# A Kernel Approach for Semi-Supervised 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 semi-supervised metric learning based on pairwise similarity or dissimilarity information. In this paper, we propose a kernel approach for semi-supervised 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*— metric learning, kernel learning, semi-supervised learning, clustering.**

## 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 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], [38]. More recent work includes [12], [13], [15], [17], [20], [21], [26], [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 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 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 semi-supervised 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], [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 semi-supervised clustering tasks by modifying the standard $k$-means clustering algorithm 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

Dr. Dit-Yan Yeung is with Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Email: dyyeung@cse.ust.hk

[1] In principle, it is possible to incorporate pairwise information that quantifies the degree of similarity or dissimilarity as well 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.

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, non-iterative 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 our recent work [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], [41], [42], [46], [49], [50], [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 the last section.

## 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 $\phi$ 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 semi-definite, 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 semi-definite 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 the next two subsections, 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 semi-supervised 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 $i$th 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

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

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

[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.

function can be rewritten as

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

where

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

and

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

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 $\rho$ to minimize the following Lagrangian:

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

We then compute the partial derivatives

$$\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{\beta}} &= 2\mathbf{D}_{\mathcal{S}}\boldsymbol{\beta} - \rho\mathbf{1} \qquad (11) \\
\frac{\partial J}{\partial \rho} &= c - \mathbf{1}^T\boldsymbol{\beta}. \qquad (12)
\end{aligned}$$

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

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

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,

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

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

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

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

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:

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

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 inter-point 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 LLMA [6].

Metric learning is formulated as an unconstrained optimization problem by minimizing

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

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

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

where

$$\begin{aligned}
\gamma_{ij} &= s_{ij} + \rho\mathcal{N}_{\sigma}(c_{ij}), \qquad (19) \\
p_{ij} &= \frac{\rho\mathcal{N}_{\sigma}(c_{ij})c_{ij}}{s_{ij} + \rho\mathcal{N}_{\sigma}(c_{ij})}, \qquad (20)
\end{aligned}$$

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 Equation (21) below, $q_{ij}^2(\mathbf{A})$ is quadratic in $\mathbf{A}$ and $q_{ij}(\mathbf{A}) =$

[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 $\boldsymbol{\beta}$.

$\|\mathbf{\Phi}^T\mathbf{A}^T(\mathbf{b}_i-\mathbf{b}_j)\|$. 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 Equation (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

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

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

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

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

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

The first term on the right-hand side of the above equation can be rewritten as

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

where

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

The second term on the right-hand side of Equation (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 Equation (22), we have

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

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

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

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

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

where

$$\begin{aligned} f_{ij}(\mathbf{B}) &= \begin{cases} \dfrac{\rho\mathcal{N}_\sigma(c_{ij})c_{ij}}{q_{ij}(\mathbf{B})} & q_{ij}(\mathbf{B})>0 \\ 0 & q_{ij}(\mathbf{B})=0 \end{cases}, \qquad (29) \\ \mathbf{F}(\mathbf{B}) &= \sum_{i,j=1}^{n}f_{ij}(\mathbf{B})(\mathbf{b}_i-\mathbf{b}_j)(\mathbf{b}_i-\mathbf{b}_j)^T. \qquad (30) \end{aligned}$$

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

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

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 $\dfrac{\partial\hat{J}(\mathbf{A},\mathbf{B})}{\partial\mathbf{A}}$ 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

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

or

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

where $\mathbf{E}^+$ is the pseudo-inverse 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:

$$\mathbf{A}^{(t)} = \mathbf{E}^+\mathbf{F}(\mathbf{A}^{(t-1)})\mathbf{A}^{(t-1)}, \qquad (34)$$

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:

$$\mathbf{A}^{(t)} = \mathbf{E}^+\mathbf{F}(\mathbf{A}^{(t-1)})\mathbf{A}^{(t-1)}.$$

3) If converged, then stop. Otherwise, repeat from step 2.

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

## 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 semi-supervised 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 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], [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 for the Gaussian window parameter $\sigma$ used in the regularization term (Equation (16)), we make it depend on the average squared Euclidean distance between all point pairs in the feature space: $\sigma^2 = \frac{\theta}{n^2}\sum_{i,j=1}^{n}\|\phi(\mathbf{x}_i)-\phi(\mathbf{x}_j)\|^2 = \frac{2\theta}{n}[\mathrm{Tr}(\mathbf{K})-n\overline{\mathbf{K}}]$, where $\overline{\mathbf{K}}$ represents the mean value of the elements in $\mathbf{K}$ and $\theta$ is set to be the same $(=5)$ for all data sets. The regularization parameter $\rho$ in the kernel-$\mathbf{A}$ method (Equation (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 Figure 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 Figure 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 2-dimensional space, as shown in Figure 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 2-moon data set which is commonly used in some recent semi-supervised 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 Figure 2 again show that the kernel-based methods can give promising results.

We further perform some semi-supervised clustering experiments on the XOR and 2-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 Figure 3 below. For each trial, 10 point pairs are randomly selected to form $\mathcal{S}$.

### C. Experiments on UCI Data

We perform more semi-supervised clustering experiments on six real-world data sets from the 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 $|\mathcal{S}|$ are shown for each data set in Table I.

Figure 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

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

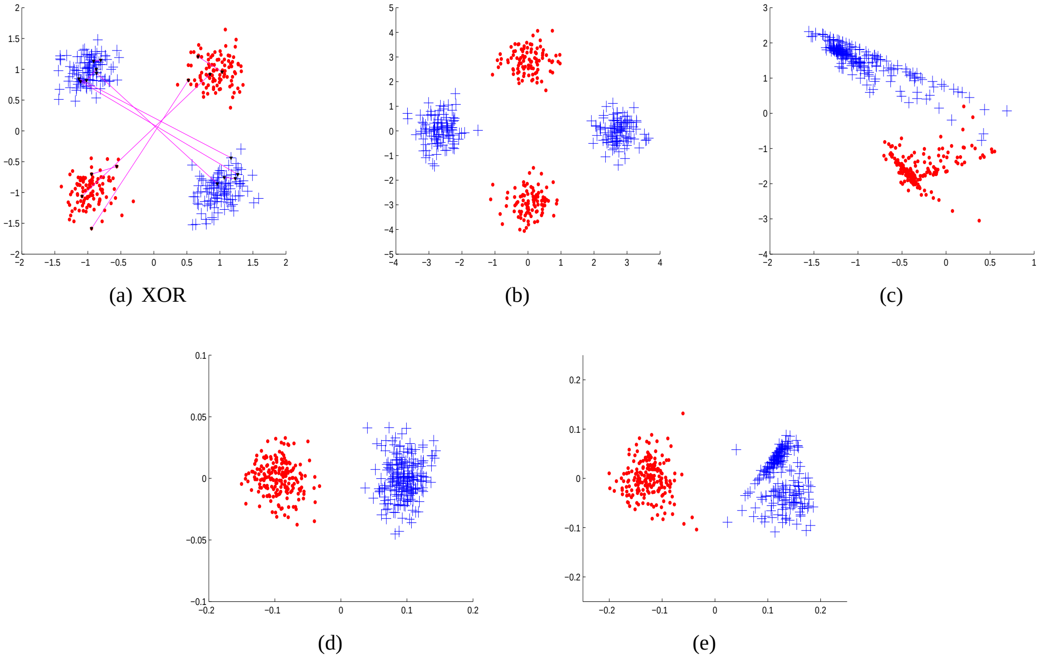


Fig. 1. Comparison of different metric learning methods on the XOR data set. (a) original data set with two classes; and the data set after applying (b) RCA; (c) LLMA; (d) kernel-$\boldsymbol{\beta}$; (e) kernel-$\mathbf{A}$.

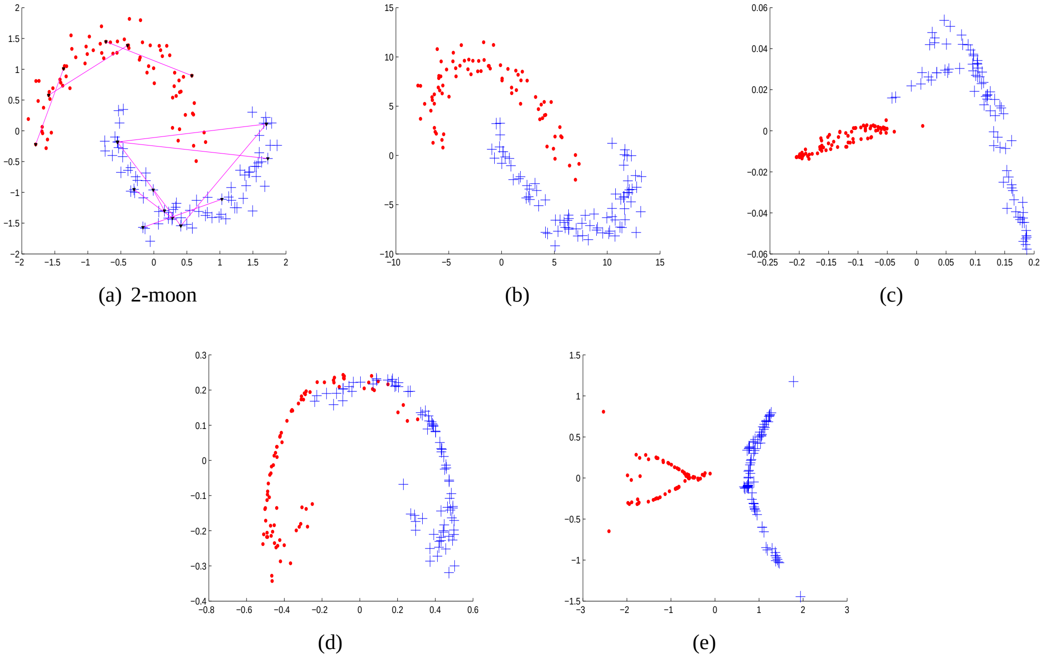
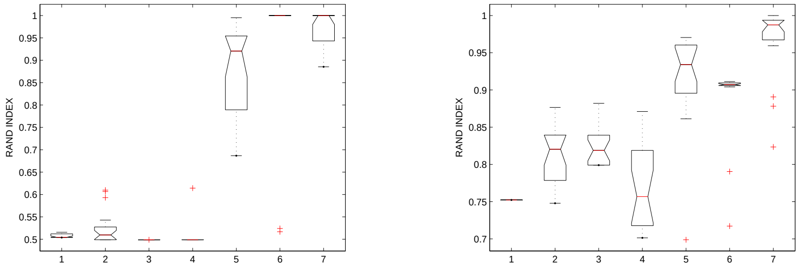


Fig. 2. Comparison of different metric learning methods on the 2-moon data set. (a) original data set with two classes; and the data set after applying (b) RCA; (c) LLMA; (d) kernel-$\boldsymbol{\beta}$; (e) kernel-$\mathbf{A}$.



(a)

(b)

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

TABLE I
SIX UCI DATA SETS USED IN THE EXPERIMENTS

| DATA SET | $n$ | $d$ | $m$ | $\|\mathcal{S}\|$ |
|---|---|---|---|---|
| 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 |

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 Figure 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 $28\times28$ gray-level images. Hence the dimensionality of the input space is 784. In our experiments, we randomly choose 1,000 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 10 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 (2 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).

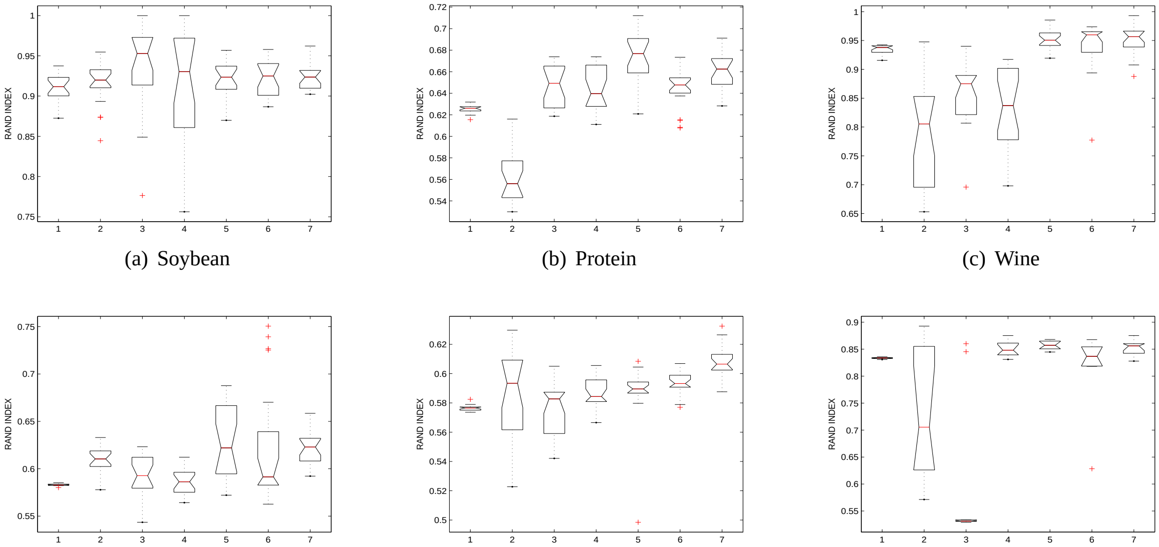
## IV. CONCLUDING REMARKS

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, non-square $\mathbf{A}$ 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], [8], [9].

## ACKNOWLEDGMENTS

The research reported in this paper has been supported by Competitive Earmarked Research Grant (CERG) HKUST6174/04E from the Research Grants Council of the Hong Kong Special Administrative Region, China.

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



(a) Soybean (b) Protein (c) Wine

(d) Ionosphere (e) Boston housing (f) Breast cancer

Fig. 4. Clustering results for six UCI data sets. The seven clustering algorithms (numbered in Section III-A) are: (1) Euclidean; (2) RCA; (3) MPCK-Means-$\mathcal{S}$; (4) MPCK-Means-$\mathcal{SD}$; (5) LLMA; (6) kernel-$\boldsymbol{\beta}$; and (7) kernel-$\mathbf{A}$.

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** |
| | ±0.0004 | ±0.0109 | ±0.0105 | ±0.0055 | ±0.0015 | ±0.0009 | ±0.0011 |
| $\{1, 5\}$ | 0.8179 | 0.8410 | 0.8254 | 0.8156 | 0.8013 | **0.8682** | 0.8590 |
| | ±0.0001 | ±0.0211 | ±0.0075 | ±0.0068 | ±0.1370 | ±0.0534 | ±0.0089 |
| $\{1, 9\}$ | 0.9531 | 0.9546 | 0.9275 | 0.9317 | 0.9527 | 0.9609 | **0.9657** |
| | ±0.0000 | ±0.0319 | ±0.0137 | ±0.0087 | ±0.0012 | ±0.0203 | ±0.0068 |

## References

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

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 11–18, Washington, DC, USA, 21–24 August 2003.

[3] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 19–26, Sydney, Australia, 8–12 July 2002.

[4] M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 81–88, Banff, Alberta, Canada, 4–8 July 2004.

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

[6] H. Chang and D.Y. Yeung. Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 153–160, Banff, Alberta, Canada, 4–8 July 2004.

[7] H. Chang and D.Y. Yeung. Kernel-based distance metric learning for content-based image retrieval. *Image and Vision Computing*, 2006. To appear.

[8] H. Chang and D.Y. Yeung. Locally linear metric adaptation with application to semi-supervised clustering and image retrieval. *Pattern Recognition*, 39(7):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 Recognition*, 39(10):1905–1917, 2006.

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

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

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

[13] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, 2002.

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

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

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

[17] J.H. Friedman. Flexible metric nearest neighbor classification. Technical report, Department of Statistics, Stanford University, Stanford, CA, USA, November 1994.

[18] K. Fukunaga and T.E. Flick. An optimal global nearest neighbor metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):314–318, 1984.

[19] K. Fukunaga and L. Hostetler. Optimization of $k$-nearest neighbor density estimates. *IEEE Transactions on Information Theory*, 19(3):320–326, 1973.

[20] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, Cambridge, MA, USA, 2005.

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

[22] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 393–400, Banff, Alberta, Canada, 4–8 July 2004.

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

[24] J.T. Kwok and I.W. Tsang. Learning with idealized kernels. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 400–407, Washington, DC, USA, 21–24 August 2003.

[25] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semi-definite programming. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 323–330, Sydney, Australia, 8–12 July 2002.

[26] D.G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.

[27] Z. Lu and T.K. Leen. Semi-supervised learning with penalized probabilistic clustering. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 849–856. MIT Press, Cambridge, MA, USA, 2005.

[28] T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, USA, 1997.

[29] C.S. Ong, A.J. Smola, and R.C. Williamson. Hyperkernels. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 478–485. MIT Press, Cambridge, MA, USA, 2003.

[30] J. Peng, D.R. Heisterkamp, and H.K. Dai. Adaptive kernel metric nearest neighbor classification. In *Proceedings of the Sixteenth International Conference on Pattern Recognition*, volume 3, pages 33–36, Québec City, Québec, Canada, 11–15 August 2002.

[31] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[32] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, USA, 2002.

[33] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, USA, 2004.

[34] A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian process kernels via hierarchical Bayes. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1209–1216. MIT Press, Cambridge, MA, USA, 2005.

[35] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, UK, 2000.

[36] S. Shalev-Shwartz, Y. Singer, and A.Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 743–750, Banff, Alberta, Canada, 4–8 July 2004.

[37] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, USA, 2004.

[38] R.D. Short and K. Fukunaga. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27(5):622–627, 1981.

[39] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14(1):217–239, 2002.

[40] I.W. Tsang and J.T. Kwok. Efficient hyperkernel learning using second-order cone programming. *IEEE Transactions on Neural Networks*, 17(1):48–58, 2006.

[41] K. Tsuda, S. Akaho, and K. Asai. The *em* algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, 2003.

[42] K. Tsuda, G. Rätsch, and M.K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1425–1432. MIT Press, Cambridge, MA, USA, 2005.

[43] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, Stanford, CA, USA, 29 June – 2 July 2000.

[44] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained $k$-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, Williamstown, MA, USA, 28 June – 1 July 2001.

[45] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press, Cambridge, MA, USA, 2003.

[46] H. Xiong, M.N.S. Swamy, and M.O. Ahmad. Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, 16(2):460–474, 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 Recognition*, 39(5):1007–1010, 2006.

[48] Z. Zhang, J.T. Kwok, and D.Y. Yeung. Parametric distance metric learning with label information. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1450–1452, Acapulco, Mexico, 9–15 August 2003.

[49] Z. Zhang, J.T. Kwok, and D.Y. Yeung. Model-based transductive learning of the kernel matrix. *Machine Learning*, 63(1):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 *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 935–942, Banff, Alberta, Canada, 4–8 July 2004.

[51] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1641–1648. MIT Press, Cambridge, MA, USA, 2005.

**Dit-Yan Yeung** received his B.Eng. degree in Electrical Engineering and M.Phil. degree in Computer Science from the University of Hong Kong, and his Ph.D. degree in Computer Science from the University of Southern California in Los Angeles. He was an Assistant Professor at the Illinois Institute of Technology in Chicago before he joined the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, where he is currently an Associate Professor. His current research interests are in machine learning and pattern recognition.

**Hong Chang** received her Bachelor degree, M.Phil. degree and Ph.D. degree in Computer Science from Hebei University of Technology, Tianjin University, and Hong Kong University of Science and Technology, China, 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 semi-supervised learning, nonlinear dimensionality reduction and related applications.