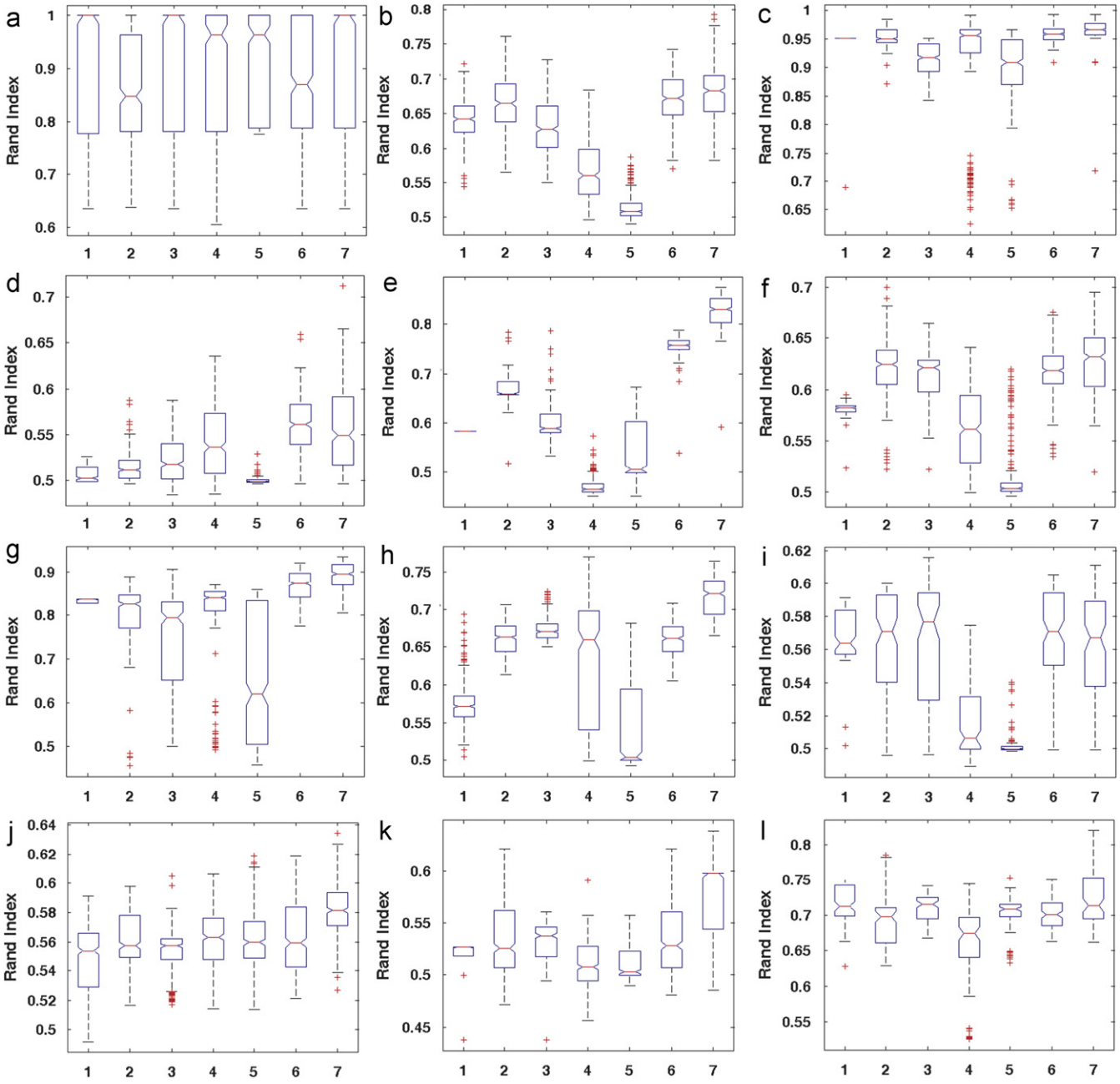

Fig. 5. Clustering results on UCI data sets using different metric learning methods ($nc=30$). The seven algorithms (numbered in Section 4.1) are as follows: (1) Euclidean, (2) Xiang's, (3) ERCA, (4) LLMA, (5) Kernel-β, (6) Proposed-L, and (7) Proposed-NL. (a) Soybean; (b) Protein; (c) Wine; (d) Sonar; (e) Ionosphere; (f) Boston housing; (g) Breast cancer; (h) Balance; (i) Diabetes; (j) Glasses; (k) Transfusion; (l) Segment.

Table 5
Paired t-test statistical evaluation of the clustering results shown in Fig. 5.

Data set	Paired t-test											
Soybean	Xiang's	$<$	Proposed-L	$\sim$	LLMA	$\sim$	Kernel-β	$<$	Euclidean	$\sim$	ERCA	$\sim$	Proposed-NL
Protein	Kernel-β	$<$	LLMA	$<$	ERCA	$<$	Euclidean	$<$	Xiang's	$<$	Proposed-L	$<$	Proposed-NL
Wine	Kernel-β	$<$	ERCA	$\sim$	LLMA	$<$	Euclidean	$\sim$	Xiang's	$<$	Proposed-L	$\sim$	Proposed-NL
Sonar	Kernel-β	$<$	Euclidean	$<$	Xiang's	$<$	ERCA	$<$	LLMA	$<$	Proposed-L	$\sim$	Proposed-NL
Ionosph.	LLMA	$<$	Kernel-β	$<$	Euclidean	$<$	ERCA	$<$	Xiang's	$<$	Proposed-L	$<$	Proposed-NL
Boston	Kernel-β	$<$	LLMA	$<$	Euclidean	$<$	ERCA	$<$	Xiang's	$\sim$	Proposed-L	$<$	Proposed-NL
Breast	Kernel-β	$<$	ERCA	$<$	LLMA	$<$	Xiang's	$<$	Euclidean	$<$	Proposed-L	$<$	Proposed-NL
Balance	Kernel-β	$<$	Euclidean	$<$	LLMA	$<$	Xiang's	$\sim$	Proposed-L	$<$	ERCA	$<$	Proposed-NL
Diabetes	Kernel-β	$<$	LLMA	$<$	Xiang's	$<$	Euclidean	$\sim$	Proposed-L	$\sim$	ERCA	$\sim$	Proposed-NL
Glasses	Euclidean	$<$	ERCA	$<$	Kernel-β	$\sim$	LLMA	$\sim$	Xiang's	$\sim$	Proposed-L	$<$	Proposed-NL
Transf.	LLMA	$\sim$	Kernel-β	$<$	Euclidean	$<$	Xiang's	$<$	ERCA	$\sim$	Proposed-L	$<$	Proposed-NL
Segment	LLMA	$<$	Xiang's	$<$	Proposed-L	$\sim$	Kernel-β	$<$	Euclidean	$\sim$	ERCA	$<$	Proposed-NL

Table 6
Mean and variance of the Rand index values obtained for different methods on some subsets of the MNIST database.

Subset	Euclidean	ERCA	Xiang's	LLMA	Kernel-β	Proposed linear	Proposed non-linear
{1,2,3}	0.8718 ($\pm$0.0452)	0.8528 ($\pm$0.0800)	0.8605 ($\pm$0.0425)	0.6284 ($\pm$0.0362)	0.5607 ($\pm$0.0911)	**0.8925** ($\pm$0.0444)	0.8688 ($\pm$0.0421)
{4,5,6}	0.7702 ($\pm$0.0782)	0.8702 ($\pm$0.0924)	0.8296 ($\pm$0.0893)	0.7128 ($\pm$0.0724)	0.5217 ($\pm$0.0409)	0.8989 ($\pm$0.0723)	**0.9042** ($\pm$0.0517)
{7,8,9}	0.7051 ($\pm$0.0149)	0.7778 ($\pm$0.0604)	0.7411 ($\pm$0.0408)	0.6371 ($\pm$0.0307)	0.5062 ($\pm$0.0093)	**0.8339** ($\pm$0.0633)	0.8216 ($\pm$0.0166)
{0,1,2,3}	0.8506 ($\pm$0.0456)	0.8411 ($\pm$0.0650)	0.8270 ($\pm$0.0542)	0.6576 ($\pm$0.0326)	0.5901 ($\pm$0.0658)	**0.8733** ($\pm$0.0315)	0.8034 ($\pm$0.0718)
{3,4,5,6}	0.7507 ($\pm$0.0344)	0.7979 ($\pm$0.0748)	0.7771 ($\pm$0.0466)	0.7082 ($\pm$0.0354)	0.5675 ($\pm$0.0528)	0.8199 ($\pm$0.0499)	**0.8330** ($\pm$0.0298)
{6,7,8,9}	0.7647 ($\pm$0.0183)	0.7809 ($\pm$0.0521)	0.7484 ($\pm$0.0376)	0.6553 ($\pm$0.0264)	0.5295 ($\pm$0.0363)	0.7965 ($\pm$0.0484)	**0.7991** ($\pm$0.0474)

experiments performed on subsets of the MNIST database. Table 6 shows the results of different methods for clustering of some subsets including samples of three or four of digits. For each method, the mean Rand index and the standard deviation over different runs (corresponding to different sets of constraints and different initializations of the k-means clustering algorithm) are shown in Table 6. From these results, we can see that our linear and non-linear methods show better results than the other methods. Compared to Xiang's (that is equal to our linear method when we set $\alpha=0$), the proposed linear method yields better results on all the subsets. Since the amount of supervisory information is low in these experiments, the incorporation of the geometrical structure of the data along with the pairwise constraints has been much helpful.

Since m (the unique number of data points appearing in positive constraints) is low (i.e., $m \leq 2 \times nc = 60$) compared with the dimensionality of data points ($d = 28 \times 28 = 784$), our non-linear metric learning method is much faster than our linear method and Xiang's. Indeed, we learn a matrix of size $m \times d_1$ ($d_1 = m/2$) in our non-linear method while our linear method learn a matrix of size $d \times d_2$ ($d_2 = d/2$). Additionally, the k-means clustering algorithm on the transformed data points by our non-linear method runs much faster because of the lower dimensionality of these data points $d_1 \ll d_2$.

5. Conclusions and future work

In this paper, we introduced a novel metric learning method for semi-supervised clustering. We proposed a general framework for learning linear and non-linear transformations using both positive and negative constraints. The proposed methods have been formulated as constrained trace ratio problems that can be solved efficiently. We considered the geometrical structure of the data along with the pairwise constraints in the proposed optimization problems. We showed that the proposed non-linear method can be considered as an efficient kernel learning method. Experimental results on the synthetic, UCI, and MNIST data sets showed the superior performance of the proposed metric learning methods. In the future, we will investigate other forms of non-linear metric learning methods. We also intend to evaluate the performance of our method on other real-world applications.

References

[1] S. Xiang, F. Nie, C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recognition 41 (12) (2008) 3600–3612.
[2] L. Yang, R. Jin, Distance metric learning: a comprehensive survey, Technical Report, Michigan State University (〈http://www.cse.msu.edu/~yangliu1/frame_survey_v2.pdf〉), 2006.
[3] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighborhood components analysis, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2004, pp. 513–520.
[4] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2006, pp. 1473–1480.
[5] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (6) (1996) 607–616.
[6] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2006, pp. 451–458.
[7] J.H. Friedman, Flexible metric nearest neighbor classification, Technical Report, Statistics Department, Stanford University, 1994.
[8] Z.H. Zhang, J.T. Kwok, D.Y. Yeung, Parametric distance metric learning with label information, in: IJCAI, Acapulco, Mexico, 2003, pp. 1450–1452.
[9] M.H.C. Law, Clustering, dimensionality reduction, and side information, Ph.D. Dissertation, Michigan University, 2006.
[10] S. Basu, Semi-supervised clustering: probabilistic models, algorithms and experiments, Ph.D. Dissertation, University of Texas at Austin, 2005.
[11] M.H.C. Law, A. Topchy, A.K. Jain, Model-based clustering with probabilistic constraints, in: SIAM Conference on Data Mining, 2005, pp. 641–645.

[12] S. Basu, A. Banerjee, R.J. Mooney, Semi-supervised clustering by seeding, in: 19th International Conference on Machine Learning (ICML-02), Sydney, Australia, 2002, pp. 19–26.
[13] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained K-means clustering with background knowledge, in: 18th International Conference on Machine Learning (ICML01), 2001, pp. 577–584.
[14] Z. Lu, T. Leen, Semi-supervised learning with penalized probabilistic clustering, in: Advances in NIPS 17, MIT Press, Cambridge, MA, USA, 2005, pp. 849–856.
[15] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Machine Learning 56 (1-3) (2004) 89–113.
[16] T. Lange, M.H. Law, A.K. Jain, J. Buhmann, Learning with constrained and unlabelled data, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 730–737.
[17] Q. Zhao, D.J. Miller, Mixture modeling with pair wise instance-level class constraints, Neural Computation 17 (11) (2005) 2482–2507.
[18] S.X. Yu, J. Shi, Segmentation given partial grouping constraints, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 173–183.
[19] B. Kulis, S. Basu, I. Dhillon, R.J. Mooney, Semi-supervised graph clustering: a kernel approach, in: 22nd International Conference on Machine Learning, 2005, pp. 457–464.
[20] S. Basu, M., Bilenko, R.J. Mooney, A probabilistic framework for semi-supervised clustering, in: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 59–68.
[21] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian mixture models with EM using equivalence constraints, in: Advances in NIPS 16, MIT Press, Cambridge, MA, USA, 2004, pp. 465–472.
[22] D. Klein, S.D. Kamvar, C. Manning, From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering, in: 19th International Conference on Machine Learning (ICML-02), Sydney, Australia, 2002, pp. 307–314.
[23] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning with application to clustering with side information, in: Advances in NIPS 15, MIT Press, Cambridge, MA, USA, 2003, pp. 505–512.
[24] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: 21st International Conference on Machine Learning, 2004, pp. 81–88.
[25] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, Learning a Mahalanobis metric from equivalence constraints, Journal of Machine Learning Research 6 (2005) 937–965.
[26] T. Hertz, A. Bar-Hillel, D. Weinshall, Boosting margin-based distance functions for clustering, in: Proceedings of 21st International Conference on Machine Learning, 2004, pp. 393–400.
[27] M. Schultz, T. Joachims, Learning a distance metric from relative comparisons, in: Advances in NIPS 16, MIT Press, Cambridge, MA, USA, 2004, pp. 41–48.
[28] D.Y. Yeung, H. Chang, Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints, Pattern Recognition 39 (2006) 1007–1010.
[29] H. Chang, D.Y. Yeung, Locally linear metric adaptation with application to semi-supervised clustering and image retrieval, Pattern Recognition 39 (2006) 1253–1264.
[30] D.Y. Yeung, H. Chang, A Kernel approach for semi-supervised metric learning, IEEE Transactions on Neural Networks 18 (1) (2007) 141–149.
[31] H. Chang, D.Y. Yeung, W.K. Cheung, Relaxational metric adaptation and its application to semi-supervised clustering and content-based image retrieval, Pattern Recognition 39 (2006) 1905–1917.
[32] T. De Bie, M. Momma, N. Cristianini, Efficiently learning the metric with side-information, Lecture Notes in Artificial Intelligence 2842 (2003) 175–189.
[33] N. Kumar, K. Kummamuru, Semi-supervised clustering with metric learning using relative comparisons, IEEE Transactions on Knowledge and Data Engineering 20 (4) (2007) 496–503.
[34] S.C.H. Hoi, W. Liu, M.R. Lyu, W.-Y. Ma, Learning distance metrics with contextual constraints for image retrieval, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, USA, 2006, pp. 2072–2078.
[35] Y.F. Guo, S.J. Li, J.Y. Yang, T.T. Shu, L.D. Wu, A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition, Pattern Recognition Letters 24 (1) (2003) 147–158.
[36] H. Wang, S. Yan, D. Xu, X. Tang, T. Huang, Trace ratio vs. ratio trace for dimensionality reduction, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[37] S.T. Roweis, L.K. Saul, Non-linear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[38] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: Proceedings of the IEEE International Conference on Computer Vision (ICCI'07), Brazil, 2007, pp. 1–7.
[39] S.C.H. Hoi, R. Jin, M.R. Lyu, Learning nonparametric kernel matrices from pairwise constraints, in: 24th International Conference on Machine Learning (ICML'07), New York, USA, 2007, pp. 361–368.
[40] J. Zhuang, I.W. Tsang, S.C.H. Hoi, SimpleNPKL: simple non-parametric kernel learning, in: 26th International Conference on Machine Learning (ICML'09), Montreal, Canada, 2009, pp. 1273–1280.

About the Author—MAHDIEH SOLEYMANI BAGHSHAH received her B.S. and M.S. degrees from Department of Computer Engineering, Sharif University of Technology, Iran in 2003 and 2005. She is now a Ph.D. candidate at Sharif University of Technology. Her research interests include machine learning and pattern recognition with primary emphasis on semi-supervised learning and clustering.

About the Author—SAEED BAGHERI SHOURAKI received his B.Sc. in Electrical Engineering and M.Sc. in Digital Electronics from Sharif University of Technology, Tehran, Iran, in 1985 and 1987. He joined soon to Computer Engineering Department of Sharif University of Technology as a faculty member. He received his Ph.D. on fuzzy control systems from Tsushin Daigaku (University of Electro-Communications), Tokyo, Japan, in 2000. He continued his activities in Computer Engineering Department up to 2008. He is currently an associate professor in Electrical Engineering Department of Sharif University of Technology. His research interests include control, robotics, artificial life, and soft computing.