International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(B)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 7017–7030

DISTANCE METRIC LEARNING BY QUADRATIC PROGRAMMING BASED ON EQUIVALENCE CONSTRAINTS

Hakan Cevikalp

Electrical and Electronics Engineering Department
Eskisehir Osmangazi University
Meselik, Eskisehir 26480, Turkey
hcevikalp@ogu.edu.tr

Received July 2011; revised November 2011

Abstract. *This paper introduces a new distance metric learning algorithm which uses pair-wise equivalence (similarity and dissimilarity) constraints to improve the original distance metric in lower-dimensional input spaces. We restrict ourselves to pseudo-metrics that are in quadratic forms parameterized by positive semi-definite matrices. Learning a pseudo distance metric from equivalence constraints is formulated as a quadratic optimization problem, and we also integrate the large margin concept into the formulation. The proposed method works in both the input space and kernel induced feature space, and experimental results on several databases show that the learned distance metric improves the performances of the subsequent classification and clustering algorithms.*

Keywords: Distance metric learning, Classification, Clustering, Quadratic programming

1. **Introduction.** Learning distance metrics is very important for various applications such as classification, image and video retrieval, and image segmentation [2, 6, 11, 16, 17]. While measuring distances seems to be a simple problem when the data samples are represented with enough discriminatory features, in many real-world applications data samples may consist of many irrelevant features for the task being considered. Consider organizing image galleries in accordance to the personal preferences: For example, one may want to group the images based on more abstract concepts such as outdoors or indoors. Similarly, we may want to group face images by race or gender. In most of the cases we consider here, the data samples have many irrelevant features, and the typical distance functions employed in these kinds of applications such as the Euclidean distance or Gaussian kernels do not give satisfactory results. Thus, we need to learn good distance functions to bridge the gap between the irrelevant data features and the goal of the user for the specific task at hand.

Learning distance metrics is much easier when the labels associated to the data samples are available. However, in many applications, there is a lack of labeled data since obtaining labels is a costly procedure as it often requires human effort. On the other hand, in some applications, side information – given in the form of pairwise equivalence (similarity and dissimilarity) constraints between points – is available without or with less extra cost. For instance, consider the surveillance application given in [25]: Faces extracted from successive video frames in roughly the same location can be assumed to represent the same person, whereas faces extracted in different locations cannot be the same person. In some applications, side information is the natural form of supervision; e.g., in image retrieval, there is only the notion of similarities between the query and retrieved images.

Side information may also come from human feedback in interactive environments often at a substantially lower cost than explicit labeled data as in semi-supervised image segmentation applications [6].

Recently, learning distance metrics has been actively studied in machine learning. Some of the distance metric learning algorithms use class labels [10, 12, 13, 23, 27], and we will not consider them here. We will focus only on semi-supervised (or weakly supervised) distance metric learning algorithms which use equivalence constraints. Existing semi-supervised distance metric learning methods [3, 8, 16, 18, 24, 26, 28, 30] revise the original distance metric (commonly chosen as the Euclidean distance) to accommodate the pair-wise equivalence constraints, and then a clustering algorithm with the learned distance metric is usually used to partition the data to discover the desired groups within data. In [28], a full-rank pseudo distance metric is learned by means of convex programming using equivalence constraints. Relevant Component Analysis [3] was introduced as an alternative to this method, but it can exploit only similarity constraints. Shalew-Shwartz et al. [24] proposed a sophisticated online distance metric learning algorithm that uses side information. The method incorporates the large margin concept, and the distance metric is modified based on two successive projections involving an eigen-decomposition. Yang et al. [29] introduced a Bayesian framework for distance metric learning that estimates a posterior distribution for the distance metric from pair-wise equivalence constraints. Davis et al. [8] proposed an information-theoretic approach to learn a Mahalanobis distance function using equivalence constraints. They formulated the metric learning problem as that of minimizing the differential relative entropy between two multivariate Gaussians under equivalence constraints on the distance function. Methods that are more closely related to ours were introduced in [18, 26]. They formulated the problem as a quadratic optimization scheme and extended their method to the nonlinear case using the kernel trick. As we discuss later, although they claim that the learned metric is a pseudo-metric, there is no guarantee that the resulting distance matrix is positive semi-definite. Note that all semi-supervised distance metric learning algorithms mentioned above attempt to learn full-rank distance metrics, and thus they are suitable for low-dimensional input spaces. In high-dimensional spaces, it is better to learn low-rank distance metrics (or low-dimensional embeddings). To this end, the authors in [1, 7] revise the Locality Preserving Projections method to exploit side information. Cevikalp and Paredes [6] introduce a low-rank distance metric learning algorithm that uses equivalence constraints and sigmoid functions. A similar semi-supervised distance metric learning method based on sigmoid functions is also introduced in [14]. In [9, 20], a low-rank Mahalanobis distance metric for high-dimensional spaces is learned based on the log-determinant matrix divergence. In addition to these methods, there are some hybrid algorithms that unify clustering and metric learning into a unique framework [5]. A comprehensive survey of semi-supervised distance metric learning techniques can be found in [30].

In this paper, we also focus on lower-dimensional input spaces and try to learn a pseudo distance metric parameterized by positive semi-definite matrices. To this end, we formulate the distance metric learning problem as a quadratic optimization problem as in [18, 26]. However, our proposed method differs from those quadratic optimization based methods in two ways. Firstly, we incorporate the large margin concept in the method which is ignored in the other quadratic learning schemes. Secondly, there are less user-chosen parameters in our method. This offers savings during training and makes the method more appealing for the users who are not experienced in using such learning algorithms.

2. **Method.**

2.1. **Problem setting.** Let $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, denote the samples in the training set. We are given a set of equivalence constraints in the form of similar and dissimilar pairs. Let S be the set of similar sample pairs

$$S = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same class}\}$$

and let D be the set of dissimilar sample pairs

$$D = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to different classes}\}.$$

Assuming consistency of the constraints, the constraint sets can be augmented by using transitivity and entailment properties as in [4].

Our objective is to find a pseudo-metric that satisfies the equivalence constraints and at the same time reflects the true underlying relationships imposed by such constraints. We focus on pseudo-metrics of the form

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_{\mathbf{A}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{1}$$

where $\mathbf{A} \geq \mathbf{0}$ is a symmetric positive semi-definite matrix. In this case there exists a rectangular projection matrix $\mathbf{W}$ of size $q \times d$ ($q \leq d$) satisfying $\mathbf{A} = \mathbf{W}^\top \mathbf{W}$ such that

$$||\mathbf{x}_i - \mathbf{x}_j||^2_{\mathbf{A}} = ||\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j||^2. \tag{2}$$

From this point of view, the distance between two points under metric $\mathbf{A}$ can be interpreted as linear projection of samples by $\mathbf{W}$ followed by the Euclidean distance in the projected space.

2.2. **Learning distance metric by quadratic programming.** Assume that the learned distance matrix is $\mathbf{A}$. Let us ignore the positive semi-definiteness constraint for the moment. Intuitively, the learned distance metric must pull similar sample pairs closer and push the dissimilar sample pairs apart. Additionally, it should generalize the unseen data well. To this end, we define the margin b, which is defined to be the minimum separation between all pairs of similar and dissimilar samples. That is

$$d^2_{\mathbf{A}}(\mathbf{x}_k, \mathbf{x}_l) - d^2_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) \geq b, \;\; (\mathbf{x}_k, \mathbf{x}_l) \in D \text{ and } (\mathbf{x}_i, \mathbf{x}_j) \in S.$$

Without loss of generality, we can scale $\mathbf{A}$ and b by any positive constant. We therefore set b to be equal to 2 and search for a distance matrix $\mathbf{A}$ which has small Frobenius norm. However, if we have m similar and n dissimilar sample pairs, the number of total constraints will be mn, which may be a large number to handle. Therefore, we introduce a threshold $\gamma' \geq 1$ and replace the constraints with

$$\begin{aligned} &d^2_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) \leq \gamma' - 1, \;\; (\mathbf{x}_i, \mathbf{x}_j) \in S, \\ &d^2_{\mathbf{A}}(\mathbf{x}_k, \mathbf{x}_l) \geq \gamma' + 1, \;\; (\mathbf{x}_k, \mathbf{x}_l) \in D. \end{aligned} \tag{3}$$

If we let $\gamma = \gamma' - 1$ and introduce slack variables for the sample pairs violating margin constraints, we obtain the following quadratic programming problem

$$\begin{aligned} \min \;& \frac{1}{2}||\mathbf{A}||_2^2 + \frac{C_S}{n_S} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} \xi_{ij} + \frac{C_D}{n_D} \sum_{(\mathbf{x}_k, \mathbf{x}_l) \in D} \xi_{kl} \\ \text{s.t. } & d^2_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) \leq \gamma + \xi_{ij}, \quad (\mathbf{x}_i, \mathbf{x}_j) \in S, \\ & d^2_{\mathbf{A}}(\mathbf{x}_k, \mathbf{x}_l) \geq \gamma + 2 - \xi_{kl}, \quad (\mathbf{x}_k, \mathbf{x}_l) \in D, \\ & \gamma, \xi_{ij}, \xi_{kl} \geq 0 \end{aligned} \tag{4}$$

where n_S and n_D are the numbers of pairs in S and D respectively, C_S, C_D are non-negative user-chosen adjustable parameters, and ξ_{ij}, ξ_{kl} are positive slack variables. Here $||\mathbf{A}||_2$ represents the Frobenius norm of matrix $\mathbf{A}$. Note that the similar sample pairs

which are far from each other contribute more to the loss function than the ones which are closer. In a similar manner, the dissimilar sample pairs which are closer to each other contribute more to the loss function than the ones which are further from each other. In fact if the square of distances between the dissimilar sample pairs are larger than the threshold ($\gamma^{'}+1$ or equivalently $\gamma+2$), those dissimilar sample pairs do not contribute to the loss function at all. Therefore, just as in the Support Vector Machine's hinge loss, our objective function is triggered by the dissimilar sample pairs in the vicinity of decision boundaries that participate the inter-class decision boundaries. In contrast, there is not such a systematical selection mechanism that respects the margin concept in the methods of [18, 26]. They just aim to pull all similar sample pairs together and to maximize the distance differences between the learned and original distance metrics for dissimilar sample pairs.

To derive the dual, we consider the Lagrangian

$$\begin{aligned} L(\mathbf{A},\xi,\gamma,\alpha,\eta,\mu) = & \frac{1}{2}||\mathbf{A}||_2^2 + \frac{C_S}{n_S}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\xi_{ij} + \frac{C_D}{n_D}\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\xi_{kl} \\ & + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij}\left\{(\mathbf{x}_i-\mathbf{x}_j)^\top\mathbf{A}(\mathbf{x}_i-\mathbf{x}_j)-\gamma-\xi_{ij}\right\} \\ & + \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl}\left\{-(\mathbf{x}_k-\mathbf{x}_l)^\top\mathbf{A}(\mathbf{x}_k-\mathbf{x}_l)+\gamma+2-\xi_{kl}\right\} \\ & - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\eta_{ij}\xi_{ij} - \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\eta_{kl}\xi_{kl} - \mu\gamma \end{aligned} \tag{5}$$

where α_{ij}, α_{kl}, η_{ij}, η_{kl}, $\mu \geq 0$. The Lagrangian L has to be maximized with respect to α, η, μ and minimized with respect to $\mathbf{A}$, ξ, γ. The optimality conditions yield

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{A}} = \mathbf{0} \rightarrow \mathbf{A} = \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl}(\mathbf{x}_k-\mathbf{x}_l)(\mathbf{x}_k-\mathbf{x}_l)^\top - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top, \\ &\frac{\partial L}{\partial \xi_{ij}} = \mathbf{0} \rightarrow \alpha_{ij} = \frac{C_S}{n_S} - \eta_{ij} \rightarrow 0 \leq \alpha_{ij} \leq \frac{C_S}{n_S}, \;\; (\mathbf{x}_i,\mathbf{x}_j)\in S, \\ &\frac{\partial L}{\partial \xi_{kl}} = \mathbf{0} \rightarrow \alpha_{kl} = \frac{C_D}{n_D} - \eta_{kl} \rightarrow 0 \leq \alpha_{kl} \leq \frac{C_D}{n_D}, \;\; (\mathbf{x}_k,\mathbf{x}_l)\in D, \\ &\frac{\partial L}{\partial \gamma} = \mathbf{0} \rightarrow \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl} - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij} = \mu \rightarrow \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl} - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij} \geq 0. \end{aligned}$$

Thus, the dual of the optimization problem becomes

$$\begin{aligned} \min \;\; & \tfrac{1}{2}\textstyle\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\sum_{(\mathbf{x}_m,\mathbf{x}_n)\in S}\alpha_{ij}\alpha_{mn}[(\mathbf{x}_i-\mathbf{x}_j)^\top(\mathbf{x}_m-\mathbf{x}_n)]^2 \\ & + \tfrac{1}{2}\textstyle\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\sum_{(\mathbf{x}_p,\mathbf{x}_q)\in D}\alpha_{kl}\alpha_{pq}[(\mathbf{x}_k-\mathbf{x}_l)^\top(\mathbf{x}_p-\mathbf{x}_q)]^2 \\ & - \textstyle\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{ij}\alpha_{kl}[(\mathbf{x}_i-\mathbf{x}_j)^\top(\mathbf{x}_k-\mathbf{x}_l)]^2 - 2\textstyle\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl} \\ \text{subject to} \;\; & \textstyle\sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D}\alpha_{kl} - \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S}\alpha_{ij} \geq 0, 0 \leq \alpha_{ij} \leq \frac{C_S}{n_S} \text{ and } 0 \leq \alpha_{kl} \leq \frac{C_D}{n_D}. \end{aligned} \tag{6}$$

This is a quadratic programming problem with $n = n_S + n_D$ variables (which is independent of input dimensionality d) as in [18]. Therefore, training time complexity of the method is $O(n^3)$. As a result, the proposed method is more efficient than most of the semi-supervised distance metric learning algorithms that require $O(n^3)$ complexity per iteration. Note that the Hessian matrix of this quadratic programming problem is not necessarily positive semi-definite because of the last minus quadratic term. However,

we can always reconstruct the Hessian matrix by using the positive eigenvalues and corresponding eigenvectors to ensure the positive semi-definiteness. In this case a global minimum exists. From the Karush-Kuhn-Tucker conditions, we get

$$d^2_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) \left\{ \begin{array}{ll} = \gamma & 0 < \alpha_{ij} < C_S/n_S, \\ \leq \gamma & \alpha_{ij} = 0, \\ \geq \gamma & \alpha_{ij} = C_S/n_S. \end{array} \right\} \quad (\mathbf{x}_i, \mathbf{x}_j) \in S \tag{7}$$

$$d^2_{\mathbf{A}}(\mathbf{x}_k, \mathbf{x}_l) \left\{ \begin{array}{ll} = \gamma + 2 & 0 < \alpha_{kl} < C_D/n_D, \\ \geq \gamma + 2 & \alpha_{kl} = 0, \\ \leq \gamma + 2 & \alpha_{kl} = C_D/n_D. \end{array} \right\} \quad (\mathbf{x}_k, \mathbf{x}_l) \in D \tag{8}$$

Thus, to find the value of γ, we take all sample pairs with $0 < \alpha_{ij} < C_S/n_S$ and $0 < \alpha_{kl} < C_D/n_D$, compute corresponding $d^2_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j)$ and $d^2_{\mathbf{A}}(\mathbf{x}_k, \mathbf{x}_l) - 2$ and average them.

It should be noted that the resulting distance matrix $\mathbf{A}$ is not necessarily a positive semi-definite matrix. One way to circumvent this problem is to work on the primal problem (4), which leads to a gradient descent algorithm, and then to ensure the positive-definiteness during the iterations as in [22]. However, the kernelization of this approach is not straightforward. A more simple solution is to solve the dual problem (6) and make sure that $\mathbf{A}$ is a positive semi-definite matrix at the end. In this approach, we first check if the returned distance matrix is positive semi-definite. If the resulting distance matrix is not positive semi-definite, we apply eigen-decomposition to $\mathbf{A}$ and reconstruct it using positive eigenvalues and corresponding eigenvectors, $\mathbf{A} = \sum_k \lambda_k \mathbf{u}_k \mathbf{u}_k^\top = \mathbf{U}\Lambda\mathbf{U}^\top$ where λ_k's are the positive eigenvalues, $\mathbf{u}_k$'s are the corresponding eigenvectors, Λ is a diagonal matrix including positive eigenvalues λ_k as diagonal entries, and $\mathbf{U}$ is the matrix whose columns are the corresponding eigenvectors $\mathbf{u}_k$. In this case, we can use any existing quadratic programming software without any modifications, and extension of the method to the nonlinear case is much easier as described later. Thus, we used this approach in this study.

Sometimes we may be interested in lower-dimensional embeddings induced by the pseudo-distance metric rather than the distance metric itself. As we mentioned earlier, the distance between two samples under positive semi-definite matrix $\mathbf{A}$ can be interpreted as linear projection of the samples by $\mathbf{W}$ followed by the Euclidean distance in the projected space. Computing embeddings (linear projections) of samples $\mathbf{x}_i$ by using $\mathbf{W}\mathbf{x}_i$ offers several advantages. For example, projections onto 2 or 3-dimensional space allow us visualization of data, so we can devise an interactive constraints selection tool and verify the effects of our selections visually. Also, we can run existing algorithms such as k-means clustering on embedded samples without any modifications. Projection matrix $\mathbf{W}$ induced by positive semi definite matrix $\mathbf{A}$ can be found as $\mathbf{W} = \Lambda^{1/2}\mathbf{U}^\top$.

2.3. **Extension to the nonlinear case.** Here we consider the case where the data samples are mapped into a higher-dimensional feature space, and the distance metric is sought in this new feature space. This is accomplished by using the kernel trick. Notice that the objective function of (6) can be written in terms of the dot products of the sample pairs. Thus, we replace all $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \mathbf{x}_i^\top \mathbf{x}_j$ with the kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ where $\phi : \mathbb{R}^d \to \Im$ is the mapping function from the input space to the feature space $\Im$. Once we compute the optimal α coefficients, the distance between two samples $\phi(\mathbf{x}_a)$ and $\phi(\mathbf{x}_b)$ in the mapped space under the distance metric $\mathbf{A}$

can be computed as

$$\begin{aligned} d_{\mathbf{A}}(\phi(\mathbf{x}_a),\phi(\mathbf{x}_b)) &= (\phi(\mathbf{x}_a)-\phi(\mathbf{x}_b))^\top \mathbf{A}(\phi(\mathbf{x}_a)-\phi(\mathbf{x}_b)) \\ &= \phi(\mathbf{x}_a)^\top \mathbf{A}\phi(\mathbf{x}_a) - 2\phi(\mathbf{x}_a)^\top \mathbf{A}\phi(\mathbf{x}_b) + \phi(\mathbf{x}_b)^\top \mathbf{A}\phi(\mathbf{x}_b) \\ &= \tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_a) - 2\tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_b) + \tilde{k}_{\mathbf{A}}(\mathbf{x}_b,\mathbf{x}_b) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_b) =& \phi(\mathbf{x}_a)^\top \mathbf{A}\phi(\mathbf{x}_b) \\ =& \sum_{(\mathbf{x}_k,\mathbf{x}_l)\in D} \alpha_{kl}\left\{\left[k(\mathbf{x}_a,\mathbf{x}_k)-k(\mathbf{x}_a,\mathbf{x}_l)\right]\left[k(\mathbf{x}_k,\mathbf{x}_b)-k(\mathbf{x}_l,\mathbf{x}_b)\right]\right\} \\ &- \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S} \alpha_{ij}\left\{\left[k(\mathbf{x}_a,\mathbf{x}_i)-k(\mathbf{x}_a,\mathbf{x}_j)\right]\left[k(\mathbf{x}_i,\mathbf{x}_b)-k(\mathbf{x}_j,\mathbf{x}_b)\right]\right\}. \end{aligned}$$

However, there is no guarantee that the resulting distance matrix $\mathbf{A}$ is positive-definite. Thus, we have to reconstruct it by using positive eigenvalues and corresponding eigenvectors as in linear case. But note that it is impossible to reach directly to matrix $\mathbf{A}$ for the nonlinear case. This has to be done by formulating the eigen-decomposition problem in terms of dot products of samples.

Now let $\Phi_S = [\phi(\mathbf{x}_{1,1})-\phi(\mathbf{x}_{1,2}),\ldots,\phi(\mathbf{x}_{n_S,1})-\phi(\mathbf{x}_{n_S,2})]$ be the matrix including the ordered difference vectors of similar sample pairs from S in the mapped space, $\Phi_D = [\phi(\mathbf{x}_{1,1})-\phi(\mathbf{x}_{1,2}),\ldots,\phi(\mathbf{x}_{n_D,1})-\phi(\mathbf{x}_{n_D,2})]$ be the matrix including the ordered difference vectors of dissimilar sample pairs in the mapped space, α_S^* be the vector including optimal coefficients corresponding to similar sample pairs returned by the quadratic optimization algorithm, and α_D^* be the vector including optimal coefficients corresponding to dissimilar sample pairs. If we define $\Phi_{dif} \equiv \begin{pmatrix}\Phi_S & \Phi_D\end{pmatrix}$ and $\alpha \equiv \begin{pmatrix}-\alpha_S^* \\ \alpha_D^*\end{pmatrix}$, the resulting distance matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \Phi_{dif}\Omega\Phi_{dif}^\top, \tag{10}$$

where $\Omega \in \mathbb{R}^{n\times n}$ is a diagonal matrix including $\alpha \in \mathbb{R}^n$ as its diagonal entries. Our goal is to find eigenvalues $\lambda > 0$ and corresponding eigenvectors $\mathbf{u}$ satisfying

$$\lambda\mathbf{u} = \mathbf{A}\mathbf{u}. \tag{11}$$

All eigenvectors $\mathbf{u}$ corresponding to positive eigenvalues lie in the span of column vectors of Φ_{dif}, i.e., $\mathbf{u} = \Phi_{dif}\mathbf{v}$. Thus, if we multiply (11) with $\Phi_{dif}^\top$ from left, we obtain

$$\begin{aligned} &\lambda\Phi_{dif}^\top\mathbf{u} = \Phi_{dif}^\top\Phi_{dif}\Omega\Phi_{dif}^\top\mathbf{u} \\ &\lambda\Phi_{dif}^\top\Phi_{dif}\mathbf{v} = \Phi_{dif}^\top\Phi_{dif}\Omega\Phi_{dif}^\top\Phi_{dif}\mathbf{v} \\ &\lambda\mathbf{K}_{dif}\mathbf{v} = \mathbf{K}_{dif}\Omega\mathbf{K}_{dif}\mathbf{v} \\ &\lambda\mathbf{v} = (\Omega\mathbf{K}_{dif})\mathbf{v} \Rightarrow \lambda\mathbf{v} = \widetilde{\mathbf{K}}_{dif}\mathbf{v}. \end{aligned} \tag{12}$$

Now let λ_k denote a positive eigenvalue of $\widetilde{\mathbf{K}}_{dif}$ and $\mathbf{v}_k$ is the corresponding eigenvector. We have to normalize $\mathbf{v}_k$ to satisfy the equation $\mathbf{u}_k^\top\mathbf{u}_k = \mathbf{v}_k^\top\mathbf{K}_{dif}\mathbf{v}_k = 1$. Assume that $\widetilde{\mathbf{v}}_k$ is the normalized eigenvector. Then, the positive semi-definite matrix $\mathbf{A}$ can be written as $\mathbf{A} = \sum_k \lambda_k\Phi_{dif}\widetilde{\mathbf{v}}_k(\Phi_{dif}\widetilde{\mathbf{v}}_k)^\top = (\Phi_{dif}\mathbf{V})\Lambda(\Phi_{dif}\mathbf{V})^\top$, where Λ is the diagonal matrix including λ_k's as diagonal entries and $\mathbf{V}$ is the matrix whose columns include corresponding eigenvectors $\widetilde{\mathbf{v}}_k$. As a result, the distance between two samples $\phi(\mathbf{x}_a)$ and $\phi(\mathbf{x}_b)$ under the pseudo distance metric $\mathbf{A}$ can be computed using (9) by replacing $\tilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_b)$ with

$$\widetilde{k}_{\mathbf{A}}(\mathbf{x}_a,\mathbf{x}_b) = \sum_k \lambda_k(\phi(\mathbf{x}_a)^\top\Phi_{dif}\widetilde{\mathbf{v}}_k)(\widetilde{\mathbf{v}}_k^\top\Phi_{dif}^\top\phi(\mathbf{x}_b)). \tag{13}$$

The rectangular embedding matrix induced by $\mathbf{A}$ is $\mathbf{W} = \Lambda^{1/2}\mathbf{V}^\top\Phi_{dif}^\top$, thus embeddings of samples $\phi(\mathbf{x}_i)$ can be computed as follows:

$$\mathbf{W}\phi(\mathbf{x}_i) \;=\; \Lambda^{1/2}\mathbf{V}^\top(\Phi_{dif}^\top\phi(\mathbf{x}_i)) = \Lambda^{1/2}\mathbf{V}^\top\mathbf{k}_{\mathbf{x}_i}^{dif}, \tag{14}$$

where $\mathbf{k}_{\mathbf{x}_i}^{dif} = (\Phi_{dif}^\top\phi(\mathbf{x}_i)) = [k(\mathbf{x}_{j,1},\mathbf{x}_i) - k(\mathbf{x}_{j,2},\mathbf{x}_i)]$ is a $n \times 1$ vector of $\mathbf{x}_i$ against the similar and dissimilar sample pairs $\mathbf{x}_{j,1}$ and $\mathbf{x}_{j,2}$.

3. **Experiments.** We performed experiments[1] on two synthetic databases, several real-world databases chosen from UCI repository (http://www.ics.uci.edu/∼mlearn/MLRepository.html) and ETH-80 [19] database. We compared the distance metric obtained by the proposed method, Quadratic Programming based Distance Metric Learning (QPDML), to the Euclidean distance metric and the distance metrics learned by the Relevant Component Analysis (RCA) [3] and the method of [26]. It should be noted that the proposed method and the quadratic distance metric learning method of [26] do not necessarily yield to positive semi-definite matrices. Therefore, we reconstructed the resulting distance matrix for both methods by using only positive eigenvalues and corresponding eigenvectors as described earlier. For the nonlinear case, we used polynomial kernels with degree 2 and the Gaussian kernels.

In order to assess the performance of the distance metrics, we evaluated both the clustering and classification performances. For classification, we used 1-nearest neighbor classification rule with the learned distance metrics. The k-means and spectral clustering are used as clustering algorithms (we report the one yielding the best result), and the pair-wise F-measure is used to evaluate the clustering results based on the underlying classes. The pairwise F-measure is the harmonic mean of the pairwise precision and recall measures which are widely used in information retrieval. We compute precision and recall over pairs of samples and consider for the pairs whether they are assigned to the same cluster by clustering algorithms and whether they contain the same class label. Let A denote the set of sample pairs assigned to the same cluster, and let B denote the set of sample pairs that contain the same class label. With $|A|$ denoting the cardinality of A (and similar for other sets), the measures are defined as:

$$\text{Precision} = \frac{|A \cap B|}{|A|}, \quad \text{Recall} = \frac{|A \cap B|}{|B|}, \quad \text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

3.1. **Experiments on synthetic databases.** The first synthetic database includes 10-dimensional data samples belonging to two classes. The first dimension is the distinctive feature, where the first class is normally distributed as $N(3,1)$ and the second class as $N(-3,1)$. The remaining dimensions are irrelevant features distributed as $N(0,16)$. Since the data are linearly separable, we only tested linear distance learning methods for this database. We created 100 samples for each class and used 50 samples per class for choosing equivalence constraints and the remaining samples are used for testing. We used only 100 (60 similarity and 40 dissimilarity) equivalence constraints. Classification and clustering accuracies are given in Table 1 and Table 2, respectively. Results are averages over 50 runs. Since the first synthetic data has identical covariance distribution for both classes, RCA performs the best as expected. Our proposed method comes the second outperforming method of [26] with a slight edge. Figure 1 illustrates 2-dimensional embeddings of test samples learned by the proposed method and method of [26]. Our proposed method clearly finds better low-dimensional embeddings where the samples are more separable compared with the method of [26]. In Figure 2, we plot affinity (similarity) matrices obtained using different distance metrics. The heat kernel function $\exp(-d(\mathbf{x}_i,\mathbf{x}_j)^2/t)$ is

[1]For software see *http://www2.ogu.edu.tr/∼mlcv/softwares.html*.

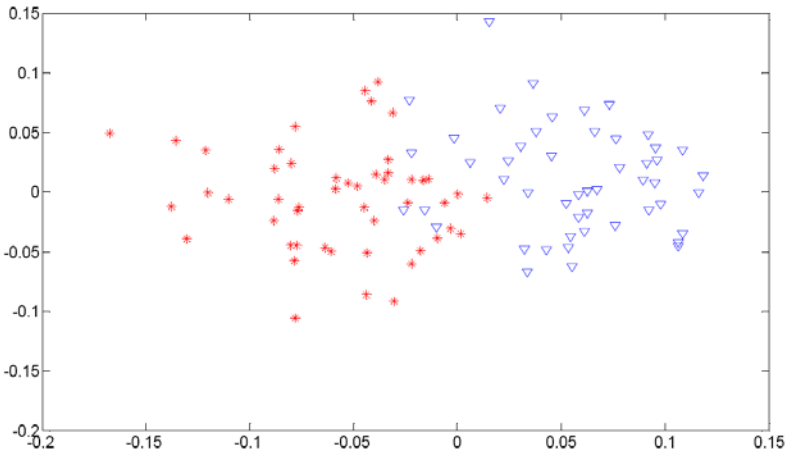
used to measure the similarity between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and brighter pixels show that the corresponding sample pairs are more similar. Again, similarity matrix learned by the proposed method is the most similar one to the ideal case.

TABLE 1. Classification accuracies on synthetic databases

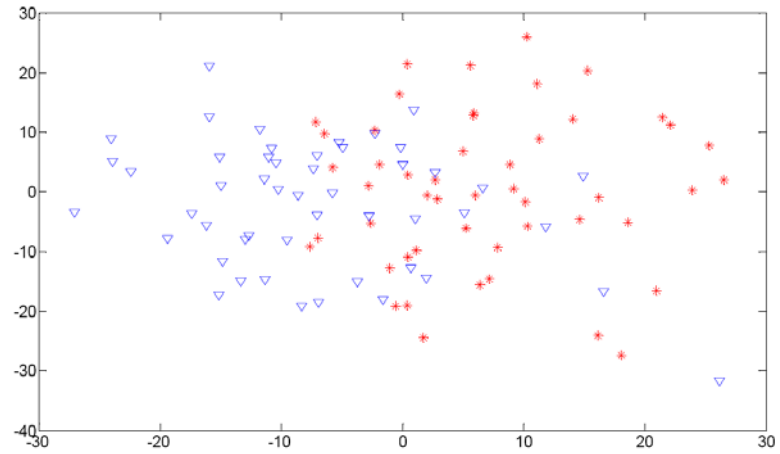
Data	Kernel	Euclidean Metric	RCA	Tsang and Kwok [26]	QPDML
1st Synt.	Linear	81.1 ± 4.0	$\mathbf{98.9} \pm 0.9$	93.8 ± 4.2	94.1 ± 4.2
2nd Synt.	Polynomial	$\mathbf{99.95} \pm 0.2$	–	$\mathbf{99.95} \pm 0.2$	$\mathbf{99.95} \pm 0.2$
	Gaussian		–	99.89 ± 0.6	$\mathbf{99.95} \pm 0.3$

TABLE 2. Clustering accuracies on synthetic databases

Data	Kernel	Euclidean Metric	RCA	Tsang and Kwok [26]	QPDML
1st Synt.	Linear	58.2 ± 8.1	$\mathbf{96.4} \pm 1.2$	87.4 ± 11.2	88.6 ± 11.8
2nd Synt.	Polynomial	66.22 ± 3.7	–	99.94 ± 0.2	$\mathbf{99.95} \pm 0.2$
	Gaussian		–	99.18 ± 5.1	$\mathbf{99.94} \pm 0.6$

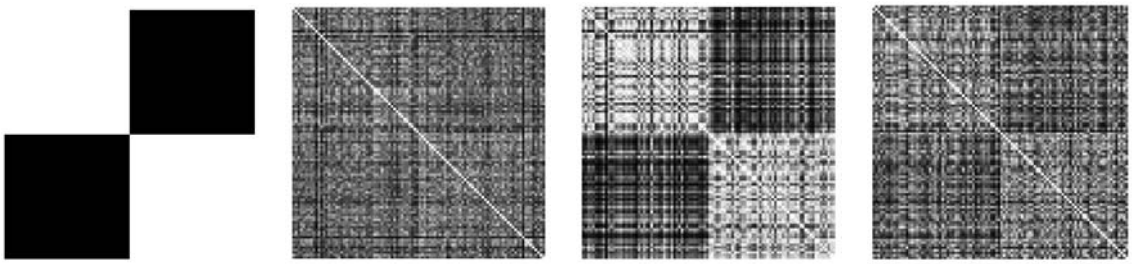


(a)



(b)

FIGURE 1. 2-dimensional embeddings formed using the most significant eigenvectors for the first synthetic database: (a) proposed method, (b) method of [26] (figure is best viewed in color)



(a) (b) (c) (d)

FIGURE 2. Visualization of affinity (similarity) matrices obtained using different distance metrics: (a) ideal case, (b) Euclidean metric in the original input space, (c) distance metric learned by the proposed method, (d) distance metric learned by the method of [26]

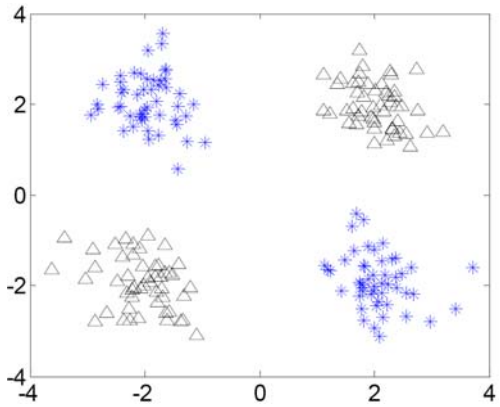


FIGURE 3. XOR data

For the second synthetic database, we used 2-dimensional samples drawn from two-component mixture models which are typically used in XOR problem. Figure 3 illustrates the data. Classes are not linearly separable, thus we tested kernel methods with the polynomial kernel with degree 2 and the Gaussian kernel. We used only 40 (20 similarity and 20 dissimilarity) constraints. Classification and clustering accuracies are given in Table 1 and Table 2, respectively. The results are again averages over 50 runs as in the previous case. For this database RCA does not work well since the data has nonlinear distribution. The best classification accuracy is obtained by both the proposed method and the Euclidean metric, whereas our proposed method is the best performer in terms of clustering accuracy. Note that the clustering performance of the Euclidean metric is very low. In general, all metric learning methods show an improvement over the Euclidean metric. The 2-dimensional embeddings of test samples learned by the tested methods are given in Figure 4, and the affinity matrices of the tested samples are plotted in Figure 5. The low-dimensional embeddings and affinity matrices are similar for both tested distance metric learning methods using quadratic programming. The lower-dimensional embeddings of the test samples obtained using the polynomial kernel are more separable than the ones obtained using the Gaussian kernel. Also affinity matrices obtained using the polynomial kernel are more similar to the ideal one, which shows that the polynomial kernel is a better choice than the Gaussian kernel for this problem.

3.2. **Experiments on UCI repository databases.** Here we tested our proposed method on four databases (Iris, Ionosphere, Wine, and Wisconsin Diagnostic Breast Cancer

TABLE 3. Low-dimensional databases selected from UCI repository

Databases	Number of Classes	Data Set Size	Dimensionality
Ionosphere	2	351	34
Iris	3	150	4
Wine	3	178	13
WDBC	2	569	30

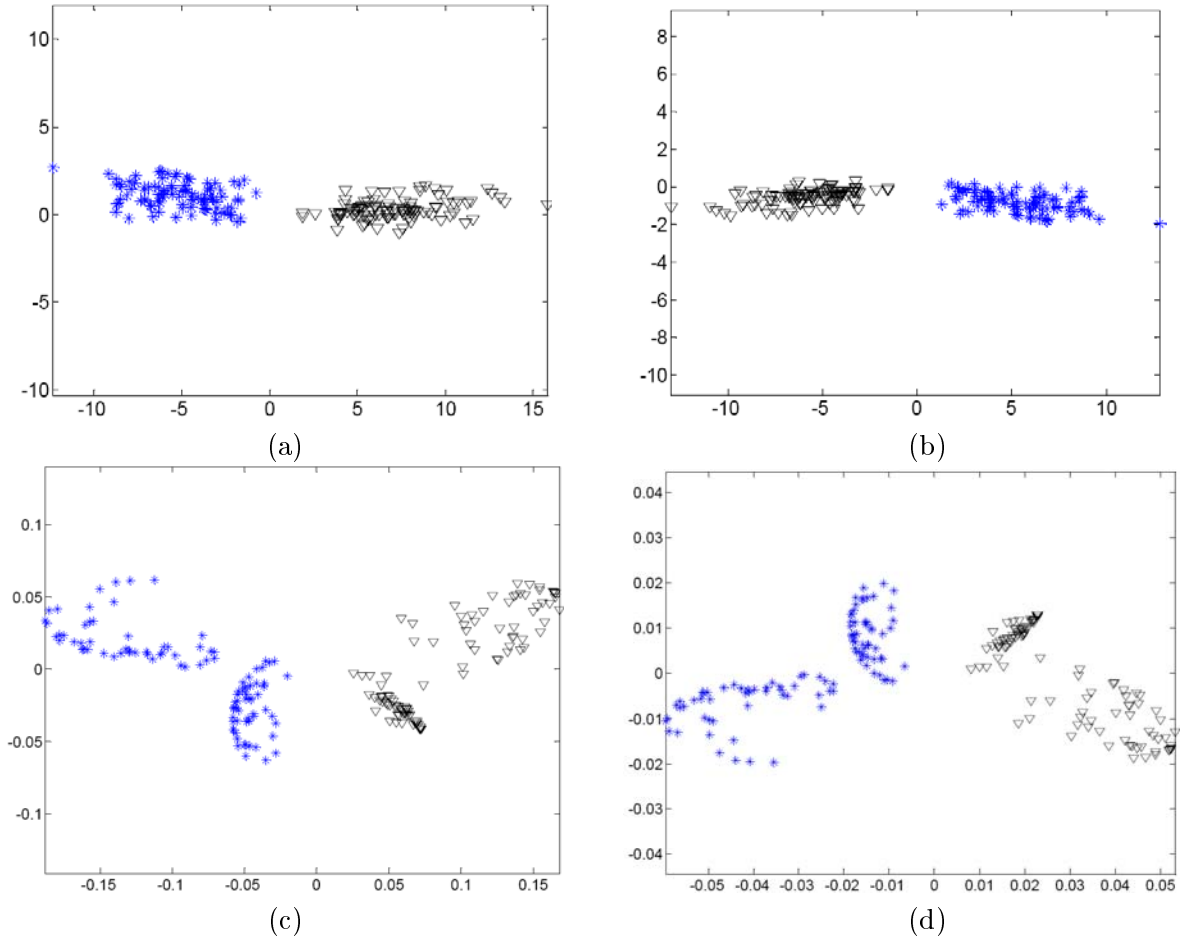


FIGURE 4. 2-dimensional embeddings formed using the most significant eigenvectors on the second synthetic database: (a) proposed method for polynomial kernel, (b) Tsang and Kwok's method [26] for polynomial kernel, (c) proposed method for the Gaussian kernel, (d) Tsang and Kwok's method [26] for the Gaussian kernel

– WDBC) chosen from UCI Repository. The key parameters of these datasets are summarized in Table 3. For all datasets, we used the half of the samples for choosing 150 pair-wise equivalence constraints, and the remaining data samples are used for testing. Classification and clustering accuracies are given in Table 4 and Table 5, respectively. Results are averages over 20 runs.

Our proposed method with the Gaussian kernel achieves the best classification accuracies for all databases as shown in Table 4. In terms of clustering accuracy, RCA wins for the Iris database whereas the proposed method again achieves the best results for the

remaining three databases. Overall, the experiments show that the proposed method is the best performer among all tested methods.

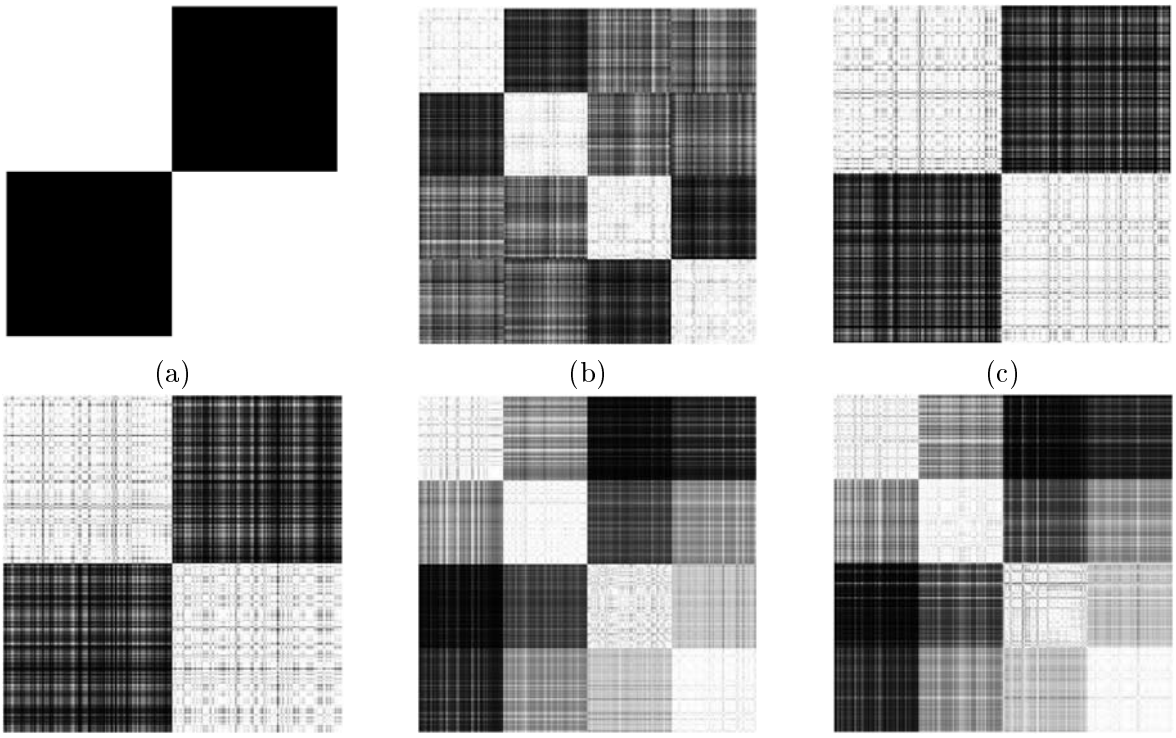


FIGURE 5. Visualization of affinity (similarity) matrices obtained using different distance metrics on the second synthetic database: (a) ideal case, (b) Euclidean metric in the original input space, (c) distance metric learned by the proposed method using polynomial kernel, (d) distance metric learned by the method of [26] using polynomial kernel, (e) distance metric learned by the proposed method using the Gaussian kernel, (f) distance metric learned by the method of [26] using the Gaussian kernel

TABLE 4. Classification accuracies (%) for UCI databases

Data	Kernel	Euclidean Metric	RCA	Tsang and Kwok [26]	QPDML
Iris	Linear	95.83 ± 2.5	95.75 ± 2.6	93.56 ± 3.2	95.00 ± 3.1
	Polynomial		–	91.00 ± 2.7	92.66 ± 3.4
	Gaussian		–	96.16 ± 3.0	$\mathbf{96.67} \pm 2.7$
Ionosphere	Linear	87.23 ± 1.9	89.77 ± 2.7	86.59 ± 2.6	88.87 ± 2.7
	Polynomial		–	86.30 ± 2.1	88.91 ± 2.5
	Gaussian		–	92.87 ± 2.3	$\mathbf{93.23} \pm 1.9$
Wine	Linear	93.15 ± 1.9	95.21 ± 2.5	95.71 ± 2.0	95.91 ± 2.0
	Polynomial		–	95.48 ± 2.0	96.59 ± 2.2
	Gaussian		–	95.15 ± 1.9	$\mathbf{96.75} \pm 2.2$
WDBC	Linear	94.26 ± 1.4	89.08 ± 2.7	93.56 ± 1.1	95.03 ± 1.0
	Polynomial		–	94.26 ± 1.1	94.16 ± 0.9
	Gaussian		–	94.27 ± 0.9	$\mathbf{95.28} \pm 0.9$

TABLE 5. Clustering accuracies (%) for UCI databases

Data	Kernel	Euclidean Metric	RCA	Tsang and Kwok [26]	QPDML
Iris	Linear	82.22 ± 8.1	**90.52** ± 7.1	86.23 ± 10.7	88.67 ± 11.3
	Polynomial		–	83.58 ± 3.9	85.93 ± 5.8
	Gaussian		–	89.74 ± 7.7	88.50 ± 8.1
Ionosphere	Linear	60.12 ± 1.8	70.91 ± 4.0	60.70 ± 4.1	74.02 ± 6.7
	Polynomial		–	61.81 ± 11.3	68.32 ± 8.6
	Gaussian		–	72.17 ± 5.6	**80.37** ± 7.7
Wine	Linear	82.67 ± 7.2	86.13 ± 5.0	84.87 ± 3.0	86.41 ± 3.5
	Polynomial		–	84.70 ± 3.4	86.62 ± 4.0
	Gaussian		–	84.90 ± 4.7	**86.83** ± 4.0
WDBC	Linear	86.09 ± 2.3	81.51 ± 2.4	87.13 ± 1.3	89.44 ± 2.2
	Polynomial		–	84.93 ± 1.1	84.58 ± 1.1
	Gaussian		–	87.13 ± 1.3	**90.11** ± 1.3

TABLE 6. Classification accuracies (%) for ETH database

Kernel	Euclidean Metric	RCA	Tsang and Kwok [26]	QPDML
Linear	99.39 ± 0.6	**99.51** ± 0.5	99.39 ± 0.6	99.39 ± 0.6
Polynomial		–	98.91 ± 1.3	99.02 ± 1.1
Gaussian		–	99.39 ± 0.6	99.39 ± 0.6

TABLE 7. Clustering accuracies (%) for ETH database

Kernel	Euclidean Metric	RCA	Tsang and Kwok [26]	QPDML
Linear	90.10 ± 4.6	**98.06** ± 3.2	94.21 ± 4.5	96.36 ± 3.7
Polynomial		–	87.28 ± 9.9	87.61 ± 10.1
Gaussian		–	97.30 ± 4.3	97.72 ± 3.8

3.3. **Experiments on ETH database.** To assess the performance of our method, we have performed experiments on ETH-80 [19] database to discover object groups. We used only four categories from the ETH-80: *Apple, Car, Cow* and *Cup*. Each category contains images of 10 to 14 objects under different viewpoints, against a flat blue background. We used a 'bag of features' representation for the images as they are too diverse to allow simple geometric alignment of their objects. In this approach, patches are sampled from the image at many different positions and scales, either densely, randomly or based on the output of some kind of salient region detector. In our case we select patches following a dense grid. Then each patch is represented by a 128-dimensional SIFT descriptor [21]. Following this process, all descriptors extracted from images are quantized in a discrete set of so-called 'visual keywords' forming a vocabulary. To build image representation, each extracted descriptor is compared with the visual keywords and associated to the closest keyword. Based on these assignments, we build histograms which are used as image feature vectors. The size of the histograms is chosen to be equal to 500. The dimensionality is too high thus we first reduced the dimensionality to 10 by using Locality Preserving Projection method [15]. We used 200 equivalence constraints to learn the distance metrics. Classification and clustering accuracies are given in Table 6 and Table 7, respectively, and results are obtained using 5-fold cross validation. Both the best classification and clustering accuracies are obtained by RCA. All remaining methods yield

the same second best classification accuracy. However, in terms of clustering accuracy, our proposed method comes the second best performer.

4. **Summary and Conclusion.** In this paper we proposed a new pseudo-distance metric learning method that uses pair-wise equivalence constraints. The metric learning problem is formulated as a quadratic optimization problem as in [26], but we encourage maximizing local margins in the process as well. Our proposed method can work over an implicit nonlinear feature space by using the kernel trick, and the number of user-chosen parameters is less compared with the method of [26] (User has to fix two parameters in our proposed method whereas three parameters must be set in the method of [26]). We also showed how to find the lower dimensional embeddings induced by the learned pseudo-distance metrics. Experimental results show that the proposed method increases performance of subsequent clustering and classification algorithms in many cases, and it usually outperforms the method of [26].

Acknowledgment. This work was supported by the Young Scientists Award Programme (TÜBA-GEBİP 2011-2012) of the Turkish Academy of Sciences.

REFERENCES

[1] S. An, W. Liu and S. Venkatesh, Exploiting side information in locality preserving projection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[2] B. Babenko, S. Branson and S. Belongie, Similarity metrics for categorization: From monolithic to category specific, *International Conference on Computer Vision*, 2009.

[3] A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall, Learning distance functions using equivalence relations, *International Conference on Machine Learning*, 2003.

[4] S. Basu, A. Banerjee and R. J. Mooney, Active semi-supervision for pairwise constrained clustering, *The SIAM International Conference on Data Mining*, 2004.

[5] M. Bilenko, S. Basu and R. J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, *International Conference on Machine Learning*, 2004.

[6] H. Cevikalp and R. Paredes, Semi-supervised distance metric learning for visual object classification, *International Conference on Computer Vision Theory and Applications*, 2009.

[7] H. Cevikalp, J. Verbeek, F. Jurie and A. Klaser, Semi-supervised dimensionality reduction using pairwise equivalence constraints, *International Conference on Computer Vision Theory and Applications*, 2008.

[8] J. V. Davis, B. Kullis, P. Jain and I. S. Dhillon, Information-theoretic metric learning, *International Conference on Machine Learning*, 2007.

[9] J. V. Davis and I. S. Dhillon, Structured metric learning for high-dimensional problems, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.195-210, 2008.

[10] C. Domeniconi, J. Peng and D. Gunopulos, Locally adaptive metric nearest-neighbor classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.9, pp.1281-1285, 2002.

[11] A. Ghodsi, D. Wilkinson and F. Southey, Improving embeddings by flexible exploitation of side information, *International Joint Conference on Artificial Intelligence*, 2007.

[12] A. Globerson and S. Roweis, Metric learning by collapsing classes, *Advances in Neural Information Processing Systems*, 2005.

[13] J. Goldberger, S. Roweis, G. Hinton and R. Salakhutdinow, Neighborhood component analysis, *Advances in Neural Information Processing Systems*, 2004.

[14] M. Guillaumin, J. Verbeek and C. Schmid, Is that you? Metric learning approaches for face identification, *International Conference on Computer Vision*, 2009.

[15] X. He and P. Niyogi, Locality preserving projections, *Advances in Neural Information Processing Systems*, 2003.

[16] T. Hertz, N. Shental, A. Bar-Hillel and D. Weinshall, Enhancing image and video retrieval: Learning via equivalence constraints, *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, 2003.

[17] S. C. H. Hoi, W. Liu and S.-F. Chang, Semi-supervised distance metric learning for collaborative image retrieval, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[18] J. Kwok and I. W. Tsang, Learning with idealized kernels, *International Conference on Machine Learning*, 2003.

[19] B. Leibe and B. Schiele, Interleaved object categorization and segmentation, *British Machine Vision Conference*, 2003.

[20] W. Liu, S. Ma, D. Tao, J. Liu and P. Liu, Semi-supervised sparse metric learning using alternating linearization optimization, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1139-1147, 2010.

[21] D. G. Lowe, Distinctive image features from scale – Invariant keypoints, *International Journal of Computer Vision*, vol.60, pp.91-110, 2004.

[22] N. Nguyen and Y. Guo, Metric learning: A support vector approach, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2008.

[23] R. Paredes and E. Vidal, Learning weighted metrics to minimize nearest-neighbor classification error, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.28, no.7, pp.1100-1110, 2006.

[24] S. Shalew-Shwartz, Y. Singer and A. Y. Ng, Online and batch learning of pseudo-metrics, *International Conference on Machine Learning*, 2004.

[25] N. Shental, T. Hertz, D. Weinshall and M. Pavel, Adjustment learning and relevant component analysis, *European Conference on Computer Vision*, 2002.

[26] I. W. Tsang and J. Kwok, Distance metric learning with kernels, *International Conference on Artificial Neural Networks*, 2003.

[27] K. Q. Weinberger and L. K. Saul, Distance metric learning for large margin nearest neighbor classification, *Journal of Machine Learning Research*, vol.10, pp.207-244, 2009.

[28] E. P. Xing, A. Y. Ng, M. Jordan and S. Russell, Distance metric learning with application to clustering with side-information, *Advances in Neural Information Processing Systems*, 2003.

[29] L. Yang, R. Jin and R. Sukthankar, Bayesian active distance metric learning, *Proc. of the 23rd Conference on Uncertainty in Artificial Intelligence*, 2007.

[30] L. Yang and R. Jin, *Distance Metric Learning: A Comprehensive Survey*, http://www.cse.msu.edu/∼yangliu1/framesurveyv2.pdf, 2006.