Semi-Supervised Metric Learning Using Pairwise Constraints

Mahdieh Soleymani Baghshah
Department of Computer Engineering
Sharif University of Technology
soleyman@ce.sharif.edu

Saeed Bagheri Shouraki
Department of Electrical Engineering
Sharif University of Technology
bagheri-s@sharif.edu

Abstract

Distance metric has an important role in many machine learning algorithms. Recently, metric learning for semi-supervised algorithms has received much attention. For semi-supervised clustering, usually a set of pairwise similarity and dissimilarity constraints is provided as supervisory information. Until now, various metric learning methods utilizing pairwise constraints have been proposed. The existing methods that can consider both positive (must-link) and negative (cannot-link) constraints find linear transformations or equivalently global Mahalanobis metrics. Additionally, they find metrics only according to the data points appearing in constraints (without considering other data points). In this paper, we consider the topological structure of data along with both positive and negative constraints. We propose a kernel-based metric learning method that provides a non-linear transformation. Experimental results on synthetic and real-world data sets show the effectiveness of our metric learning method.

1 Introduction

Distance metric is a key issue in many machine learning algorithms [Xiang *et al.*, 2008]. Over the last few years, there has been considerable research on distance metric learning [Yang and Jin, 2006]. Many of the earlier studies optimize the metric with class labels for classification tasks [Lebanon, 1994; Hastie and Tibshirani, 1996; Zhang *et al.*, 2003; Goldberger *et al.*, 2004]. More recently, researchers have given much attention to distance learning for semi-supervised algorithms and specially semi-supervised clustering algorithms. Since 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 (for example, $\mathbf{x}$ is closer to $\mathbf{y}$ than to $\mathbf{z}$) have also been considered in some studies [Schultz and Joachims, 2004; Kumar and Kummamuru, 2007].

Distance learning based on constraints has been studied by many researchers. [Klein *et al.*, 2002] introduced one of the first distance learning methods for semi-supervised clustering. This method finds a distance metric according to the shortest path in a version of the similarity graph that has been altered by positive constraints. Some latter studies have considered a more popular approach that learns global Mahalanobis metrics from pairwise constraints. [Xing *et al.*, 2003] proposed a convex optimization problem to learn a global Mahalanobis metric according to pairwise constraints. [Bar–Hillel *et al.*, 2005] devised a more efficient, non-iterative algorithm called *Relevant Component Analysis* (RCA) for learning a Mahalanobis metric. This method only incorporates positive constraints. An extension of the RCA method that can consider both positive and negative constraints has also been introduced [Yeung and Chang, 2006]. [Hoi *et al.*, 2006] proposed *Discriminative Component Analysis* (DCA) method that uses the ratio of between chunklets and within chunklets convariance determinants as the objective function. Recently, [Xiang *et al.*, 2008] introduced the trace ratio optimization problem as a more appropriate objective function. They have also provided a nice heuristic search to solve this problem. [Chang and Yeung, 2006] proposed a method that finds a locally linear metric using positive constraints. However, the objective function of this method has many local optima and the topology cannot be preserved well during this approach [Yeung and Chang, 2007]. [Chang *et al.*, 2006] proposed 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 [Chang *et al.*, 2006]. In [Yeung and Chang, 2007], two kernel-based metric learning methods have been presented that have some limitations [Yeung and Chang, 2007].

Among the existing metric learning methods for semi-supervised clustering, some of them [Xing *et al.*, 2003; Yeung and Chang, 2006; Hoi *et al.*, 2006; Xiang *et al.*, 2008] can incorporate both positive and negative constraints for

metric learning. But, none of these methods (that can consider both positive and negative constraints) have used the topological structure of data. Moreover, all of them find a linear transformation (or equivalently a Mahalanobis distance metric) according to pairwise constraints. In this paper, we formulate an objective function considering all data points along with pairwise similarity and dissimilarity constraints and generalize this objective function to learn a nonlinear transformation. We find the global optimum of the proposed objective function using the search algorithm introduced by [Xiang *et al.*, 2008]. As our metric learning method can find nonlinear metrics and also it considers the topological structure of data, it shows higher capability compared with the existing methods.

The rest of this paper is organized as follows: In Section 2, a metric learning method considering both pairwise constraints and the intrinsic structure of data is proposed. In this section, first we introduce an optimization problem to find an appropriate linear metric and then we provide a nonlinear metric learning method as a special case of the kernelized version of our linear method. Section 3 presents experimental results on some synthetic and real-world data sets. Concluding remarks are given in the last section.

2 Proposed Metric Learning Approach

In this section, first we propose a metric learning method using pairwise constraints while considering the topological structure of data. Then, we introduce a non-linear extension of this method.

2.1 Linear metric learning

To find an appropriate metric, the manifold structure of data is incorporated along with pairwise (positive and negative) constraints in our method. We are given a set of data points $X = \{\mathbf{x}_i\}_{i=1}^n$ and two sets including positive $P = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are in the same class}\}$ and negative $D = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are in two different classes}\}$ constraints. The optimization problem for the transformation $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ is defined as:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D} \left(\mathbf{y}_i - \mathbf{y}_j\right)^2}{\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P} \left(\mathbf{y}_i - \mathbf{y}_j\right)^2 + \alpha J(\mathbf{W})}. \quad (1)$$

Here, $J(\mathbf{W})$ shows a penalty (regularizer) term that tries to preserve the topological structure of data during the transformation, and $\alpha \geq 0$ balances between distances of similar pairs and the regularizer term. In (1), a metric $\mathbf{A} = \mathbf{W}\mathbf{W}^T$ is sought that makes distances between point pairs in D as large as possible while making a combination of distances between point pairs in P and the penalty term as small as possible. The constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ has also been considered to prevent improper solutions [Xiang *et al.*, 2008; Hoi *et al.*, 2006]. Based on $\mathbf{y} = \mathbf{W}^T\mathbf{x}$, we can rewrite the optimization problem in (1) as:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W}) + \alpha J(\mathbf{W})}. \quad (2)$$

Here, tr shows the trace operator, and $\mathbf{S}_w$ and $\mathbf{S}_b$ are defined as:

$$\mathbf{S}_w = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in P} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T. \quad (3)$$

$$\mathbf{S}_b = \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D} \left(\mathbf{x}_k - \mathbf{x}_l\right)\left(\mathbf{x}_k - \mathbf{x}_l\right)^T. \quad (4)$$

If we set $\alpha = 0$ in (2), the problem becomes similar to the optimization problem introduced in [Xiang *et al.*, 2008]. In (2), $\mathbf{W} \in R^{d\times d'}$ shows the transformation matrix (with $d' \leq d$), where d and d' denote the dimensionality of the input and the transformed space respectively. When $d = d'$, we have $\mathbf{W}\mathbf{W}^T = \mathbf{W}^T\mathbf{W} = \mathbf{I}$ which generates the Euclidean metric [Xiang *et al.*, 2008] and thus we consider $d' < d$.

To preserve the topological structure of data via the term $J(\mathbf{W})$ in the objective function, we use the idea of *Locally Linear Embedding* (LLE) method [Roweis and Saul, 2000]. Indeed, we try to preserve the manifold structure of data by retaining locally linear relationships between close data points in the transformed space. Given the set of data points, a k-nearest neighbor graph models the relations between close data points. The optimal weight matrix $\mathbf{S}^* = [s_{ij}^*]$ providing minimal error for the linear reconstruction of data points from their neighbors is obtained according to:

$$\mathbf{S}^* = \min_{\mathbf{S}=[s_{ij}]} \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j\in N_k(\mathbf{x}_i)} s_{ij}\mathbf{x}_j \right\|^2, \text{ s.t. } \forall i, \sum_{\mathbf{x}_j\in N_k(\mathbf{x}_i)} s_{ij} = 1, \quad (5)$$

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 [Roweis and Saul, 2000]. After finding the optimal weight matrix $\mathbf{S}^*$, we define the penalty term $J(\mathbf{W})$ as:

$$J(\mathbf{W}) \equiv \sum_{i=1}^n \left\| \mathbf{y}_i - \sum_{\mathbf{x}_j\in N_k(\mathbf{x}_i)} s_{ij}^*\mathbf{y}_j \right\|^2 = tr\left(\mathbf{Y}\mathbf{E}\mathbf{Y}^T\right), \quad (6)$$

where $\mathbf{E} = (\mathbf{I} - \mathbf{S}^*)^T(\mathbf{I} - \mathbf{S}^*)$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$. Thus, for $\mathbf{y} = \mathbf{W}^T\mathbf{x}$, we have $J(\mathbf{W}) = tr\left(\mathbf{W}^T\mathbf{X}\mathbf{E}\mathbf{X}^T\mathbf{W}\right)$ where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$. In (6), $J(\mathbf{W})$ denotes the locally linear reconstruction error of the transformed data points according to the weight matrix $\mathbf{S}^*$. Indeed, it reflects the assumption of preserving the geometrical structure of data in the transformed space.

By substituting (6) into (2), we have

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{tr(\mathbf{W}^T(\mathbf{S}_w + \alpha\mathbf{X}\mathbf{E}\mathbf{X}^T)\mathbf{W})}. \quad (7)$$

This problem cannot be solved by eigenvalue decomposition approaches. However, we can use the search algorithm introduced in [Xiang *et al.*, 2008] for optimizing the constrained trace ratio problem. Since the matrix $\mathbf{E}$ and consequently $\mathbf{S}_w + \alpha\mathbf{X}\mathbf{E}\mathbf{X}^T$ are symmetric positive semi-definite matrices, the necessary condition of the introduced algorithm in [Xiang *et al.*, 2008] is satisfied even when we consider the regularizer term in the objective function. Thus, we can use the heuristic search algorithm presented in [Xiang *et al.*, 2008] to solve the proposed optimization problem and find the optimal transformation matrix $\mathbf{W}^*$ or equivalently the optimal metric $\mathbf{A}^* = \mathbf{W}^*(\mathbf{W}^*)^T$.

2.2 Kernel-Based Metric Learning

In this section, first we introduce a kernelized version of our linear metric learning method presented in the above section and then we consider a special case of this kernelized method as the proposed kernel-based metric learning 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 $\phi: R^d \to F$ [Cai *et al.*, 2007]. For a proper chosen ϕ, we can define an inner product on F using Mercer kernel $<\phi(\mathbf{x}),\phi(\mathbf{y})>= K(\mathbf{x},\mathbf{y})$ where $K(.,.)$ is a positive semi-definite kernel function. Many choices for kernel functions that satisfy Mercer's condition are possible such as polynomial, Gaussian, and exponential kernels.

In the kernel-based method, we apply the transformation matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{d'}]$ consisting of orthonormal vectors $\{\mathbf{w}_i \in F \mid i = 1,2,...,d'\}$ $(<\mathbf{w}_i, \mathbf{w}_j>= \delta_{i,j})$ on $\phi(\mathbf{x}) \in F$ via the transformation $\mathbf{y} = \mathbf{W}^T\phi(\mathbf{x})$. This transformation performs a mapping from R^d to $R^{d'}$. To express the optimization problem in the kernel space, first we rewrite the problem in (1) as:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{Y}\mathbf{U}_D\mathbf{Y}^T)}{tr(\mathbf{Y}\mathbf{U}_P\mathbf{Y}^T) + \alpha tr(\mathbf{Y}\mathbf{E}\mathbf{Y}^T)}, \tag{8}$$

where

$$\mathbf{U}_P = \mathbf{D}_P - \mathbf{S}_P. \tag{9}$$

$$\mathbf{S}_P(i,j) = \begin{cases} 1 & (\mathbf{x}_i, \mathbf{x}_j) \in P \text{ or } (\mathbf{x}_j, \mathbf{x}_i) \in P \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

$$\mathbf{D}_P(i,j) = \begin{cases} \sum_j \mathbf{S}_P(i,j) & i = j \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

$\mathbf{U}_D$ can also be defined similarly according to the set of negative constraints D. Let $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_n)]$ denotes the data points in the kernel space. As $\mathbf{y} = \mathbf{W}^T\phi(\mathbf{x})$, we obtain the following optimization problem in the RKHS:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \frac{tr(\mathbf{W}^T\mathbf{\Phi}\mathbf{U}_D\mathbf{\Phi}^T\mathbf{W})}{tr(\mathbf{W}^T\mathbf{\Phi}(\mathbf{U}_P + \alpha\mathbf{E})\mathbf{\Phi}^T\mathbf{W})}. \tag{12}$$

Since the vectors $\{\mathbf{w}_i \in F \mid i = 1,2,...,d'\}$ 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 $\mathbf{W} = \mathbf{\Phi}\mathbf{V}$. Indeed, each n-dimensional vector $\mathbf{v}_i$ contains coefficients required for computing $\mathbf{w}_i$ from data points in the kernel space. Thus, we can rewrite (12) as:

$$\mathbf{V}^* = \arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}} \frac{tr(\mathbf{V}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{U}_D\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{V})}{tr(\mathbf{V}^T\mathbf{\Phi}^T\mathbf{\Phi}(\mathbf{U}_P + \alpha\mathbf{E})\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{V})}$$
$$= \arg\max_{\mathbf{V}^T\mathbf{V}=\mathbf{I}} \frac{tr(\mathbf{V}^T\mathbf{K}\mathbf{U}_D\mathbf{K}\mathbf{V})}{tr(\mathbf{V}^T\mathbf{K}(\mathbf{U}_P + \alpha\mathbf{E})\mathbf{K}\mathbf{V})}, \tag{13}$$

where $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ is the kernel matrix. Since $\mathbf{K}\mathbf{U}_D\mathbf{K}$ and $\mathbf{K}(\mathbf{U}_P + \alpha\mathbf{E})\mathbf{K}$ are positive semi-definite matrices, we can solve this optimization problem using the search algorithm introduced in [Xiang *et al.*, 2008].

Unfortunately, for many kernel learning settings, the amount of supervisory information is typically very limmited [Yeung *et al.*, 2007]. Allowing too much flexibility in the model while having limited supervisory information may lead to model over-fitting [Yeung *et al.*, 2007]. Here, we restrict the transformation matrix $\mathbf{W}$ in (12) and revise the optimization problem for finding a smaller matrix. We limit the columns of matrix $\mathbf{W}$ such that they are calculated only from positively constrained data points (these data points are usually the most informative ones). Let $\{\mathbf{x}_{l_1}, \mathbf{x}_{l_2}, \ldots, \mathbf{x}_{l_m}\}$ be the set of data points appearing in positive constraints and m be the number of unique data points involved in positive constraints. If we restrict the columns of matrix $\mathbf{W}$ to 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 $\mathbf{W} = \mathbf{\Phi}'\mathbf{V}'$ where $\mathbf{V}'$ is an $m \times d'$ matrix and $\mathbf{\Phi}' = [\phi(\mathbf{x}_{l_1}), \phi(\mathbf{x}_{l_2}), \ldots, \phi(\mathbf{x}_{l_m})]$. In this case, we obtain the following optimization problem:

$$\mathbf{V}'^* = \arg\max_{\mathbf{V}'^T\mathbf{V}'=\mathbf{I}} \frac{tr(\mathbf{V}'^T\mathbf{\Phi}'^T\mathbf{\Phi}\mathbf{U}_D\mathbf{\Phi}^T\mathbf{\Phi}'\mathbf{V}')}{tr(\mathbf{V}'^T\mathbf{\Phi}'^T\mathbf{\Phi}(\mathbf{U}_P + \alpha\mathbf{E})\mathbf{\Phi}^T\mathbf{\Phi}'\mathbf{V}')}$$
$$= \arg\max_{\mathbf{V}'^T\mathbf{V}'=\mathbf{I}} \frac{tr(\mathbf{V}'^T\mathbf{K}'\mathbf{U}_D\mathbf{K}'^T\mathbf{V}')}{tr(\mathbf{V}'^T\mathbf{K}'(\mathbf{U}_P + \alpha\mathbf{E})\mathbf{K}'^T\mathbf{V}')}, \tag{14}$$

where $\mathbf{K}' = \mathbf{\Phi}'^T\mathbf{\Phi} = [\mathbf{K}(.,l_1), ..., \mathbf{K}(.,l_m)]$. Also, we have $\mathbf{Y} = \mathbf{W}^T\mathbf{\Phi} = (\mathbf{\Phi}'\mathbf{V}')^T\mathbf{\Phi} = \mathbf{V}'^T\mathbf{K}'$. According to (14), we need to optimize a smaller matrix $\mathbf{V}'$ than $\mathbf{V}$. We can use the heuristic search algorithm introduced in [Xiang *et al.*,

[1] Using the search algorithm presented in [Xiang *et al.* 2008], the columns of $\mathbf{W}$ in (12) 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 $\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n)$.

2008] to find the optimal matrix $\mathbf{V}'$ and then we compute the transformed data points accordingly $\mathbf{Y} = \mathbf{V}'^{T}\mathbf{K}'$. We use the exponential kernel $K(\mathbf{x},\mathbf{y}) = \exp\left(-\|\mathbf{x}-\mathbf{y}\|/w\right)$ in our method.

3 Experimental Results

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

3.1 Comparison of Metrics

We compare our non-linear method with the metric learning algorithms introduced in [Xiang *et al.*, 2008; Xing *et al.*, 2003; Yeung and Chang, 2006], as they are the most effective methods considering both positive and negative constraints. We also include the LLMA algorithm [Chang and Yeung, 2006] in our evaluations. This algorithm is one of the most powerful metric learning methods that only use positive constraints (it provides a globally non-linear metric).

As in [Xing *et al.*, 2003; Yeung and Chang, 2006; Chang and Yeung, 2006], we use the Euclidean distance (without metric learning) for the baseline comparison and apply the *k*-means clustering algorithm with different distance metrics to compare these metrics. Thus, the performance of our non-linear metric learning algorithm is evaluated by comparing the following algorithms:

1. *k*-means without metric learning;
2. *k*-means with the metric learning method introduced in [Xiang *et al.*, 2008];
3. *k*-means with the LLMA [Chang and Yeung, 2006] method for metric learning;
4. *k*-means with the extended RCA [Yeung and Chang, 2006] method for metric learning;
5. *k*-means with the metric learning method introduced in [Xing *et al.*, 2003];
6. *k*-means with our non-linear metric learning method.

The parameters of our algorithm are set to $k = 10$, $d' = m/2$, and

$$\alpha = \begin{cases} 0.2 & d > 5 \\ 0.02 & d \le 5 \end{cases}. \qquad (15)$$

The kernel parameter w of our method and the Gaussian window parameter of the LLMA method are specified as $w = 2\beta\sum_{i<j}\|\mathbf{x}_i - \mathbf{x}_j\|/[n(n+1)]$ according to [Chang and Yeung, 2006] where we set $\beta = 1.5$ for our method. The number of nearest neighbors k of the LLMA has also been set to $k = 10$ and the parameter λ of this algorithm has been set to $\lambda = 0.2$ (this value provides better results than $\lambda = 5$ used in [Chang and Yeung, 2006]). For optimization in the LLMA algorithm, we used the spectral method. Finally, we set the reduced dimensionality of the method of [Xiang *et al.*, 2008] to half of the dimensionality of the input space [Xiang *et al.*, 2008].

As in [Yeung and Chang, 2006], we set $nc = |P| = |D|$ for methods that use both positive and negative constraints. Since results depend on P and/or D sets, we generate 20 different P and/or D sets for each data set. Finally, we run the *k*-means algorithm 20 times with different random initializations for each P and/or D set.

3.2 Performance Measure

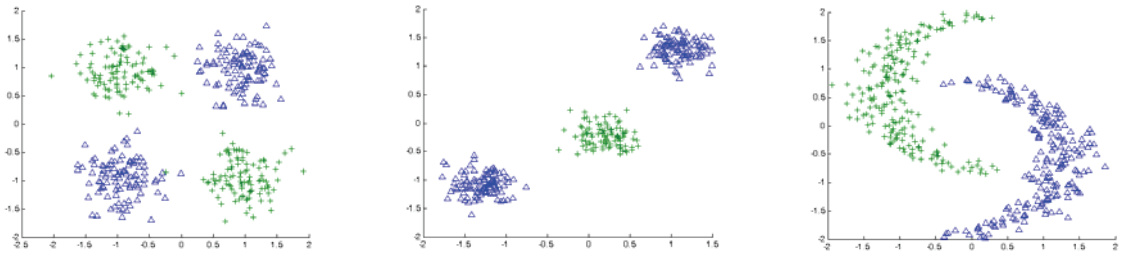
To evaluate the performance of clustering in our experiments, we employ the *Rand index* as the most widely used measure by semi-supervised metric learning algorithms. It shows how well the clustering results agree with the ground truth clusters [Chang and Yeung, 2006]. Let n_s be the number of data pairs 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 assigned to different clusters both in the ground truth and the resultant clustering (i.e., mismatched pairs). The Rand index is defined as $RI = 2(n_s + n_d)/(n(n-1))$ [Xing *et al.*, 2003] (n denotes the number of data points). This index favors assigning data points to different clusters when there are more than two clusters [Xing *et al.*, 2003]. Thus, we modify the Rand index as in [Xing *et al.*, 2003] such that the matched pairs and mismatched pairs are assigned weights to give them equal chances of occurrence (0.5) [Chang and Yeung, 2006].

3.3 Experiments on Synthetic and UCI Data Sets

At first, we conduct experiments on three synthetic data sets displayed in Figure 1. In this figure, the data points that belong to the same class are shown with the same style. Figure 2 shows the results of applying different algorithms on these data sets as box-plots ($nc = |P| = |D|$). Although the LLMA algorithm gives good results on two out of the three synthetic data sets, our method performs well on all of them.

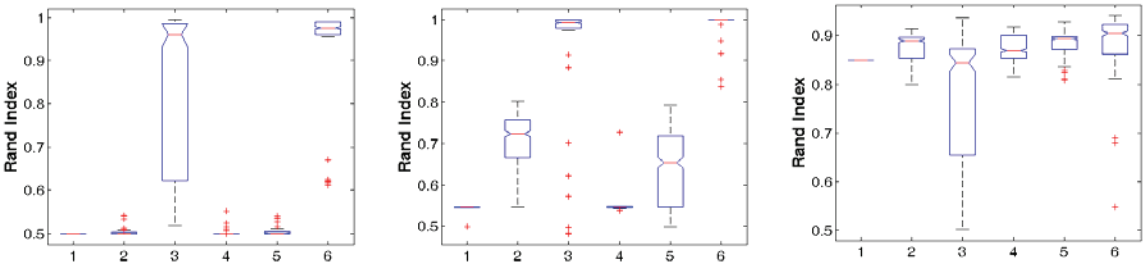
We then conduct further experiments on nine real-world data sets obtained from the Machine Learning Repository[2] of the University of California, Irvine (UCI): Soybean (47/35/4), Protein (116/20/6), Iris (150/4/3), Wine (178/13/3), Ionosphere (351/34/2), Boston housing (506/13/3), Breast cancer (569/31/2), Balance (625/4/3), and Diabetes (768/8/2). The numbers inside parentheses (n/d/c) show the number of data points n, the number of features d, and the number of classes c. Figure 3 shows the results of different algorithms on the nine UCI data sets. The number of constraints nc used for different data sets has been specified according to the numbers used in [Yeung and Chang, 2006; Chang and Yeung, 2006]. All of the data sets are normalized before use in the clustering

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



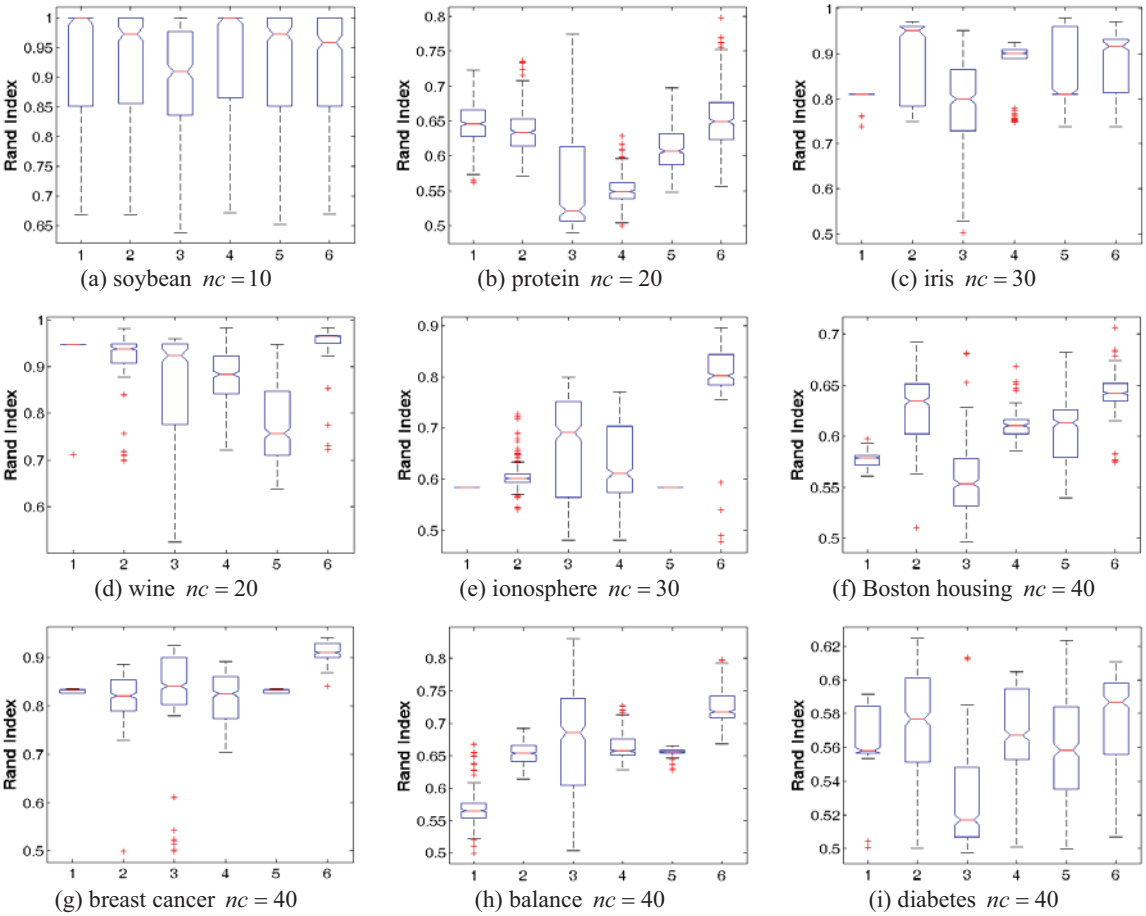
(a) data set 1 (b) data set 2 (c) data set 3

Figure 1. Synthetic data sets.



(a) data set 1, $nc = 5$ (b) data set 2, $nc = 10$ (c) data set 3, $nc = 15$

Figure 2. Clustering results on the three synthetic data sets using different metrics (numbered as in Section 3.1).



(a) soybean $nc = 10$ (b) protein $nc = 20$ (c) iris $nc = 30$

(d) wine $nc = 20$ (e) ionosphere $nc = 30$ (f) Boston housing $nc = 40$

(g) breast cancer $nc = 40$ (h) balance $nc = 40$ (i) diabetes $nc = 40$

Figure 3. Clustering results on the nine UCI data sets using different metrics (numbered as in Section 3.1).

algorithms (each feature is normalized to zero mean and unit standard deviation).

As we can see in Figure 3, our method generally provides better results than the other methods. By comparing the proposed method with [Xiang *et al.*, 2008], the most recent metric learning method, we find that our method is clearly better than it for six out of the nine data sets and comparable with it for two of the data sets.

4 Conclusions and Future Works

In this paper, we introduced a novel metric learning method for semi-supervised clustering. The existing metric learning methods that can use both positive and negative constraints have not incorporated the geometrical structure of data. Additionally, they only provide linear metrics. The proposed method uses the topological structure of data along with positive and negative constraints to find an appropriate metric. We proposed a kernel-based method to find a non-linear metric that is usually more useful than linear metrics. Experimental results on synthetic and UCI data sets showed the superior performance of our algorithm. In the future, we will investigate other forms of kernel-based metric learning methods. We also intend to evaluate the performance of our method on other real-world applications.

References

[Bar-Hillel *et al.*, 2005] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[Cai *et al.*, 2007] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV'07)*, pages 1-7, Brazil, 2007.

[Chang and Yeung, 2006] H. Chang and D.Y. Yeung. Locally linear metric adaptation with application to semi-supervised clustering and image retrieval. *Pattern Recognition*, 39:1253-1264, 2006.

[Chang *et al.*, 2006] H. Chang, D.Y. Yeung, and W.K. Cheung. Relaxational metric adaptation and its application to semi-supervised clustering and content-based image retrieval. *Pattern Recognition*, 39:1905-1917, 2006.

[Goldberger *et al.*, 2004] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighborhood components analysis. In *Advances in NIPS*, pages 513–520, MIT Press, Cambridge, MA, USA, 2004.

[Hastie and Tibshirani, 1996] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Analysis Machine Intelligence.*, 18(6):607–616, 1996.

[Hoi *et al.*, 2006] S.C.H. Hoi, W. Liu, M.R. Lyu, and W.Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2072-2078, New York, USA, 2006.

[Klein *et al.*, 2002] D. Klein, S.D. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: making the most of prior knowledge in data clustering. In *Proc. of the 19th Int. Conf. on Machine Learning (ICML-02)*, pages 307–314, Sydney, Australia, 2002.

[Kumar and Kummamuru, 2007] N. Kumar and K. Kummamuru. Semi-supervised clustering with metric learning using relative comparisons. *IEEE Trans. on Knowledge and Data engineering*, 20(4):496-503, 2007.

[Lebanon, 1994] G. Lebanon. Flexible metric nearest neighbor classification. Technical Report, Statistics Department, Stanford University, 1994.

[Roweis and Saul, 2000] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323-2326, 2000.

[Schultz and Joachims, 2004] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in NIPS*, pages 41-48, MIT Press, Cambridge, MA, USA, 2004.

[Xiang *et al.*, 2008] S. Xiang, F. Nie, and C. Zhang. Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 2008. doi: 10.1016/j.patcog.2008.05.018.

[Xing *et al.*, 2003] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. In *Advances in NIPS*, pages 505–512, MIT Press, Cambridge, MA, USA, 2003.

[Yang and Jin, 2006] L. Yang and R. Jin. Distance metric learning: a comprehensive survey. Technical Report, Michigan State University, 2006.

[Yeung and Chang, 2006] D.Y. Yeung and H. Chang. Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. *Pattern Recognition*, 39:1007-1010, 2006.

[Yeung and Chang, 2007] D.Y. Yeung and H. Chang. A Kernel Approach for semi-supervised metric learning. *IEEE Trans. on Neural Networks*, 18(1):141-149, 2007.

[Yeung *et al.*, 2007] D.Y. Yeung, H. Chang, and G. Dai. A Scalable Kernel-Based Algorithm for Semi-Supervised Metric Learning. In *Proceedings of IJCAI*, pages 1138-1143, 2007.

[Zhang *et al.*, 2003] Z.H. Zhang, J.T. Kwok, and D.Y. Yeung. Parametric distance metric learning with label information. In *Proceedings of IJCAI*, pages 1450–1452, Acapulco, Mexico, 2003.