A Kernel Approach for Semisupervised Metric Learning

Dit-Yan Yeung, *Member, IEEE*, and Hong Chang

***Abstract*—While distance function learning for supervised learning tasks has a long history, extending it to learning tasks with weaker supervisory information has only been studied recently. In particular, some methods have been proposed for semisupervised metric learning based on pairwise similarity or dissimilarity information. In this paper, we propose a kernel approach for semisupervised metric learning and present in detail two special cases of this kernel approach. The metric learning problem is thus formulated as an optimization problem for kernel learning. An attractive property of the optimization problem is that it is convex and, hence, has no local optima. While a closed-form solution exists for the first special case, the second case is solved using an iterative majorization procedure to estimate the optimal solution asymptotically. Experimental results based on both synthetic and real-world data show that this new kernel approach is promising for nonlinear metric learning.**

***Index Terms*—Clustering, kernel learning, metric learning, semisupervised learning.**

I. Introduction

DISTANCE functions or dissimilarity measures are central to many models and algorithms in machine learning, pattern recognition, and computer vision [14], [16], [28], [32]. Some common examples are nearest neighbor classifiers, radial basis function (RBF) networks, and support vector machines for classification (or supervised learning) tasks and the k-means algorithm for clustering (or unsupervised learning) tasks. The performance of these methods often depends critically on the choice of an appropriate distance function. Instead of predefining a distance function based on prior knowledge about the application at hand, a more appealing approach is to learn an appropriate distance function, possibly starting from some initial choice, based on supervisory information available about the application.

A. Distance Function Learning for Supervised Learning

For supervised learning applications such as classification and regression tasks, one can easily formulate the distance function learning problem as a well-defined optimization problem based on the supervisory information available in the training data. This approach has been pursued by many researchers. Early work taking this approach includes various metric learning methods for nearest neighbor classifiers, e.g., [18], [19], and [38]. More recent work includes [12], [13], [15], [17], [20], [21], [26], and [30].

B. Distance Function Learning for Other Learning Problems

It is natural to ask if distance function learning can also be applied to more difficult learning tasks. More specifically, we want to consider unsupervised learning applications such as clustering, dimensionality reduction, density estimation, and novelty detection. Unfortunately, under the unsupervised learning setting, the distance function learning problems are ill-posed with no well-defined optimization criteria. For example, using the same clustering algorithm (e.g., k-means) with different distance measures generally leads to different clustering results, but, without class label information, there is no ground truth against which different clustering results can be compared to make a choice. As an another example in the context of dimensionality reduction using methods such as principal component analysis (PCA) [23], a special form of distance learning which simply reweighs the features may end up turning a relevant dimension into irrelevant one and vice versa. Again, there does not exist any optimality criterion for us to formulate a well-defined optimization problem.

A more sensible territory to explore is the class of semisupervised learning problems [35]. Typically, in addition to the usually large quantity of unlabeled data, limited additional knowledge is also available to provide supervisory information that can be utilized for distance function learning. The supervisory information may be in the form of labeled data, which are typically limited in quantity. Strictly speaking, such problems may also be regarded as supervised learning tasks with only limited labeled data. For such problems, the classification accuracy can usually be improved with the aid of additional unlabeled data. Some methods that adopt this approach include [3], [39], and [48].

An arguably more challenging setting is when the supervisory information is given in a weaker form in terms of pairwise similarity or dissimilarity information. Very often, the pairwise information simply states whether two examples belong to the same class or different classes.[1] Wagstaff *et al.* [43], [44] first used such pairwise constraints for semisupervised clustering tasks by modifying the standard k-means clustering algorithm

Manuscript received May 25, 2005; revised May 4, 2006. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Competitive Earmarked Research Grant (CERG) HKUST6174/04E.

The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: dyyeung@cse.ust.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/TNN.2006.883723

[1] In principle, it is possible to incorporate pairwise information that quantifies the degree of similarity or dissimilarity as well as to provide more informative knowledge for distance function learning. The pairwise side information can be seen as part of the dissimilarity matrix in multidimensional scaling (MDS) problems.

to take into account pairwise similarity and dissimilarity. Extensions have also been made to model-based clustering based on the expectation–maximization (EM) algorithm for Gaussian mixture models [27], [37]. However, no distance function is explicitly learned in these methods. Some methods have been proposed for learning global Mahalanobis metrics and related distance functions from pairwise information [2], [4], [22], [33], [36], [45]. Xing *et al.* [45] proposed using pairwise constraints in a novel way to learn a global Mahalanobis metric before performing clustering with the constraints. Instead of using an iterative algorithm as in [45], Bar–Hillel *et al.* [2] devised a more efficient, noniterative algorithm called relevant component analysis (RCA) for learning a global Mahalanobis metric. We proposed a simple extension to RCA that allows both similarity and dissimilarity constraints to be incorporated [47]. Schultz and Joachims [33] made use of a different type of pairwise information which compares the pairwise constraints between two pairs of examples. More specifically, each relational constraint states that example A is closer to B than A is to C. Another distance function learning method is called DistBoost [22], which is based on boosting by incorporating pairwise constraints to learn a nonmetric distance function. However, the distance functions learned by these methods are either nonmetric or globally linear metrics. In [6], we generalized the globally linear metrics to a new metric that is linear locally but nonlinear globally. However, the criterion function of the optimization problem has local optima and the topology cannot be preserved well during metric learning.

C. Organization of the Paper

Along the same direction pursued in our previous work [6] to devise nonlinear extensions of linear metric learning methods, we propose in this paper a kernel approach for the learning of distance metrics based on pairwise similarity information. This essentially formulates metric learning as a kernel learning problem [1], [5], [10], [11], [24], [25], [29], [34], [40]–[42], [46], [49]–[51].

In Section II, we present a general kernel-based approach for the metric learning problem and then provide details of the optimization problems for two special cases. Section III presents some experiments based on both synthetic and real-world data to compare our kernel-based metric learning methods with some other methods. Finally, we give some concluding remarks in Section IV.

II. Our Kernel-Based Metric Learning Approach

Let $\mathbf{x}_i (i = 1, \ldots, n)$ denote n points in the input space $\mathcal{X}$. Suppose we use a kernel function k, such as RBF kernel or polynomial kernel, which induces a nonlinear mapping ϕ from $\mathcal{X}$ to some feature space $\mathcal{F}$. The images of the n points in $\mathcal{F}$ are $\phi(\mathbf{x}_i)(i = 1, \ldots, n)$ and the corresponding kernel matrix $\mathbf{K} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{n\times n} = [\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle]_{n\times n} = \mathbf{\Phi}\mathbf{\Phi}^T$, where $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)]^T$.

Since the kernel matrix $\mathbf{K}$ is symmetric and positive semidefinite, we can perform eigendecomposition on $\mathbf{K}$ to express it as

$$\mathbf{K} = \sum_{r=1}^{p} \xi_r \boldsymbol{\alpha}_r \boldsymbol{\alpha}_r^T \tag{1}$$

where $\xi_1 \geq \cdots \geq \xi_p > 0$ denote the $p \leq n$ positive eigenvalues of $\mathbf{K}$ and $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$ are the corresponding normalized eigenvectors.[2] Note that (1) may also be expressed as

$$\mathbf{K} = \sum_{r=1}^{p} \xi_r \mathbf{K}_r \tag{2}$$

where $\mathbf{K}_r = \boldsymbol{\alpha}_r \boldsymbol{\alpha}_r^T (r = 1, \ldots, p)$ are rank-one matrices. Using these base kernel matrices, we can define a parameterized family $\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}}$ of kernel matrices as

$$\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}} = \sum_{r=1}^{p} \beta_r^2 (\mathbf{A}\boldsymbol{\alpha}_r)(\mathbf{A}\boldsymbol{\alpha}_r)^T = \sum_{r=1}^{p} \beta_r^2 \mathbf{A}\mathbf{K}_r\mathbf{A}^T \tag{3}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ denotes p coefficients and $\mathbf{A}$ is an $n \times n$ matrix. It is easy to show that all matrices in the family are symmetric and positive semidefinite and, hence, the corresponding kernel functions are Mercer kernels or reproducing kernels [32]. While the use of $\boldsymbol{\beta}$ for defining a class of spectral variants of $\mathbf{K}$ is commonly found in other kernel learning work [5], [11], [25], [41], [50], we are not aware of other work that uses $\mathbf{A}$ for this purpose.

In Sections II-A and II-B, we consider kernel-based metric learning based on two special cases of the form in (3). The supervisory information available for metric learning is expressed as a set of point pairs $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are similar}\}$, which represents the pairwise similarity constraints. Note that the initial kernel matrix $\mathbf{K}$ is constructed from all n data points regardless of whether they are involved in the supervisory information (pairwise constraints), and $\mathbf{K}$ influences the final kernel matrix $\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}}$ learned. Thus, the kernel-based metric learning problem belongs to the semisupervised learning paradigm.

To facilitate our subsequent derivation, let us define indicator vectors $\mathbf{b}_i (i = 1, \ldots, n)$ where $\mathbf{b}_i$ is the ith column of the $n \times n$ identity matrix. The (i, j)th entry of $\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}}$ can then be expressed as

$$(\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}})_{ij} = \mathbf{b}_i^T \mathbf{K}_{\boldsymbol{\beta},\mathbf{A}} \mathbf{b}_j. \tag{4}$$

A. Case 1: Learning $\boldsymbol{\beta}$ Only

We first consider a special case which fixes $\mathbf{A}$ to the identity matrix and learns the coefficients $\boldsymbol{\beta}$ only. Hence, we have

$$\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}} = \mathbf{K}_{\boldsymbol{\beta}} = \sum_{r=1}^{p} \beta_r^2 \mathbf{K}_r. \tag{5}$$

[2]In practice, instead of choosing p to be the rank of $\mathbf{K}$, we usually approximate $\mathbf{K}$ by discarding those eigenvectors whose corresponding eigenvalues are very small in value.

Let $\psi(\mathbf{x}_i)(i = 1, \ldots, n)$ denote the n points in the feature space induced by $\mathbf{K}_{\boldsymbol{\beta}}$. Based on the set of pairwise similarity constraints $\mathcal{S}$, we define the following criterion function:

$$J_{\mathcal{S}}(\boldsymbol{\beta}) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2 \tag{6}$$

which is the total squared Euclidean distance between feature vectors in $\mathbf{K}_{\boldsymbol{\beta}}$ corresponding to point pairs in $\mathcal{S}$. The criterion function can be rewritten as

$$\begin{aligned} J_{\mathcal{S}}(\boldsymbol{\beta}) &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} [(\mathbf{K}_{\boldsymbol{\beta}})_{ii} + (\mathbf{K}_{\boldsymbol{\beta}})_{jj} - 2(\mathbf{K}_{\boldsymbol{\beta}})_{ij}] \\ &= \sum_{r=1}^{p} \beta_r^2 \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} [(\mathbf{K}_r)_{ii} + (\mathbf{K}_r)_{jj} - 2(\mathbf{K}_r)_{ij}] \\ &= \sum_{r=1}^{p} \beta_r^2 \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{K}_r (\mathbf{b}_i - \mathbf{b}_j) \\ &= \sum_{r=1}^{p} \beta_r^2 f_r \\ &= \boldsymbol{\beta}^T \mathbf{D}_{\mathcal{S}} \boldsymbol{\beta} \end{aligned} \tag{7}$$

where

$$\begin{aligned} f_r &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{K}_r (\mathbf{b}_i - \mathbf{b}_j) \\ &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \left[(\mathbf{b}_i - \mathbf{b}_j)^T \boldsymbol{\alpha}_r\right]^2 \geq 0 \end{aligned} \tag{8}$$

and

$$\mathbf{D}_{\mathcal{S}} = \operatorname{diag}(f_1, \ldots, f_p). \tag{9}$$

To prevent $\boldsymbol{\beta}$ from degenerating to the zero vector $\mathbf{0}$, we minimize the convex function $J_{\mathcal{S}}(\boldsymbol{\beta})$ subject to the linear constraint $\mathbf{1}^T\boldsymbol{\beta} = c$ for some constant $c > 0$.[3] The linear constraint eliminates the scale factor in the criterion function. This is a constrained optimization problem with an equality constraint, which can be solved by introducing a Lagrange multiplier ρ to minimize the following Lagrangian:

$$J(\boldsymbol{\beta}, \rho) = J_{\mathcal{S}}(\boldsymbol{\beta}) + \rho(c - \mathbf{1}^T\boldsymbol{\beta}). \tag{10}$$

We then compute the partial derivatives

$$\frac{\partial J}{\partial \boldsymbol{\beta}} = 2\mathbf{D}_{\mathcal{S}}\boldsymbol{\beta} - \rho\mathbf{1} \tag{11}$$

$$\frac{\partial J}{\partial \rho} = c - \mathbf{1}^T\boldsymbol{\beta}. \tag{12}$$

Setting $\partial J/\partial\boldsymbol{\beta} = \mathbf{0}$ and $\partial J/\partial\rho = 0$, we can obtain the optimal value of $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \frac{c\mathbf{D}_{\mathcal{S}}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{D}_{\mathcal{S}}^{-1}\mathbf{1}}. \tag{13}$$

[3]Bousquet and Herrmann [5] and Lanckriet *et al.* [25] set $\operatorname{Tr}(\mathbf{K}_{\boldsymbol{\beta}}) = c$ as constraint which is equivalent to $\boldsymbol{\beta}^T\boldsymbol{\beta} = c$. However, we use a constraint that is linear in $\boldsymbol{\beta}$ so that the constrained optimization problem will not lead to a value of $\mathbf{0}$ for $\boldsymbol{\beta}$.

The constant c in the constraint is set to $\sum_{r=1}^{p}\sqrt{\xi_r}$. Since $\mathbf{D}_{\mathcal{S}}$ is a diagonal matrix, $\mathbf{D}_{\mathcal{S}}^{-1}$ exists as long as all the diagonal entries are nonzero.

B. Case 2: Learning $\mathbf{A}$ Only

As another special case, we fix the coefficients $\boldsymbol{\beta}$ and learn $\mathbf{A}$ only. Specifically, we set $\beta_k^2 = \xi_k(k = 1, \ldots, p)$. Hence

$$\mathbf{K}_{\boldsymbol{\beta},\mathbf{A}} = \mathbf{K}_{\mathbf{A}} = \mathbf{A}\mathbf{K}\mathbf{A}^T. \tag{14}$$

A major advantage of this method is that no eigendecomposition of $\mathbf{K}$ is needed.

Based on $\mathcal{S}$, we define the following criterion:

$$J_{\mathcal{S}}(\mathbf{A}) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\|^2 = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} q_{ij}^2(\mathbf{A}) \tag{15}$$

where $q_{ij}(\mathbf{A})$ denotes the Euclidean distance between $\psi(\mathbf{x}_i)$ and $\psi(\mathbf{x}_j)$ in the feature space induced by $\mathbf{K}_{\mathbf{A}}$, with its dependency on $\mathbf{A}$ explicitly shown.

Unlike the previous case which learns relatively few parameters in the p-dimensional vector $\boldsymbol{\beta}$, here we need to learn all the entries of the $n \times n$ matrix $\mathbf{A}$ where n is typically much larger than p. To impose stronger capacity control to restrict the search space, we introduce a regularization term to constrain the degree of transformation that $\mathbf{A}$ can bring about. While minimizing the term $J_{\mathcal{S}}(\mathbf{A})$ tends to pull the points together, the regularization term tries to go against this trend by limiting the degree of deformation from the initial positions of the feature vectors. Specifically, the regularization term is as follows:

$$J_{\mathcal{C}}(\mathbf{A}) = \sum_{i,j=1}^{n} \mathcal{N}_{\sigma}(c_{ij}) \left(\|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\| - c_{ij}\right)^2 \tag{16}$$

where $c_{ij} = \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\| = \sqrt{(\mathbf{K})_{ii} + (\mathbf{K})_{jj} - 2(\mathbf{K})_{ij}}$ is the initial Euclidean distance between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ before metric learning, and $\mathcal{N}_{\sigma}(\cdot)$ is a Gaussian function with $\mathcal{N}_{\sigma}(a) = \exp(-a^2/\sigma^2)$ for some parameter $\sigma > 0$ that specifies the spread of the Gaussian window. The squared term penalizes deviation from the original interpoint distance c_{ij} and the Gaussian weight regulates the degree of penalty by taking into consideration the magnitude of c_{ij}. This form of the regularization term is similar to that used in locally linear metric adaptation (LLMA) [6].

Metric learning is formulated as an unconstrained optimization problem by minimizing

$$J(\mathbf{A}) = J_{\mathcal{S}}(\mathbf{A}) + \rho J_{\mathcal{C}}(\mathbf{A}) \tag{17}$$

where $\rho > 0$ is a regularization parameter that adjusts the relative strength of the regularization term.

Let $s_{ij}(i, j = 1, \ldots, n)$ be defined such that $s_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ and 0, otherwise. We can rewrite $J(\mathbf{A})$ as

$$\begin{aligned} J(\mathbf{A}) &= \sum_{i,j=1}^{n} (s_{ij} + \rho\mathcal{N}_{\sigma}(c_{ij})) \left(q_{ij}(\mathbf{A}) - \frac{\rho\mathcal{N}_{\sigma}(c_{ij})c_{ij}}{s_{ij} + \rho\mathcal{N}_{\sigma}(c_{ij})}\right)^2 + C \\ &= \sum_{i,j=1}^{n} \gamma_{ij} \left(q_{ij}(\mathbf{A}) - p_{ij}\right)^2 + C \end{aligned} \tag{18}$$

where

$$\gamma_{ij} = s_{ij} + \rho \mathcal{N}_\sigma(c_{ij}) \tag{19}$$

$$p_{ij} = \frac{\rho \mathcal{N}_\sigma(c_{ij}) c_{ij}}{s_{ij} + \rho \mathcal{N}_\sigma(c_{ij})} \tag{20}$$

and C is a term that does not depend on $\mathbf{A}$. Hence, the optimal value of $\mathbf{A}$ that minimizes $J(\mathbf{A})$ also minimizes $\hat{J}(\mathbf{A}) = \sum_{i,j=1}^{n} \gamma_{ij}(q_{ij}(\mathbf{A}) - p_{ij})^2$. As we can see from (21), $q_{ij}^2(\mathbf{A})$ is quadratic in $\mathbf{A}$ and $q_{ij}(\mathbf{A}) = ||\mathbf{\Phi}^T \mathbf{A}^T(\mathbf{b}_i - \mathbf{b}_j)||$. Since $\mathbf{\Phi}^T \mathbf{A}^T(\mathbf{b}_i - \mathbf{b}_j)$ is linear in $\mathbf{A}$ and the norm function is convex, we can conclude that $q_{ij}(\mathbf{A})$ is convex in $\mathbf{A}$. By incorporating (18), we know that $J(\mathbf{A})$ and $\hat{J}(\mathbf{A})$ are also convex in $\mathbf{A}$. As in [6], we use the method of *iterative majorization* to find the optimal value of $\mathbf{A}$. This method is guaranteed to find the optimal solution asymptotically since the criterion function $J(\mathbf{A})$ is convex.

Note that

$$\begin{aligned} q_{ij}^2(\mathbf{A}) &= ||\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)||^2 \\ &= (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{K}_\mathbf{A}(\mathbf{b}_i - \mathbf{b}_j) \\ &= (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{A}\mathbf{\Phi}\mathbf{\Phi}^T \mathbf{A}^T(\mathbf{b}_i - \mathbf{b}_j) \end{aligned} \tag{21}$$

so $q_{ij}(\mathbf{A}) = ||\mathbf{\Phi}^T \mathbf{A}^T(\mathbf{b}_i - \mathbf{b}_j)||$. Similarly, we define $q_{ij}(\mathbf{B}) = ||\mathbf{\Phi}^T \mathbf{B}^T(\mathbf{b}_i - \mathbf{b}_j)||$. From the Cauchy–Schwarz inequality, we have

$$\begin{aligned} q_{ij}(\mathbf{A}) q_{ij}(\mathbf{B}) &\geq (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{A}\mathbf{\Phi}\mathbf{\Phi}^T \mathbf{B}^T(\mathbf{b}_i - \mathbf{b}_j) \\ &= (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{A}\mathbf{K}\mathbf{B}^T(\mathbf{b}_i - \mathbf{b}_j). \end{aligned} \tag{22}$$

Expanding $\hat{J}(\mathbf{A})$ gives

$$\hat{J}(\mathbf{A}) = \sum_{i,j=1}^{n} \gamma_{ij} q_{ij}^2(\mathbf{A}) - 2 \sum_{i,j=1}^{n} \gamma_{ij} p_{ij} q_{ij}(\mathbf{A}) + \sum_{i,j=1}^{n} \gamma_{ij} p_{ij}^2. \tag{23}$$

The first term on the right-hand side of (23) can be rewritten as

$$\begin{aligned} \sum_{i,j=1}^{n} \gamma_{ij} q_{ij}^2(\mathbf{A}) &= \sum_{i,j=1}^{n} \gamma_{ij} (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{A}\mathbf{K}\mathbf{A}^T(\mathbf{b}_i - \mathbf{b}_j) \\ &= \mathrm{Tr}(\mathbf{A}\mathbf{K}\mathbf{A}^T\mathbf{E}) \end{aligned} \tag{24}$$

where

$$\mathbf{E} = \sum_{i,j=1}^{n} \gamma_{ij} (\mathbf{b}_i - \mathbf{b}_j)(\mathbf{b}_i - \mathbf{b}_j)^T. \tag{25}$$

The second term on the right-hand side of (23) may be rewritten as the sum of two terms $\sum_{(i,j)\in H_+} \gamma_{ij} p_{ij} q_{ij}(\mathbf{A})$ and $\sum_{(i,j)\in H_0} \gamma_{ij} p_{ij} q_{ij}(\mathbf{A})$, with H_+ representing the set of all point pairs (i,j) for which $q_{ij}(\mathbf{B}) > 0$ and H_0 the set of all point pairs (i,j) for which $q_{ij}(\mathbf{B}) = 0$. For $(i,j) \in H_+$, by (22), we have

$$\gamma_{ij} p_{ij} q_{ij}(\mathbf{A}) \geq \frac{\gamma_{ij} p_{ij}}{q_{ij}(\mathbf{B})} (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{A}\mathbf{K}\mathbf{B}^T(\mathbf{b}_i - \mathbf{b}_j). \tag{26}$$

For $(i,j) \in H_0$, since $\gamma_{ij} > 0$, $p_{ij} \geq 0$, and $q_{ij}(\mathbf{A}) \geq 0$, we have

$$\gamma_{ij} p_{ij} q_{ij}(\mathbf{A}) \geq 0. \tag{27}$$

Combining (26) and (27), the second term on the right-hand side of (23) can be expressed as the following:

$$\begin{aligned} -2 \sum_{i,j=1}^{n} \gamma_{ij} p_{ij} q_{ij}(\mathbf{A}) &\leq -2 \sum_{i,j=1}^{n} f_{ij}(\mathbf{B}) (\mathbf{b}_i - \mathbf{b}_j)^T \mathbf{A}\mathbf{K}\mathbf{B}^T \\ &\quad \cdot (\mathbf{b}_i - \mathbf{b}_j) \\ &= -2\mathrm{Tr}\left(\mathbf{A}\mathbf{K}\mathbf{B}^T\mathbf{F}(\mathbf{B})\right) \end{aligned} \tag{28}$$

where

$$f_{ij}(\mathbf{B}) = \begin{cases} \frac{\rho \mathcal{N}_\sigma(c_{ij}) c_{ij}}{q_{ij}(\mathbf{B})}, & q_{ij}(\mathbf{B}) > 0 \\ 0, & q_{ij}(\mathbf{B}) = 0 \end{cases} \tag{29}$$

$$\mathbf{F}(\mathbf{B}) = \sum_{i,j=1}^{n} f_{ij}(\mathbf{B})(\mathbf{b}_i - \mathbf{b}_j)(\mathbf{b}_i - \mathbf{b}_j)^T. \tag{30}$$

From (23), (24), and (28), we can obtain an upper bound on $\hat{J}(\mathbf{A})$ as

$$\begin{aligned} \hat{J}(\mathbf{A}) \leq \hat{J}(\mathbf{A}, \mathbf{B}) &= \mathrm{Tr}(\mathbf{A}\mathbf{K}\mathbf{A}^T\mathbf{E}) - 2\mathrm{Tr}\left(\mathbf{A}\mathbf{K}\mathbf{B}^T\mathbf{F}(\mathbf{B})\right) \\ &\quad + \sum_{i,j=1}^{n} \gamma_{ij} p_{ij}^2. \end{aligned} \tag{31}$$

Note that the equality holds, i.e., $\hat{J}(\mathbf{A}) = \hat{J}(\mathbf{A}, \mathbf{B})$, when $\mathbf{B} = \mathbf{A}$. In the method of iterative majorization, $\hat{J}(\mathbf{A})$ is called the *majorized function* and $\hat{J}(\mathbf{A}, \mathbf{B})$ is called the *majorizing function*. By setting the partial derivative $(\partial \hat{J}(\mathbf{A}, \mathbf{B}))/\partial \mathbf{A}$ to a zero matrix of the same dimension as $\mathbf{A}$, we can see that the optimal value of $\mathbf{A}$ that minimizes $\hat{J}(\mathbf{A}, \mathbf{B})$ should satisfy

$$\mathbf{E}\mathbf{A} = \mathbf{F}(\mathbf{B})\mathbf{B} \tag{32}$$

or

$$\mathbf{A} = \mathbf{E}^+\mathbf{F}(\mathbf{B})\mathbf{B} \tag{33}$$

where $\mathbf{E}^+$ is the pseudoinverse of $\mathbf{E}$. Thus, we can use an iterative procedure based on the following update equation to estimate the optimal value of $\mathbf{A}$ in a stepwise manner:

$$\mathbf{A}^{(t)} = \mathbf{E}^+\mathbf{F}\left(\mathbf{A}^{(t-1)}\right)\mathbf{A}^{(t-1)} \tag{34}$$

where $\mathbf{A}^{(t)}$ denotes the estimate at step t.

The iterative majorization procedure can be summarized as the following steps:

1) $t = 0$, $\mathbf{A}^{(0)} = \mathbf{I}$;
2) $t = t + 1$, compute $\mathbf{A}^{(t)} = \mathbf{E}^+\mathbf{F}\left(\mathbf{A}^{(t-1)}\right)\mathbf{A}^{(t-1)}$;
3) if converged, then stop. Otherwise, repeat from step 2).

Note that $\hat{J}(\mathbf{A})$ decreases over time monotonically since $\hat{J}(\mathbf{A}^{(t)}) \leq \hat{J}(\mathbf{A}^{(t)}, \mathbf{A}^{(t-1)}) \leq \hat{J}(\mathbf{A}^{(t-1)}, \mathbf{A}^{(t-1)}) = \hat{J}(\mathbf{A}^{(t-1)})$.

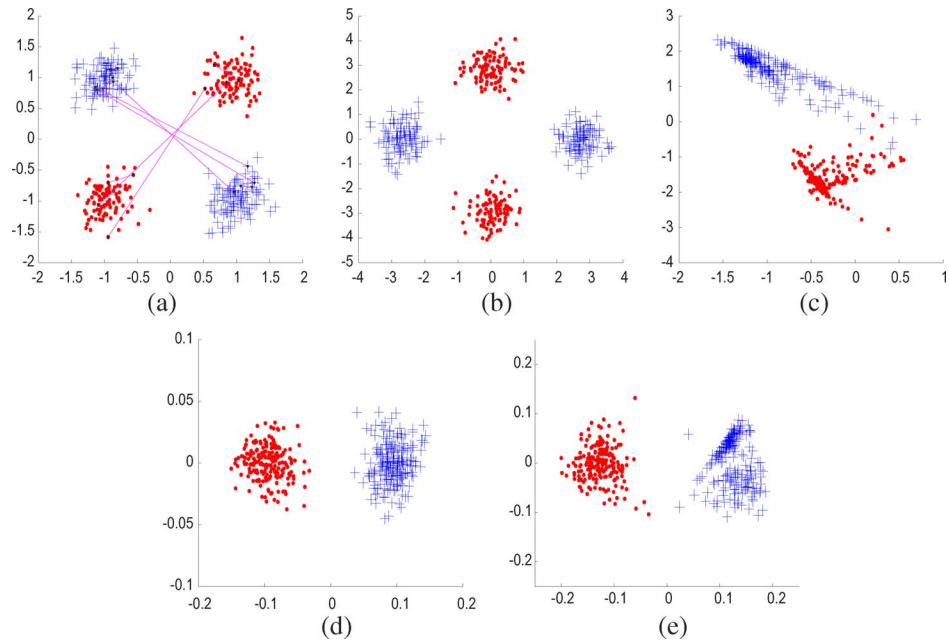

Fig. 1. Comparison of different metric learning methods on the XOR data set. (a) Original data set with two classes; and the data set after applying (b) RCA; (c) LLMA; (d) kernel-$\boldsymbol{\beta}$; and (e) kernel-$\mathbf{A}$.

III. Experiments

In this section, we describe some experiments we have performed based on both synthetic and real-world data to compare our kernel-based metric learning methods with some previous methods. We measure the effectiveness of a metric learning scheme indirectly by how much it can improve the clustering results in semisupervised clustering tasks with pairwise similarity constraints.

A. Experimental Setup

We compare the two kernel-based metric learning methods described in Sections II with some previous methods. The first method is RCA [2] which performs globally linear transformation in the input space. The RCA algorithm performs whitening transformation on the data set, which assigns lower weights to the "irrelevant" directions in the original feature space. The second method, called metric pairwise constrained k-means (MPCK-means) , unifies metric learning and pairwise constraints [4].[4] As in their experiments, a single metric parameterized by a diagonal matrix for all clusters is learned during k-means clustering. Since their method can make use of both similarity and dissimilarity information, we perform experiments in two different settings, without or with dissimilarity constraints. The number of dissimilarity constraints used by MPCK-means-$\mathcal{SD}$ is set to be the same as the number of similarity constraints. Another method is LLMA [6] which is more general in that it is linear locally but nonlinear globally. We also use the Euclidean distance without metric learning for baseline comparison. Since both RCA and LLMA make use of pairwise similarity constraints only, we also use such supervisory information only for our methods. In summary, the following seven distance measures for the k-means clustering algorithm are included in our comparative study (the short forms inside brackets will be used subsequently):

1) k-means without metric learning (Euclidean);
2) k-means with RCA for metric learning (RCA);
3) metric pairwise constrained k-means using similarity constraints (MPCK-means-$\mathcal{S}$);
4) metric pairwise constrained k-means using both similarity and dissimilarity constraints (MPCK-means-$\mathcal{SD}$);
5) k-means with LLMA for metric learning (LLMA);
6) k-means with our kernel-based metric learning method based on learning $\boldsymbol{\beta}$ (kernel-$\boldsymbol{\beta}$);
7) k-means with our kernel-based metric learning method based on learning $\mathbf{A}$ (kernel-$\mathbf{A}$).

We use RBF kernel for the initial kernel for our kernel-based metric learning methods. As in [2], [6], and [45], we use the rand index [31] as the clustering performance measure. The rand index reflects the agreement of the clustering result with the ground truth. Let n_s be the number of pattern pairs that are assigned to the same cluster (i.e., matched pairs) in both the resultant partition and the ground truth, and n_d be the number of pattern pairs that are assigned to different clusters (i.e., mismatched pairs) in both the resultant partition and the ground truth. The rand index is defined as the ratio of (n_s+n_d) to the total number of pattern pairs, i.e., $n(n-1)=2$. When there are more than two clusters, however, the standard rand index will favor assigning patterns to different clusters. We modify the rand index as in [45], so that matched pairs and mismatched pairs are assigned weights to give them equal chance of occurrence (0.5). For each data set, we randomly generate 20 different $\mathcal{S}$ sets to provide pairwise similarity constraints. In addition, for each $\mathcal{S}$ set, we perform 20 runs of k-means with different random initializations and report the average rand index over the 20 runs.

The two parameters used in our kernel-based metric learning methods are easy to set based on their physical meanings. As

[4]The Java code for MPCK-means was obtained from the authors of [4].

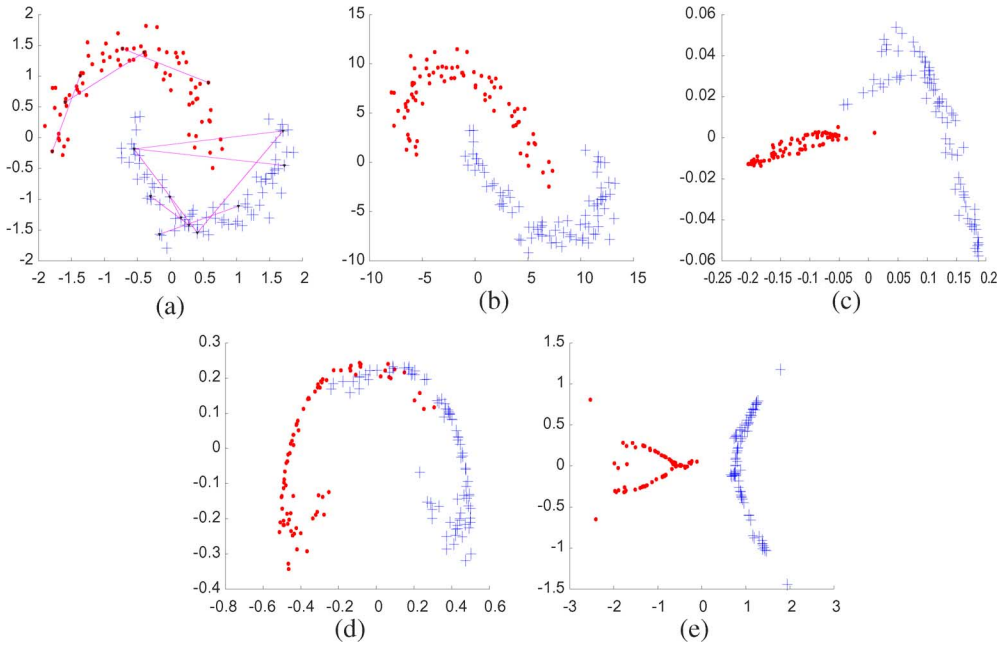

Fig. 2. Comparison of different metric learning methods on the two-moon data set. (a) Original data set with two classes; and the data set after applying (b) RCA; (c) LLMA; (d) kernel-$\boldsymbol{\beta}$; and (e) kernel-$\mathbf{A}$.

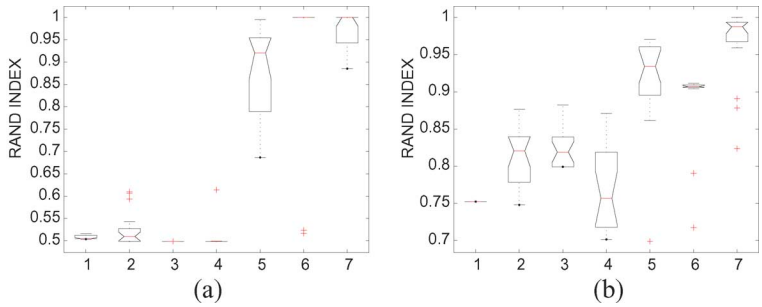

Fig. 3. Semisupervised clustering results: (a) XOR data set, and (b) two-moon data set. The seven clustering algorithms (numbered in Section III-A) are as follows: 1) Euclidean, 2) RCA, 3) MPCK-means-$\mathcal{S}$, 4) MPCK-means-$\mathcal{SD}$, 5) LLMA, 6) kernel-$\boldsymbol{\beta}$, and 7) kernel-$\mathbf{A}$.

for the Gaussian window parameter σ used in the regularization term (16), we make it depend on the average squared Euclidean distance between all point pairs in the feature space: $\sigma^2 = (\theta/n^2)\sum_{i,j=1}^{n} \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = (2\theta/n)[\mathrm{Tr}(\mathbf{K}) - n\overline{\mathbf{K}}]$, where $\overline{\mathbf{K}}$ represents the mean value of the elements in $\mathbf{K}$ and θ is set to be the same $(= 5)$ for all data sets. The regularization parameter ρ in the kernel-$\mathbf{A}$ method (17) is set to $[1, 3]$ in our experiments.

B. Experiments on Synthetic Data

We first perform some experiments on the XOR data set, as shown in Fig. 1(a). Data points shown with the same point style and color belong to the same class. Point pairs in $\mathcal{S}$ are connected by solid lines. Both RCA and LLMA perform metric learning directly in the input space. The transformed data set using RCA and LLMA is shown in Fig. 1(b) and (c), respectively. For our kernel-based methods, there is no need to embed the points in the feature space first before performing clustering. However, for the sake of visualization, we apply kernel PCA based on the learned kernel matrix to embed the points in a two-dimensional (2-D) space, as shown in Fig. 1(d) and (e). Obviously, RCA, which performs globally linear metric learning, cannot give satisfactory result. The performance of LLMA is significantly better, although some points from the two classes are quite close to each other. On the other hand, our kernel-based methods can not only group the data points according to their class but can also preserve the topology of the points inside clusters.

We also try the two-moon data set which is commonly used in some recent semisupervised learning research. However, the difference is that we do not exploit the underlying manifold structure here. Instead, only some limited pairwise similarity constraints are provided. The results in Fig. 2 again show that the kernel-based methods can give promising results.

We further perform some semisupervised clustering experiments on the XOR and two-moon data sets. We also include the clustering results of MPCK-means-$\mathcal{S}$ and MPCK-means-$\mathcal{SD}$ for comparison. The results are shown in Fig. 3. For each trial, ten-point pairs are randomly selected to form $\mathcal{S}$.

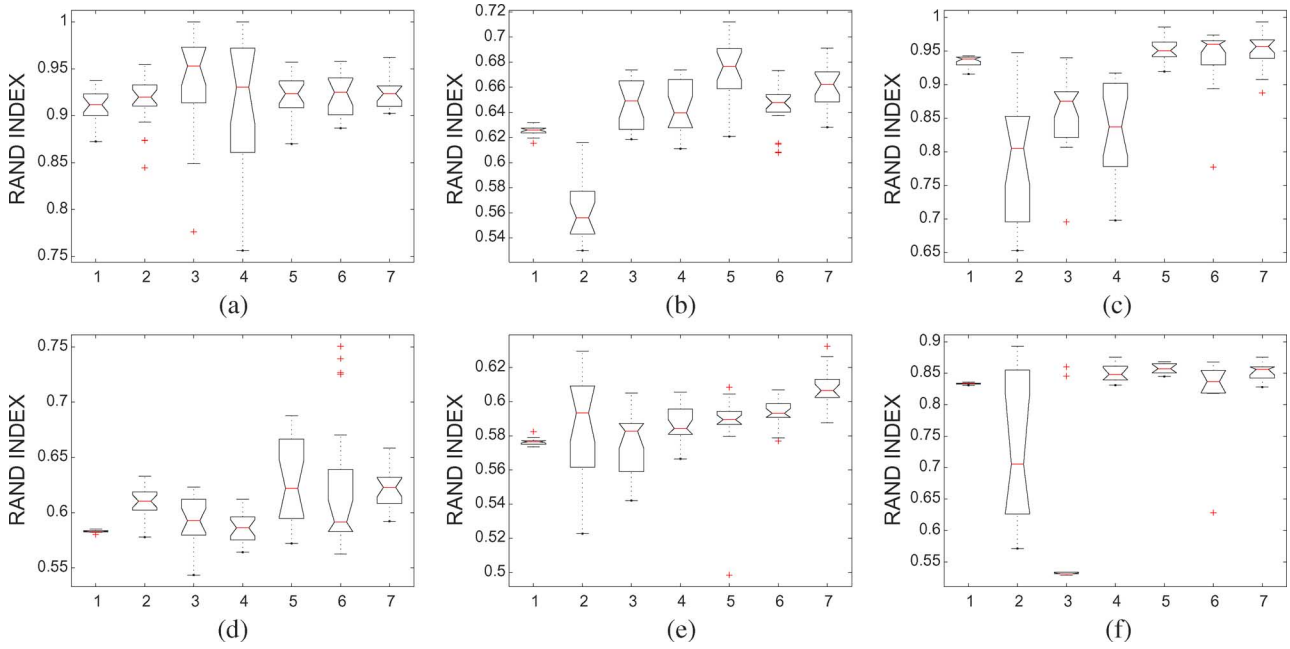

Fig. 4. Clustering results for six UCI data sets. The seven clustering algorithms (numbered in Section III-A) are as follows: 1) Euclidean, 2) RCA, 3) MPCK-means-$\mathcal{S}$, 4) MPCK-means-$\mathcal{SD}$, 5) LLMA, 6) kernel-$\boldsymbol{\beta}$, and 7) kernel-$\mathbf{A}$. (a) Soybean. (b) Protein. (c) Wine. (d) Ionosphere. (e) Boston housing. (f) Breast cancer.

TABLE I
SIX UCI DATA SETS USED IN THE EXPERIMENTS

DATA SET	n	d	m	$\lvert\mathcal{S}\rvert$
SOYBEAN	47	35	4	10
PROTEIN	116	20	6	15
WINE	178	13	3	20
IONOSPHERE	351	34	2	30
BOSTON HOUSING	506	13	3	40
BREAST CANCER	569	31	2	50

C. Experiments on UCI Data

We perform more semisupervised clustering experiments on six real-world data sets from the University of California at Irvine (UCI) Machine Learning Repository. Table I shows some characteristics of the data sets. The number of data points n, the number of features d, the number of clusters m, and the number of randomly selected point pairs $\lvert\mathcal{S}\rvert$ are shown for each data set in Table I.

Fig. 4 shows the clustering results based on k-means using different distance measures as numbered in Section III-A. The k-means algorithm with RCA for metric learning can sometimes improve the clustering results without metric learning. However, MPCK-means, LLMA and our kernel-based methods generally outperform RCA. For more accurate comparison, we perform paired t-test with significance level 0.05 to statistically evaluate which result is better. The comparison results are summarized in Table II. We use $\sim$ to indicate that the clustering results of the two methods are not significantly different for the given confidence level, and $<$ to indicate that the mean of the rand index values of the latter method is statistically higher than that of the former one. From the paired t-test results, we can conclude with a 95% confidence level that the kernel-based methods generally outperform MPCK-means and are comparable with or even better than LLMA. Here, we use MPCK-means to represent the better clustering results between Algorithms 3 and 4 (without and with dissimilarity constraints). As we can see from Fig. 4, the MPCK-means method with dissimilarity constraints incorporated cannot always improve the clustering results.

D. Experiments on MNIST Digits

We further perform some experiments on handwritten digits from the MNIST database.[5] The digits in the database have been size-normalized and centered to 28×28 gray-level images. Hence, the dimensionality of the input space is 784. In our experiments, we randomly choose 1000 images for each digit from a total of 60 000 digit images in the MNIST training set. We randomly select 50 similarity constraints to form an $\mathcal{S}$ set. Table III shows the results of different clustering algorithms for three-digit subsets. For each algorithm, we show the mean rand index (upper) and standard deviation (lower) over ten random runs corresponding to different $\mathcal{S}$ sets. From the results, we can see that the metric learned by our kernel-based methods gives the best clustering results.

While the kernel-$\boldsymbol{\beta}$ algorithm is efficient due to its closed-form solution, the optimization problem defined for the kernel-$\mathbf{A}$ algorithm is solved in an iterative manner. In our experiments, we use the maximum number of iterations (two for all data sets) as the stopping criterion for the iterative majorization procedure in the kernel-$\mathbf{A}$ algorithm. Fast convergence is observed in all cases and, hence, the number of iterations can be set to a very small number. In general, our kernel-based metric learning methods are slower than the global metric learning methods (RCA and MPCK-means) but are significantly faster than the nonlinear metric learning method (LLMA).

[5]http://yann.lecun.com/exdb/mnist/

TABLE II
PAIRED t-TEST FOR STATISTICAL EVALUATION OF THE CLUSTERING RESULTS

DATA SET	PAIRED t-TEST
SOYBEAN	MPCK-MEANS $\sim$ LLMA $\sim$ KERNEL-$\boldsymbol{\beta}$ $\sim$ KERNEL-$\mathbf{A}$
PROTEIN	MPCK-MEANS $\sim$ KERNEL-$\boldsymbol{\beta}$ $<$ KERNEL-$\mathbf{A}$ $<$ LLMA
WINE	MPCK-MEANS $<$ LLMA $<$ KERNEL-$\boldsymbol{\beta}$ $\sim$ KERNEL-$\mathbf{A}$
IONOSPHERE	MPCK-MEANS $<$ LLMA $\sim$ KERNEL-$\boldsymbol{\beta}$ $\sim$ KERNEL-$\mathbf{A}$
BOSTON HOUSING	MPCK-MEANS $\sim$ LLMA $\sim$ KERNEL-$\boldsymbol{\beta}$ $<$ KERNEL-$\mathbf{A}$
BREAST CANCER	MPCK-MEANS $\sim$ LLMA $\sim$ KERNEL-$\boldsymbol{\beta}$ $\sim$ KERNEL-$\mathbf{A}$

TABLE III
CLUSTERING RESULTS FOR MNIST DATA SETS

	EUCLIDEAN	RCA	MPCK-MEANS-$\mathcal{S}$	MPCK-MEANS-$\mathcal{SD}$	LLMA	KERNEL-$\boldsymbol{\beta}$	KERNEL-$\mathbf{A}$
$\{0,1\}$	0.9790	0.9814	0.9752	0.9800	0.9802	0.9896	**0.9900**
	$\pm$0.0004	$\pm$0.0109	$\pm$0.0105	$\pm$0.0055	$\pm$0.0015	$\pm$0.0009	$\pm$0.0011
$\{1,5\}$	0.8179	0.8410	0.8254	0.8156	0.8013	**0.8682**	0.8590
	$\pm$0.0001	$\pm$0.0211	$\pm$0.0075	$\pm$0.0068	$\pm$0.1370	$\pm$0.0534	$\pm$0.0089
$\{1,9\}$	0.9531	0.9546	0.9275	0.9317	0.9527	0.9609	**0.9657**
	$\pm$0.0000	$\pm$0.0319	$\pm$0.0137	$\pm$0.0087	$\pm$0.0012	$\pm$0.0203	$\pm$0.0068

IV. CONCLUDING REMARKS

We have proposed two kernel-based metric learning methods and demonstrated their promising performance over some existing linear and nonlinear methods. While the two kernel-based metric learning methods are quite effective, they do have some limitations. For the kernel-$\boldsymbol{\beta}$ method, the limitation is its need for performing eigendecomposition of the kernel matrix $\mathbf{K}$, which may lead to high computational demand when $\mathbf{K}$ is large. For the kernel-$\mathbf{A}$ method, it is not necessary to do eigencomposition of $\mathbf{K}$. However, learning $\mathbf{A}$ involves more parameters, which require stronger bias when the supervisory information is limited. One possible extension is to consider a smaller, nonsquare $\mathbf{A}$ matrix which essentially represents the n points by a smaller set of points. An interesting direction to explore is to devise a general scheme for learning both $\boldsymbol{\beta}$ and $\mathbf{A}$ simultaneously. As another direction, we will incorporate dissimilarity constraints into the methods to further improve the metric learning performance. Moreover, we will explore the application of the proposed methods to other real-world problems such as content-based image retrieval [7]–[9].

REFERENCES

[1] F. R. Bach, R. Thibaux, and M. I. Jordan, "Computing regularization paths for learning multiple kernels," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 73–80.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proc. 20th Int. Conf. Mach. Learn.*, Washington, DC, Aug. 21–24, 2003, pp. 11–18.

[3] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *Proc. 19th Int. Conf. Mach. Learn.*, Sydney, Australia, Jul. 8–12, 2002, pp. 19–26.

[4] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 4–8, 2004, pp. 81–88.

[5] O. Bousquet and D. J. L. Herrmann, "On the complexity of learning the kernel matrix," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 399–406.

[6] H. Chang and D. Y. Yeung, "Locally linear metric adaptation for semisupervised clustering," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 4–8, 2004, pp. 153–160.

[7] ——, "Kernel-based distance metric learning for content-based image retrieval," *Image Vis. Comput.*, 2006, to be published.

[8] ——, "Locally linear metric adaptation with application to semi-supervised clustering and image retrieval," *Pattern Recognit.*, vol. 39, no. 7, pp. 1253–1264, 2006.

[9] 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 Recognit.*, vol. 39, no. 10, pp. 1905–1917, 2006.

[10] K. Crammer, J. Keshet, and Y. Singer, "Kernel design using boosting," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 537–544.

[11] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 367–373.

[12] C. Domeniconi and D. Gunopulos, "Adaptive nearest neighbor classification using support vector machines," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 665–672.

[13] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest-neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1285, Sep. 2002.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[15] M. Fink, "Object classification from a single example utilizing class relevance metrics," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 449–456.

[16] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall, 2003.

[17] J. H. Friedman, "Flexible Metric Nearest Neighbor Classification" Dept. Statistics, Stanford Univ., Stanford, CA, Tech. Rep., November 1994.

[18] K. Fukunaga and T. E. Flick, "An optimal global nearest neighbor metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 3, pp. 314–318, Mar. 1984.

[19] K. Fukunaga and L. Hostetler, "Optimization of k-nearest neighbor density estimates," *IEEE Trans. Inf. Theory*, vol. 19, no. 3, pp. 320–326, May 1973.

[20] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighborhood components analysis," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 513–520.

[21] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.

[22] T. Hertz, A. Bar-Hillel, and D. Weinshall, "Boosting margin based distance functions for clustering," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 4–8, 2004, pp. 393–400.

[23] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York: Springer-Verlag, 2002.

[24] J. T. Kwok and I. W. Tsang, "Learning with idealized kernels," in *Proc. 20th Int. Conf. Mach. Learn.*, Washington, DC, Aug. 21–24, 2003, pp. 400–407.

[25] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semi-definite programming," in *Proc. 19th Int. Conf. Mach. Learn.*, Sydney, Australia, Jul. 8–12, 2002, pp. 323–330.
[26] D. G. Lowe, "Similarity metric learning for a variable-kernel classifier," *Neural Comput.*, vol. 7, no. 1, pp. 72–85, 1995.
[27] Z. Lu and T. K. Leen, "Semi-supervised learning with penalized probabilistic clustering," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 849–856.
[28] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[29] C. S. Ong, A. J. Smola, and R. C. Williamson, "Hyperkernels," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 478–485.
[30] J. Peng, D. R. Heisterkamp, and H. K. Dai, "Adaptive kernel metric nearest neighbor classification," in *Proc. 16th Int. Conf. Pattern Recognit.*, Québec City, QC, Canada, Aug. 11–15, 2002, vol. 3, pp. 33–36.
[31] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, pp. 846–850, 1971.
[32] B. Schölkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.
[33] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.
[34] A. Schwaighofer, V. Tresp, and K. Yu, "Learning Gaussian process kernels via hierarchical Bayes," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1209–1216.
[35] M. Seeger, "Learning With labeled and Unlabeled Data" Inst. Adaptive and Neural Computation, Univ. Edinburgh, Edinburgh, U.K., Tech. Rep., 2000.
[36] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng, "Online and batch learning of pseudo-metrics," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 4–8, 2004, pp. 743–750.
[37] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, "Computing Gaussian mixture models with EM using equivalence constraints," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.
[38] R. D. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 622–627, Sep. 1981.
[39] J. Sinkkonen and S. Kaski, "Clustering based on conditional distributions in an auxiliary space," *Neural Comput.*, vol. 14, no. 1, pp. 217–239, 2002.
[40] I. W. Tsang and J. T. Kwok, "Efficient hyperkernel learning using secondorder cone programming," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 48–58, Jan. 2006.
[41] K. Tsuda, S. Akaho, and K. Asai, "The em algorithm for kernel matrix completion with auxiliary data," *J. Mach. Learn. Res.*, vol. 4, pp. 67–81, 2003.
[42] K. Tsuda, G. Rätsch, and M. K. Warmuth, "Matrix exponentiated gradient updates for on-line learning and Bregman projection," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1425–1432.
[43] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints," in *Proc. 17th Int. Conf. Mach. Learn.*, Stanford, CA, Jun. 29–Jul. 2 2000, pp. 1103–1110.
[44] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proc. 18th Int. Conf. Mach. Learn.*, Williamstown, MA, Jun. 28– Jul. 1 2001, pp. 577–584.
[45] 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 Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 505–512.
[46] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 460–474, Mar. 2005.
[47] D. Y. Yeung and H. Chang, "Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints," *Pattern Recognit.*, vol. 39, no. 5, pp. 1007–1010, 2006.
[48] Z. Zhang, J. T. Kwok, and D. Y. Yeung, "Parametric distance metric learning with label information," in *Proc. 18th Int. Joint Conf. Artif. Intell.*, Acapulco, Mexico, Aug. 9–15, 2003, pp. 1450–1452.
[49] ——, "Model-based transductive learning of the kernel matrix," *Mach. Learn.*, vol. 63, no. 1, pp. 69–101, 2006.
[50] Z. Zhang, D. Y. Yeung, and J. T. Kwok, "Bayesian inference for transductive learning of kernel matrix using the Tanner-Wong data augmentation algorithm," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, Jul. 4–8, 2004, pp. 935–942.
[51] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty, "Nonparametric transforms of graph kernels for semi-supervised learning," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1641–1648.

Dit-Yan Yeung (M'05) received the B.Eng. degree in electrical engineering and the M.Phil. degree in computer science from the University of Hong Kong, Hong Kong, in 1983 and 1985, respectively, and the Ph.D. degree in computer science from the University of Southern California, Los Angeles, in 1989.

He was an Assistant Professor at the Illinois Institute of Technology, Chicago, before he joined the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, Kowloon, Hong Kong, where he is currently an Associate Professor. His current research interests are in machine learning and pattern recognition.

Hong Chang received the B.S. degree from Hebei University of Technology, Hebei, China, the M.Phil. degree from Tianjin University, Tianjin, China, and the Ph.D. degree from Hong Kong University of Science and Technology, Kowloon, Hong Kong, all in computer science, in 1998, 2001, and 2006, respectively.

She is currently a Postdoctoral Fellow in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. Her main research interests include semisupervised learning, nonlinear dimensionality reduction, and related applications.