A Kernel Classification Framework for Metric Learning

Faqiang Wang, Wangmeng Zuo, *Member, IEEE,* Lei Zhang, *Senior Member, IEEE,* Deyu Meng, and David Zhang, *Fellow, IEEE*

***Abstract*—Learning a distance metric from the given training samples plays a crucial role in many machine learning tasks, and various models and optimization algorithms have been proposed in the past decade. In this paper, we generalize several state-of-the-art metric learning methods, such as large margin nearest neighbor (LMNN) and information theoretic metric learning (ITML), into a kernel classification framework. First, doublets and triplets are constructed from the training samples, and a family of degree-2 polynomial kernel functions are proposed for pairs of doublets or triplets. Then, a kernel classification framework is established to generalize many popular metric learning methods such as LMNN and ITML. The proposed framework can also suggest new metric learning methods, which can be efficiently implemented, interestingly, by using the standard support vector machine (SVM) solvers. Two novel metric learning methods, namely doublet-SVM and triplet-SVM, are then developed under the proposed framework. Experimental results show that doublet-SVM and triplet-SVM achieve competitive classification accuracies with state-of-the-art metric learning methods but with significantly less training time.**

***Index Terms*—Metric learning, support vector machine, nearest neighbor, kernel method, polynomial kernel.**

I. Introduction

HOW to measure the distance (or similarity/dissimilarity) between two data points is a fundamental issue in unsupervised and supervised pattern recognition. The desired distance metrics can vary a lot in different applications due to the underlying data structures and distributions, as well as the specificity of the learning tasks. Learning a distance metric from the given training examples has been an active topic in the past decade [1], [2], and it can improve much the performance of many clustering (e.g., k-means) and classification (e.g., k-nearest neighbors) methods. Distance metric learning has been successfully adopted in many real world applications, e.g., face identification [3], face verification [4], image retrieval [5], [6], and activity recognition [7].

Generally speaking, the goal of distance metric learning is to learn a distance metric from a given collection of similar/dissimilar samples by punishing the large distances between similar pairs and the small distances between dissimilar pairs. So far, numerous methods have been proposed to learn distance metrics, similarity metrics, and even nonlinear distance metrics. Among them, learning the Mahalanobis distance metrics for k-nearest neighbor classification has been receiving considerable research interests [3], [8]–[15]. The problem of similarity learning has been studied as learning correlation metrics and cosine similarity metrics [16]–[20]. Several methods have been proposed for nonlinear distance metric learning [21]–[23]. Extensions of metric learning have also been investigated for multiple kernel learning [21], semi-supervised learning [5], [24], [25], multiple instance learning [26], and multi-task learning [27], [28], etc.

Despite that many metric learning approaches have been proposed, there are still some issues to be further studied. First, since metric learning learns a distance metric from the given training dataset, it is interesting to investigate whether we can recast metric learning as a standard supervised learning problem. Second, most existing metric learning methods are motivated from specific convex programming or probabilistic models, and it is interesting to investigate whether we can unify them into a general framework. Third, it is highly demanded that the unified framework can provide a good platform for developing new metric learning algorithms, which can be easily solved by standard and efficient learning tools.

With the above considerations, in this paper we present a kernel classification framework to learn a Mahalanobis distance metric in the original feature space, which can unify many state-of-the-art metric learning methods, such as large margin nearest neighbor (LMNN) [8], [29], [30], information theoretic metric learning (ITML) [10], and logistic discriminative based metric learning (LDML) [3]. This framework allows us to easily develop new metric learning methods by using existing kernel classifiers such as the support vector machine (SVM) [31]. Under the proposed framework, we consequently present two novel metric learning methods, namely doublet-SVM and triplet-SVM, by modeling metric learning as an SVM problem, which can be efficiently solved by the existing SVM solvers like LibSVM [32].

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents the proposed kernel classification framework for metric learning. Section IV introduces the doublet-SVM and triplet-SVM methods. Section V presents the experimental results, and Section VI concludes the paper.

Throughout the paper, we denote matrices, vectors and

This work was supported in part by the National Natural Science Foundation of China under Grant 61271093 and Grant 61001037, the Hong Kong Scholar Program, HK RGC GRF grant (PolyU 5313/12E), and the Program of Ministry of Education for New Century Excellent Talents under Grant NCET-12-0150.

F. Wang and W. Zuo are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: tshfqw@163.com; cswmzuo@gmail.com).

L. Zhang and D. Zhang are with the Biometrics Research Center, Department of Computing, the Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: cslzhang@comp.polyu.edu.hk; csdzhang@comp.polyu.edu.hk).

D. Meng is with the Institute for Information and System Sciences, Faculty of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: dymeng@mail.xjtu.edu.cn).

scalars by the upper-case bold-faced letters, lower-case bold-faced letters, and lower-case letters, respectively.

II. Related Work

As a fundamental problem in supervised and unsupervised learning, metric learning has been widely studied and various models have been developed, e.g., LMNN [8], ITML [10] and LDML [3]. Kumar *et al.* extended LMNN for transformation invariant classification [33]. Huang *et al.* proposed a generalized sparse metric learning method to learn low rank distance metrics [11]. Saenko *et al.* extended ITML for visual category domain adaptation [34], while Kulis *et al.* showed that in visual category recognition tasks, asymmetric transform would achieve better classification performance [35]. Cinbis *et al.* adapted LDML to unsupervised metric learning for face identification with uncontrolled video data [36]. Several relaxed pairwise metric learning methods have been developed for efficient Mahalanobis metric learning [37], [38].

Metric learning via dual approaches and kernel methods has also been studied. Shen *et al.* analyzed the Lagrange dual of the exponential loss in the metric learning problem [12], and proposed an efficient dual approach for semi-definite metric learning [15], [39]. Actually, such boosting-like approaches usually represent the metric matrix $\mathbf{M}$ as a linear combination of rank-one matrices [40]. Liu and Vemuri proposed a doubly regularized metric learning method by incorporating two regularization terms in the dual problem [41]. Shalev-Shwartz *et al.* proposed a pseudo-metric online learning algorithm (POLA) to learn distance metric in the kernel space [42]. Besides, a number of pairwise SVM methods have been proposed to learn distance metrics or nonlinear distance functions [43].

In this paper, we will see that most of the aforementioned metric learning approaches can be unified into a kernel classification framework, while this unified framework can allow us to develop new metric learning methods which can be efficiently implemented by off-the-shelf SVM tools.

Wang *et al.* studied nonlinear metric learning with multiple kernel learning, and proposed a framework for metric learning with multiple kernels [21]. In our work, a kernel classification framework is proposed for metric learning in the original feature space, while in [21] a nonlinear distance metric is learned in the kernel induced feature space.

Very recently, Perez-Suay *et al.* [44] studied the connection between SVM and metric learning with doublet-based constraints, and proposed a batch scheme and an online scheme for metric learning. Compared with [44], our proposed kernel classification framework considers both doublet based constraints and triplet based constraints, and the proposed doublet-SVM model is also different from the model in [44].

III. A Kernel Classification based Metric Learning Framework

Current metric learning models largely depend on convex or non-convex optimization techniques, some of which are very inefficient to solve large-scale problems. In this section, we present a kernel classification framework which can unify many state-of-the-art metric learning methods. It also provides a good platform for developing new metric learning algorithms, which can be easily solved by using the efficient kernel classification tools. The connections between the proposed framework and LMNN, ITML, and LDML will also be discussed in detail.

A. Doublets and Triplets

Unlike conventional supervised learning problems, metric learning usually considers a set of constraints imposed on the doublets or triplets of training samples to learn the desired distance metric. It is very interesting and useful to evaluate whether metric learning can be casted as a conventional supervised learning problem. To build a connection between the two problems, we model metric learning as a supervised learning problem operating on a set of doublets or triplets, as described below.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, 2, \cdots, n\}$ be a training dataset, where vector $\mathbf{x}_i \in \mathbb{R}^d$ represents the ith training sample, and scalar y_i represents the class label of $\mathbf{x}_i$. Any two samples extracted from $\mathcal{D}$ can form a doublet $(\mathbf{x}_i, \mathbf{x}_j)$, and we assign a label h to this doublet as follows: $h = -1$ if $y_i = y_j$ and $h = 1$ if $y_i \neq y_j$. For each training sample $\mathbf{x}_i$, we find from $\mathcal{D}$ its m_1 nearest similar neighbors, denoted by $\{\mathbf{x}^s_{i,1}, \cdots, \mathbf{x}^s_{i,m_1}\}$, and its m_2 nearest dissimilar neighbors, denoted by $\{\mathbf{x}^d_{i,1}, \cdots, \mathbf{x}^d_{i,m_2}\}$, and then construct $(m_1 + m_2)$ doublets $\{(\mathbf{x}_i, \mathbf{x}^s_{i,1}), \cdots, (\mathbf{x}_i, \mathbf{x}^s_{i,m_1}), (\mathbf{x}_i, \mathbf{x}^d_{i,1}), \cdots, (\mathbf{x}_i, \mathbf{x}^d_{i,m_2})\}$. By combining all such doublets constructed from all training samples, we build a doublet set, denoted by $\{\mathbf{z}_1, \cdots, \mathbf{z}_{N_d}\}$, where $\mathbf{z}_l = (\mathbf{x}_{l,1}, \mathbf{x}_{l,2})$, $l = 1, 2, \cdots, N_d$. The label of doublet $\mathbf{z}_l$ is denoted by h_l. Note that doublet based constraints are used in ITML [10] and LDML [3], but the details of the construction of doublets are not given.

We call $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ a triplet if three samples $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ are from $\mathcal{D}$ and their class labels satisfy $y_i = y_j \neq y_k$. We adopt the following strategy to construct a triplet set. For each training sample $\mathbf{x}_i$, we find its m_1 nearest neighbors $\{\mathbf{x}^s_{i,1}, \cdots, \mathbf{x}^s_{i,m_1}\}$ which have the same class label as $\mathbf{x}_i$, and m_2 nearest neighbors $\{\mathbf{x}^d_{i,1}, \cdots, \mathbf{x}^d_{i,m_2}\}$ which have different class labels from $\mathbf{x}_i$. We can thus construct $m_1 m_2$ triplets $\{(\mathbf{x}_i, \mathbf{x}^s_{i,j}, \mathbf{x}^d_{i,k}) | j = 1, \cdots, m_1;\ k = 1, \cdots, m_2\}$ for each sample $\mathbf{x}_i$. By grouping all the triplets, we form a triplet set $\{\mathbf{t}_1, \cdots, \mathbf{t}_{N_t}\}$, where $\mathbf{t}_l = (\mathbf{x}_{l,1}, \mathbf{x}_{l,2}, \mathbf{x}_{l,3})$, $l = 1, 2, \cdots, N_t$. Note that for the convenience of expression, here we remove the super-script "s" and "d" from $\mathbf{x}_{l,2}$ and $\mathbf{x}_{l,3}$, respectively. A similar way to construct the triplets was used in LMNN [8] based on the k-nearest neighbors of each sample.

B. A Family of Degree-2 Polynomial Kernels

We then introduce a family of degree-2 polynomial kernel functions which can operate on pairs of the doublets or triplets defined above. With the introduced degree-2 polynomial kernels, distance metric learning can be readily formulated as a kernel classification problem.

Given two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, we define the following function:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{tr}(\mathbf{x}_i \mathbf{x}_i^T \mathbf{x}_j \mathbf{x}_j^T), \tag{1}$$

where $\mathrm{tr}(\bullet)$ represents the trace operator of a matrix. One can easily see that $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2$ is a degree-2 polynomial kernel, and $K(\mathbf{x}_i, \mathbf{x}_j)$ satisfies the Mercer's condition [45].

The kernel function defined in (1) can be extended to a pair of doublets or triplets. Given two doublets $\mathbf{z}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2})$ and $\mathbf{z}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2})$, we define the corresponding degree-2 polynomial kernel as

$$\begin{aligned} K_D(\mathbf{z}_i, \mathbf{z}_j) &= \mathrm{tr}\Big((\mathbf{x}_{i,1} - \mathbf{x}_{i,2})(\mathbf{x}_{i,1} - \mathbf{x}_{i,2})^T \\ &\quad (\mathbf{x}_{j,1} - \mathbf{x}_{j,2})(\mathbf{x}_{j,1} - \mathbf{x}_{j,2})^T\Big) \\ &= \Big[(\mathbf{x}_{i,1} - \mathbf{x}_{i,2})^T (\mathbf{x}_{j,1} - \mathbf{x}_{j,2})\Big]^2. \end{aligned} \tag{2}$$

The kernel function in (2) defines an inner product of two doublets. With this kernel function, we can learn a decision function to tell whether the two samples of a doublet have the same class label. In Section III-C we will show the connection between metric learning and kernel decision function learning.

Given two triplets $\mathbf{t}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3})$ and $\mathbf{t}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2}, \mathbf{x}_{j,3})$, we define the corresponding degree-2 polynomial kernel as

$$K_T(\mathbf{t}_i, \mathbf{t}_j) = \mathrm{tr}\left(\mathbf{T}_i \mathbf{T}_j\right), \tag{3}$$

where

$$\begin{aligned} \mathbf{T}_i &= (\mathbf{x}_{i,1} - \mathbf{x}_{i,3})(\mathbf{x}_{i,1} - \mathbf{x}_{i,3})^T \\ &\quad - (\mathbf{x}_{i,1} - \mathbf{x}_{i,2})(\mathbf{x}_{i,1} - \mathbf{x}_{i,2})^T, \\ \mathbf{T}_j &= (\mathbf{x}_{j,1} - \mathbf{x}_{j,3})(\mathbf{x}_{j,1} - \mathbf{x}_{j,3})^T \\ &\quad - (\mathbf{x}_{j,1} - \mathbf{x}_{j,2})(\mathbf{x}_{j,1} - \mathbf{x}_{j,2})^T. \end{aligned}$$

The kernel function in (3) defines an inner product of two triplets. With this kernel, we can learn a decision function based on the inequality constraints imposed on the triplets. In Section III-C we will also show how to deduce the Mahalanobis metric from the decision function.

C. Metric Learning via Kernel Methods

With the degree-2 polynomial kernels defined in Section III-B, the task of metric learning can be easily solved by kernel methods. More specifically, we can use any kernel classification method to learn a kernel classifier with one of the following two forms

$$g_d(\mathbf{z}) = \mathrm{sgn}\left(\sum\nolimits_l h_l \alpha_l K_D(\mathbf{z}_l, \mathbf{z}) + b\right), \tag{4}$$

$$g_t(\mathbf{t}) = \mathrm{sgn}\left(\sum\nolimits_l \alpha_l K_T(\mathbf{t}_l, \mathbf{t})\right), \tag{5}$$

where $\mathbf{z}_l$, $l = 1, 2, \cdots, N$, is the doublet constructed from the training dataset, h_l is the label of $\mathbf{z}_l$, $\mathbf{t}_l$ is the triplet constructed from the training dataset, $\mathbf{z} = \left(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}\right)$ is the test doublet, $\mathbf{t}$ is the test triplet, α_l is the weight, and b is the bias.

For doublet, we have

$$\begin{aligned} &\sum\nolimits_l h_l \alpha_l \,\mathrm{tr}\Big((\mathbf{x}_{l,1} - \mathbf{x}_{l,2})(\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T \\ &\quad (\mathbf{x}_{(i)} - \mathbf{x}_{(j)})(\mathbf{x}_{(i)} - \mathbf{x}_{(j)})^T\Big) + b \\ &= (\mathbf{x}_{(i)} - \mathbf{x}_{(j)})^T \mathbf{M} (\mathbf{x}_{(i)} - \mathbf{x}_{(j)}) + b, \end{aligned} \tag{6}$$

where

$$\mathbf{M} = \sum\nolimits_l h_l \alpha_l (\mathbf{x}_{l,1} - \mathbf{x}_{l,2})(\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T \tag{7}$$

is the matrix $\mathbf{M}$ of the Mahalanobis distance metric. Thus, the kernel decision function $g_d(\mathbf{z})$ can be used to determine whether $\mathbf{x}_{(i)}$ and $\mathbf{x}_{(j)}$ are similar or dissimilar to each other.

For triplet, the matrix $\mathbf{M}$ can be derived as follows.

Theorem 1: Denote by $\mathbf{t} = \left(\mathbf{x}_{(i)}, \mathbf{x}_{(j)}, \mathbf{x}_{(k)}\right)$ the test triplet and by $\mathbf{t}_l = (\mathbf{x}_{l,1}, \mathbf{x}_{l,2}, \mathbf{x}_{l,3})$ the l^{th} triplet in the training set. Let $\mathbf{T}_l = (\mathbf{x}_{l,1} - \mathbf{x}_{l,3})(\mathbf{x}_{l,1} - \mathbf{x}_{l,3})^T - (\mathbf{x}_{l,1} - \mathbf{x}_{l,2})(\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T$, and $\mathbf{T} = \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)^T - \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)^T$. For the decision function defined in (5), if we re-parameterize the Mahalanobis distance metric matrix $\mathbf{M}$ as

$$\begin{aligned} \mathbf{M} &= \sum\nolimits_l \alpha_l \mathbf{T}_l \\ &= \sum\nolimits_l \alpha_l \Big[(\mathbf{x}_{l,1} - \mathbf{x}_{l,3})(\mathbf{x}_{l,1} - \mathbf{x}_{l,3})^T \\ &\quad - (\mathbf{x}_{l,1} - \mathbf{x}_{l,2})(\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T\Big], \end{aligned} \tag{8}$$

then there is

$$\begin{aligned} \sum\nolimits_l \alpha_l K_T(\mathbf{t}_l, \mathbf{t}) &= \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)^T \mathbf{M} \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right) \\ &\quad - \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)^T \mathbf{M} \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right). \end{aligned}$$

Proof: Based on the definition of $K_T(\mathbf{t}_l, \mathbf{t})$ in (3), we have

$$\begin{aligned} &\sum\nolimits_l \alpha_l K_T(\mathbf{t}_l, \mathbf{t}) = \sum\nolimits_l \alpha_l \,\mathrm{tr}(\mathbf{T}_l \mathbf{T}) \\ &= \sum\nolimits_l \alpha_l \,\mathrm{tr}\Big(\mathbf{T}_l \Big(\left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)^T \\ &\quad - \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)^T\Big)^T\Big) \\ &= \sum\nolimits_l \alpha_l \,\mathrm{tr}\left(\mathbf{T}_l \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)^T\right) \\ &\quad - \sum\nolimits_l \alpha_l \,\mathrm{tr}\left(\mathbf{T}_l \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)^T\right) \\ &= \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)^T \left(\sum\nolimits_l \alpha_l \mathbf{T}_l\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right) \\ &\quad - \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)^T \left(\sum\nolimits_l \alpha_l \mathbf{T}_l\right)\left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right) \\ &= \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right)^T \mathbf{M} \left(\mathbf{x}_{(i)} - \mathbf{x}_{(k)}\right) \\ &\quad - \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right)^T \mathbf{M} \left(\mathbf{x}_{(i)} - \mathbf{x}_{(j)}\right). \end{aligned} \tag{9}$$

End of proof. ■

Clearly, equations (4)∼(9) provide us a new perspective to view and understand the distance metric matrix $\mathbf{M}$ under a kernel classification framework. Meanwhile, this perspective provides us new approaches for learning distance metric, which can be much easier and more efficient than the previous metric learning approaches. In the following, we introduce two kernel classification methods for metric learning: regularized kernel SVM and kernel logistic regression. Note that by modifying the construction of doublet or triplet set, using different kernel classifier models, or adopting different optimization algorithms, other new metric learning algorithms can also be developed under the proposed framework.

1) Kernel SVM-like Model: Given the doublet or triplet training set, an SVM-like model can be proposed to learn the distance metric:

$$\min_{\mathbf{M},b,\boldsymbol{\xi}} \; r(\mathbf{M}) + \rho(\boldsymbol{\xi}) \tag{10}$$

$$\text{s.t.} \quad f_l^{(d)}\left((\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,2}), b, \xi_l\right) \geq 0 \quad \text{(doublet set)}, \tag{11}$$

$$\text{or } f_l^{(t)}\left((\mathbf{x}_{l,1}-\mathbf{x}_{l,3})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,3}) -(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,2}), \xi_l\right) \geq 0 \text{ (triplet set)}, \tag{12}$$

$$\xi_l \geq 0, \tag{13}$$

where $r(\mathbf{M})$ is the regularization term, $\rho(\boldsymbol{\xi})$ is the margin loss term, the constraint $f_l^{(d)}$ can be any linear function of $(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})$, b, and ξ_l, and the constraint $f_l^{(t)}$ can be any linear function of $(\mathbf{x}_{l,1}-\mathbf{x}_{l,3})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,3}) - (\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})$ and ξ_l. To guarantee that (10) is convex, we can simply choose convex regularizer $r(\mathbf{M})$ and convex margin loss $\rho(\boldsymbol{\xi})$. By plugging (7) or (8) in the model in (10), we can employ the SVM and kernel methods to learn all α_l to obtain the matrix $\mathbf{M}$.

If we adopt the Frobenius norm to regularize $\mathbf{M}$ and the hinge loss penalty on ξ_l, the model in (10) would become the standard SVM. SVM and its variants have been extensively studied [31], [46], [47] and various algorithms have been proposed for large-scale SVM training [48], [49]. Thus, the SVM-like model in (10) can allow us to learn good metrics efficiently from large-scale training data.

2) Kernel logistic regression: Under the kernel logistic regression model (KLR) [50], we let $h_l = 1$ if the samples of doublet $\mathbf{z}_l$ belong to the same class and let $h_l = 0$ if the samples of it belong to different classes. Meanwhile, suppose that the label of a doublet $\mathbf{z}_l$ is unknown, and we can calculate the probability that $\mathbf{z}_l$'s label is 1 as follows:

$$P(p_l = 1|\mathbf{z}_l) = \frac{1}{1+\exp\left(\sum_i \alpha_i K_D(\mathbf{z}_i, \mathbf{z}_l) + b\right)}. \tag{14}$$

The coefficient vector $\boldsymbol{\alpha}$ and the bias b can be obtained by maximizing the following log-likelihood function:

$$(\boldsymbol{\alpha}, b) = \arg\max_{\boldsymbol{\alpha},b}\left\{ l(\boldsymbol{\alpha}, b) = \sum\nolimits_l h_l \ln P(p_l = 1|\mathbf{z}_l) + (1-h_l)\ln P(p_l = 0|\mathbf{z}_l)\right\}. \tag{15}$$

KLR is a powerful probabilistic approach for classification. By modeling metric learning as a KLR problem, we can easily use the existing KLR algorithms to learn the desired metric. Moreover, the variants and improvements of KLR, e.g., sparse KLR [51], can also be used to develop new metric learning methods.

D. Connections with LMNN, ITML, and LDML

The proposed kernel classification framework provides a unified explanation of many state-of-the-art metric learning methods. In this subsection, we show that LMNN and ITML can be considered as certain SVM models, while LDML is an example of the kernel logistic regression model.

1) LMNN: LMNN [8] learns a distance metric that penalizes both large distances between samples with the same label and small distances between samples with different labels. LMNN is operated on a set of triplets $\{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)\}$, where $\mathbf{x}_i$ has the same label as $\mathbf{x}_j$ but has different label from $\mathbf{x}_k$. The optimization problem of LMNN can be stated as follows:

$$\min_{\mathbf{M},\xi_{ijk}} \; \sum\nolimits_{i,j} (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) + C\sum\nolimits_{i,j,k}\xi_{ijk} \tag{16}$$

$$\text{s.t.} \quad (\mathbf{x}_i - \mathbf{x}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_k) - (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{ijk}, \tag{17}$$

$$\xi_{ijl} \geq 0, \tag{18}$$

$$\mathbf{M} \succcurlyeq 0. \tag{19}$$

Since $\mathbf{M}$ is required to be positive semi-definite in LMNN, we introduce the following indicator function:

$$\iota_{\succcurlyeq}(\mathbf{M}) = \begin{cases} 0, & \text{if } \mathbf{M} \succcurlyeq 0, \\ +\infty, & \text{otherwise}, \end{cases} \tag{20}$$

and choose the following regularizer and margin loss:

$$r_{\text{LMNN}}(\mathbf{M}) = \sum\nolimits_{i,j}(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) + \iota_{\succcurlyeq}(\mathbf{M}), \tag{21}$$

$$\rho_{\text{LMNN}}(\boldsymbol{\xi}) = C\sum\nolimits_{i,j,k}\xi_{ijk}. \tag{22}$$

Then we can define the following SVM-like model on the same triplet set:

$$\min_{\mathbf{M},\boldsymbol{\xi}} \; r_{\text{LMNN}}(\mathbf{M}) + \rho_{\text{LMNN}}(\boldsymbol{\xi}) \tag{23}$$

$$\text{s.t.} \quad (\mathbf{x}_i - \mathbf{x}_k)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_k) - (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{ijk}, \tag{24}$$

$$\xi_{ijk} \geq 0. \tag{25}$$

It is obvious that the SVM-like model in (23) is equivalent to the LMNN model in (16).

2) ITML: ITML [10] is operated on a set of doublets $\{(\mathbf{x}_i, \mathbf{x}_j)\}$ by solving the following minimization problem

$$\min_{\mathbf{M},\boldsymbol{\xi}} \; D_{ld}(\mathbf{M}, \mathbf{M}_0) + \gamma \cdot D_{ld}(\text{diag}(\boldsymbol{\xi}), \text{diag}(\boldsymbol{\xi}_0)) \tag{26}$$

$$\text{s.t.} \quad (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \leq \xi_{u(i,j)} \quad (i,j) \in \mathcal{S}, \tag{27}$$

$$(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geq \xi_{l(i,j)} \quad (i,j) \in \mathcal{D}, \tag{28}$$

$$\mathbf{M} \succcurlyeq 0, \tag{29}$$

where $\mathbf{M}_0$ is the given prior of the metric matrix, $\boldsymbol{\xi}_0$ is the given prior on $\boldsymbol{\xi}$, $\mathcal{S}$ is the set of doublets where $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same label, $\mathcal{D}$ is the set of doublets where $\mathbf{x}_i$ and $\mathbf{x}_j$ have different labels, and $D_{ld}(\cdot,\cdot)$ is the LogDet divergence of two matrices defined as:

$$D_{ld}(\mathbf{M}, \mathbf{M}_0) = \text{tr}\left(\mathbf{M}\mathbf{M}_0^{-1}\right) - \log\det\left(\mathbf{M}\mathbf{M}_0^{-1}\right) - n. \tag{30}$$

Davis *et al.* also proposed an iterative Bregman projection algorithm for ITML to avoid the positive semi-definite projection of the distance metric matrix $\mathbf{M}$ [10].

By introducing the following regularizer and margin loss:

$$r_{\text{ITML}}(\mathbf{M}) = D_{ld}(\mathbf{M}, \mathbf{M}_0) + \iota_{\succcurlyeq}(\mathbf{M}), \tag{31}$$

$$\rho_{\text{ITML}}(\boldsymbol{\xi}) = \gamma \cdot D_{ld}(\text{diag}(\boldsymbol{\xi}), \text{diag}(\boldsymbol{\xi}_0)), \tag{32}$$

we can then define the following SVM-like model on the same doublet set:

$$\min_{\mathbf{M},\boldsymbol{\xi}} \; r_{\text{ITML}}(\mathbf{M}) + \rho_{\text{ITML}}(\boldsymbol{\xi}) \tag{33}$$

$$\text{s.t.} \quad (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \le \xi_{u(i,j)} \;\; (i,j) \in \mathcal{S}, \tag{34}$$

$$(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \ge \xi_{l(i,j)} \;\; (i,j) \in \mathcal{D}, \tag{35}$$

$$\xi_{ij} \ge 0, \tag{36}$$

where $\mathbf{z}_{ij} = (\mathbf{x}_i, \mathbf{x}_j)$. One can easily see that the SVM-like model in (33) is equivalent to the ITML model in (26).

3) LDML: LDML [3] is a logistic discriminant based metric learning approach based on a set of doublets. Given a doublet $\mathbf{z}_l = (\mathbf{x}_{l(i)}, \mathbf{x}_{l(j)})$ and its label h_l, LDML defines the probability that $y_{l(i)} = y_{l(j)}$ as follows:

$$\begin{aligned} p_l &= P(y_{l(i)} = y_{l(j)} | \mathbf{x}_{l(i)}, \mathbf{x}_{l(j)}, \mathbf{M}, b) \\ &= \sigma(b - d_{\mathbf{M}}(\mathbf{x}_{l(i)}, \mathbf{x}_{l(j)})), \end{aligned} \tag{37}$$

where $\sigma(z)$ is the sigmoid function, b is the bias, and $d_{\mathbf{M}}(\mathbf{x}_{l(i)}, \mathbf{x}_{l(j)}) = (\mathbf{x}_{l(i)} - \mathbf{x}_{l(j)})^T \mathbf{M}(\mathbf{x}_{l(i)} - \mathbf{x}_{l(j)})$. With the p_l defined in (37), LDML learns $\mathbf{M}$ and b by maximizing the following log-likelihood:

$$\max_{\mathbf{M},b} \left\{ l(\mathbf{M}, b) = \sum\nolimits_l h_l \ln p_l + (1 - h_l)\ln(1 - p_l) \right\}. \tag{38}$$

Note that $\mathbf{M}$ is not constrained to be positive semi-definite in LDML.

With the same doublet set, let $\boldsymbol{\alpha}$ be the solution obtained by the kernel logistic model in (15), and $\mathbf{M}$ be the solution of LDML in (38). It is easy to see that:

$$\mathbf{M} = \sum\nolimits_l \alpha_l (\mathbf{x}_{l(i)} - \mathbf{x}_{l(j)})(\mathbf{x}_{l(i)} - \mathbf{x}_{l(j)})^T. \tag{39}$$

Thus, LDML is equivalent to kernel logistic regression under the proposed kernel classification framework.

IV. Metric Learning via SVM

The kernel classification framework proposed in Section III can not only generalize the existing metric learning models (as shown in Section III-D), but also be able to suggest new metric learning models. Actually, for both ITML and LMNN, the positive semi-definite constraint is imposed on $\mathbf{M}$ to guarantee that the learned distance metric is a Mahalanobis metric, which makes the models unable to be solved using the efficient kernel learning toolbox. In this section, a two-step greedy strategy is adopted for metric learning. We first neglect the positive semi-definite constraint and use the SVM toolbox to learn a preliminary matrix $\mathbf{M}$, and then map $\mathbf{M}$ onto the space of positive semi-definite matrices. The projected sub-gradient algorithm used in many metric learning methods [30] share similar spirits with the two-step greedy strategy. As examples, we present two novel metric learning methods, namely doublet-SVM and triplet-SVM, based on the proposed framework. Like in conventional SVM, we adopt the Frobenius norm to regularize $\mathbf{M}$ and employ the hinge loss penalty, and hence the doublet-SVM and triplet-SVM can be efficiently solved by using the standard SVM toolbox.

A. Doublet-SVM

In doublet-SVM, we set the Frobenius norm regularizer as $r_{\text{SVM}}(\mathbf{M}) = \frac{1}{2}\|\mathbf{M}\|_F^2$, and set $\rho_{\text{SVM}}(\boldsymbol{\xi}) = C\sum_l \xi_l$ as the margin loss term, resulting in the following model:

$$\min_{\mathbf{M},b,\boldsymbol{\xi}} \; \frac{1}{2}\|\mathbf{M}\|_F^2 + C\sum\nolimits_l \xi_l \tag{40}$$

$$\text{s.t.} \quad h_l\left((\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T \mathbf{M}(\mathbf{x}_{l,1} - \mathbf{x}_{l,2}) + b\right) \ge 1 - \xi_l, \tag{41}$$

$$\xi_l \ge 0, \; \forall l, \tag{42}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The Lagrange dual problem of the above doublet-SVM model is:

$$\max_{\boldsymbol{\alpha}} \; -\frac{1}{2}\sum\nolimits_{i,j} \alpha_i \alpha_j h_i h_j K_D(\mathbf{z}_i, \mathbf{z}_j) + \sum\nolimits_i \alpha_i \tag{43}$$

$$\text{s.t.} \quad 0 \le \alpha_l \le C, \quad \forall l, \tag{44}$$

$$\sum\nolimits_l \alpha_l h_l = 0, \tag{45}$$

which can be easily solved by many existing SVM solvers such as LibSVM [32]. The detailed deduction of the dual of doublet-SVM can be found in Appendix A.

B. Triplet-SVM

In triplet-SVM, we also choose $r_{\text{SVM}}(\mathbf{M}) = \frac{1}{2}\|\mathbf{M}\|_F^2$ as the regularization term, and choose $\rho_{\text{SVM}}(\boldsymbol{\xi}) = C\sum_l \xi_l$ as the margin loss term. Since the triplets do not have label information, we choose the linear inequality constraints which are adopted in LMNN, resulting in the following triplet-SVM model:

$$\min_{\mathbf{M},\boldsymbol{\xi}} \; \frac{1}{2}\|\mathbf{M}\|_F^2 + C\sum\nolimits_l \xi_l \tag{46}$$

$$\begin{aligned} \text{s.t.} \quad & (\mathbf{x}_{l,1} - \mathbf{x}_{l,3})^T \mathbf{M}(\mathbf{x}_{l,1} - \mathbf{x}_{l,3}) \\ & - (\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T \mathbf{M}(\mathbf{x}_{l,1} - \mathbf{x}_{l,2}) \ge 1 - \xi_l, \end{aligned} \tag{47}$$

$$\xi_l \ge 0, \; \forall l. \tag{48}$$

Actually, the proposed triplet-SVM can be regarded as a one-class SVM model, and the formulation of triplet-SVM is similar to the one-class SVM in [47]. The dual problem of triplet-SVM is:

$$\max_{\boldsymbol{\alpha}} \; -\frac{1}{2}\sum\nolimits_{i,j} \alpha_i \alpha_j K_T(\mathbf{t}_i, \mathbf{t}_j) + \sum\nolimits_i \alpha_i \tag{49}$$

$$\text{s.t.} \quad 0 \le \alpha_l \le C, \quad \forall l, \tag{50}$$

which can also be efficiently solved by existing SVM solvers [32]. The detailed deduction of the dual of triplet-SVM can be found in Appendix B.

C. Discussions

The matrix $\mathbf{M}$ learned by doublet-SVM and triplet-SVM may not be positive semi-definite. To learn a Mahalanobis distance metric, which requires $\mathbf{M}$ to be positive semi-definite, we can compute the singular value decomposition of $\mathbf{M} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues, and then preserve only the positive eigenvalues in $\boldsymbol{\Lambda}$ to form another diagonal matrix $\boldsymbol{\Lambda}_+$. Finally, we let $\mathbf{M}_+ = \mathbf{U}\boldsymbol{\Lambda}_+\mathbf{V}$ be the Mahalanobis metric matrix.

The proposed doublet-SVM and triplet-SVM are easy to implement since the use of Frobenius norm regularizer and hinge loss penalty allows us to readily employ the available SVM toolbox to solve them. A number of efficient algorithms, e.g., sequential minimal optimization [52], have been proposed for SVM training, making doublet-SVM and triplet-SVM very efficient to optimize. Moreover, using the large-scale SVM training algorithms [48], [49], [53], [54], we can easily extend doublet-SVM and triplet-SVM to deal with large-scale metric learning problems.

A number of kernel methods have been proposed for supervised learning [45]. With the proposed framework, we can easily couple them with the degree-2 polynomial kernel to develop new metric learning approaches. Semi-supervised, multiple instance, and multi-task metric learning have been investigated in [5], [26], [27], [55]. Fortunately, the proposed kernel classification framework can also allow us to develop such kind of metric learning approaches based on the recent progress of kernel methods for semi-supervised, multiple instance, and multitask learning [56]–[59]. Taking semi-supervised metric learning as an example, based on Laplacian SVM [56] and doublet-SVM, we can readily extend the kernel classification framework for semi-supervised metric learning.

Let $\{(\mathbf{z}_i, h_i)\}_{i=1}^{L}$ be a set of L labeled doublets, and $\{\mathbf{z}_i\}_{i=L+1}^{i=L+U}$ be a set of U unlabeled doublets. With the degree-2 polynomial kernel $K_D(\mathbf{z}_i, \mathbf{z}_j)$, the decision function can be expressed as:

$$f(\mathbf{z}) = \sum\nolimits_{i=1}^{L} \alpha_i h_i K_D(\mathbf{z}, \mathbf{z}_i) + \sum\nolimits_{i=L+1}^{L+U} \alpha_i K_D(\mathbf{z}, \mathbf{z}_i),$$

where $\mathbf{z} = \left(\mathbf{x}_{(j)}, \mathbf{x}_{(k)}\right)$, $\mathbf{z}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2})$. Analogous to Laplacian SVM, one can combine the Frobenius norm regularizer and the manifold regularizer:

$$r(f) = \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(U+L)^2} \mathbf{f}^T (\mathbf{D} - \mathbf{W}) \mathbf{f},$$

where $\|f\|_K$ denotes the norm in the kernel feature space, $f_i = \sum_{j=1}^{L+U} \alpha_j K_D(\mathbf{z}_i, \mathbf{z}_j)$, $\mathbf{f} = (f_1, \cdots, f_{L+U})^T$, $\mathbf{W}$ is introduced to model the adjacency between doublets with $W_{ij} = \exp((-K_D(\mathbf{z}_i, \mathbf{z}_i) + 2K_D(\mathbf{z}_i, \mathbf{z}_j) - K_D(\mathbf{z}_j, \mathbf{z}_j))/4t\ )$ (t is the constant parameter), where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_{j=1}^{L+U} W_{ij}$. By using hinge loss as the margin loss term $\rho(\boldsymbol{\xi})$ and introducing the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, semi-supervised metric learning can then be formulated as Laplacian SVM:

$$\begin{aligned} \min_{f}\ & \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathbf{f}^T \mathbf{L} \mathbf{f} + C \sum\nolimits_i \xi_i \\ \text{s.t.}\ & h_i (f(\mathbf{z}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0,\ i = 1, \cdots, L. \end{aligned}$$

The Lagrange dual problem of Laplacian SVM can be represented as

$$\begin{aligned} \min_{\boldsymbol{\alpha}}\ & \sum\nolimits_{i=1}^{L} \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} \\ \text{s.t.}\ & 0 \leq \alpha_i \leq C,\ i = 1, \cdots, L, \\ & \sum\nolimits_{i=1}^{L} \alpha_i h_i = 0, \end{aligned}$$

where $\mathbf{Q} = \mathbf{YJK}\left(2\gamma_A \mathbf{I} + 2\frac{\gamma_I}{(L+U)^2} \mathbf{LK}\right)^{-1} \mathbf{J}^T \mathbf{Y}$, $\mathbf{K}$ is the kernel Gram matrix with $K_{ij} = K_D(\mathbf{z}_i, \mathbf{z}_j)$, $\mathbf{Y}$ is an $(L+U) \times (L+U)$ diagonal matrix with $Y_{ii} = h_i$ when $i \leq L$ and 0 otherwise, J is an $(L+U) \times (L+U)$ diagonal matrix with $J_{ii} = 1$ when $i \leq L$ and 0 otherwise.

The above Laplacian SVM problem can be solved by the standard SVM solver [56]. Given the optimal solution on $\boldsymbol{\alpha}$, the positive semi-definite matrix $\mathbf{M}$ can be obtained by

$$\begin{aligned} \mathbf{M} = & \sum\nolimits_{i=1}^{L} \alpha_i h_i (\mathbf{x}_{i,1} - \mathbf{x}_{i,2}) (\mathbf{x}_{i,1} - \mathbf{x}_{i,2})^T \\ & + \sum\nolimits_{i=L+1}^{L+U} \alpha_i (\mathbf{x}_{i,1} - \mathbf{x}_{i,2}) (\mathbf{x}_{i,1} - \mathbf{x}_{i,2})^T. \end{aligned}$$

Similarly, one can extend the kernel classification framework for multiple instance and multi-task metric learning based on the multiple instance and multi-task kernel learning methods [57]–[59].

V. Experimental Results

In the experiments, we evaluate the proposed doublet-SVM and triplet-SVM for k-NN classification with $k = 1$ by using the UCI datasets and the handwritten digit datasets. We compare the proposed methods with five representative and state-of-the-art metric learning models, i.e., LMNN [8], ITML [10], LDML [3], neighbourhood component analysis (NCA) [9] and maximally collapsing metric learning (MCML) [2], in terms of classification error rate and training time (in seconds). We implemented doublet-SVM and triplet-SVM based on the popular SVM toolbox LibSVM[1] . The source codes of LMNN[2], ITML[3], LDML[4], NCA[5] and MCML[6] are online available, and we tuned their parameters to get the best results. The Matlab source code of our algorithm can be downloaded at http://www4.comp.polyu.edu.hk/∼cslzhang/PSML.v1.zip. In the training stage, the doublet set used in doublet-SVM is exactly the same as that used in ITML, but is different from that used in the other models, i.e., NCA, MCML, and LDML. The triplet set used in Triplet-SVM is different from that used in LMNN. The reason that we do not use the same doublet or triplet sets as LMNN, NCA, MCML, and LDML is that the released codes of these approaches either include inherent default doublet or triplet sets, or dynamically tune the doublet or triplet sets during the training stage.

A. UCI Dataset Classification

Ten datasets selected from the UCI machine learning repository [60] are used in the experiment. For the Statlog Satellite, SPECTF Heart and Letter datasets, we use the defined training and test sets to perform the experiment. For the other 7 datasets, we use 10-fold cross validation to evaluate the competing metric learning methods, and the reported error rate and training time are obtained by averaging over the 10

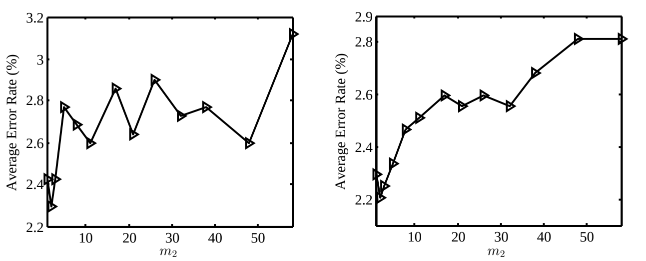
[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/
[2] http://www.cse.wustl.edu/∼kilian/code/code.html
[3] http://www.cs.utexas.edu/∼pjain/itml/
[4] http://lear.inrialpes.fr/people/guillaumin/code.php
[5] http://www.cs.berkeley.edu/∼fowlkes/software/nca/
[6] http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html

TABLE I
THE UCI DATASETS USED IN THE EXPERIMENT

Dataset	# of training samples	# of test samples	Feature dimension	# of classes
Parkinsons	176	19	22	2
Sonar	188	20	60	2
Statlog Segmentation	2,079	231	19	7
Breast Tissue	96	10	9	6
ILPD	525	58	10	2
Statlog Satellite	4,435	2,000	36	6
Blood Transfusion	674	74	4	2
SPECTF Heart	80	187	44	2
Cardiotocography	1,914	212	21	10
Letter	16,000	4,000	16	26

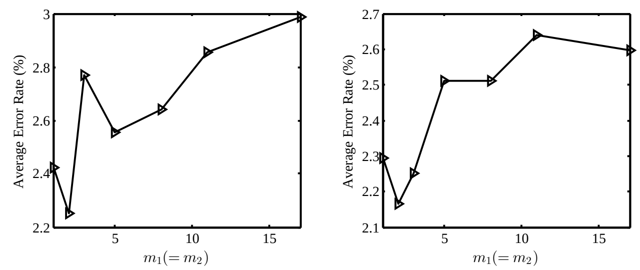
runs. Table I summarizes the basic information of the 10 UCI datasets.

Both doublet-SVM and triplet-SVM involve three hyper-parameters, i.e., m_1, m_2, and C. Using the Statlog Segmentation dataset as an example, we analyze the sensitivity of classification error rate to those hyper-parameters. By setting $m_1 = 1$ and $C = 1$, we investigate the influence of m_2 on classification performance. Fig. 1 shows the curves of classification error rate versus m_2 for doublet-SVM and triplet-SVM. One can see that both doublet-SVM and triplet-SVM achieve their lowest error rates when $m_2 = 2$. Moreover, the error rates tend to be a little higher when $m_2 > 3$. Thus, we set m_2 to $1 \sim 3$ in our experiments.



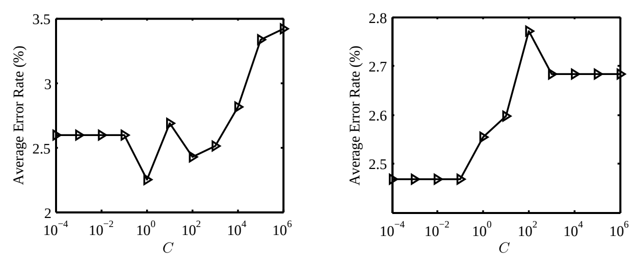
(a) (b)

Fig. 1. Classification error rate (%) versus m_2 for (a) doublet-SVM and (b) triplet-SVM with $m_1 = 1$ and $C = 1$.



(a) (b)

Fig. 2. Classification error rate (%) versus $m_1(= m_2)$ for (a) doublet-SVM and (b) triplet-SVM with $C = 1$.



(a) (b)

Fig. 3. Classification error rate (%) versus C for (a) doublet-SVM and (b) triplet-SVM with $m_1 = m_2 = 2$.

By setting $m_1 = m_2$, we study the influence of m_1 on classification error rate. The curves of error rate versus $m_1\ (= m_2)$ for doublet-SVM and triplet-SVM are shown in Fig. 2. One can see that, the lowest classification error is obtained when $m_1 = m_2 = 2$. Thus, we also set m_1 to $1 \sim 3$ in our experiments.

We further investigate the influence of C on the classification error rate by fixing $m_1 = m_2 = 2$. Fig. 3 shows the curves of classification error rate versus C for doublet-SVM and triplet-SVM. One can see that the error rate is insensitive to C in a wide range, but it jumps when C is no less than 10^4 for doublet-SVM and no less than 10^1 for triplet-SVM. Thus, we set $C < 10^4$ for doublet-SVM and $C < 10^1$ for triplet-SVM in our experiments.

Table II lists the classification error rates of the seven metric learning models on the 10 UCI datasets. On the Letter, ILPD and SPECTF Heart datasets, doublet-SVM obtains the lowest error rates. On the Statlog Segmentation dataset, triplet-SVM achieves the lowest error rate. In order to compare the recognition performance of these metric learning models, we list the average ranks of these models in the last row of Table II. On each dataset, we rank the methods based on their error rates, i.e., we assign rank 1 to the best method and rank 2 to the second best method, and so on. The average rank is defined as the mean rank of one method over the 10 datasets, which can provide a fair comparison of the algorithms [61].

From Table II, we can see that doublet-SVM achieves the best average rank and triplet-SVM achieves the fourth best

TABLE II
THE CLASSIFICATION ERROR RATES (%) AND AVERAGE RANKS OF THE COMPETING METHODS ON THE UCI DATASETS

Method	Doublet-SVM	Triplet-SVM	NCA	LMNN	ITML	MCML	LDML
Parkinsons	5.68	7.89	**4.21**	5.26	6.32	12.94	7.15
Sonar	13.07	14.29	14.43	**11.57**	14.86	24.29	22.86
Statlog Segmentation	2.42	**2.29**	2.68	2.64	**2.29**	2.77	2.86
Breast Tissue	38.37	33.37	**30.75**	34.37	36.75	**30.75**	48.00
ILPD	**32.09**	35.16	34.79	34.12	33.59	34.79	35.84
Statlog Satellite	10.80	10.75	10.95	**10.05**	11.30	15.65	15.90
Blood Transfusion	29.47	34.37	**28.38**	28.78	31.51	31.89	31.40
SPECTF Heart	**27.27**	33.69	38.50	34.76	35.29	29.95	33.16
Cardiotocography	20.71	19.34	21.84	**19.21**	19.90	20.76	22.26
Letter	**2.47**	2.77	**2.47**	3.45	2.78	4.20	11.05
Average Rank	***2.70***	*3.70*	*3.40*	*2.80*	*4.00*	*5.00*	*6.00*

average rank. The results validate that, by incorporating the degree-2 polynomial kernel into the standard (one-class) kernel SVM classifier, the proposed kernel classification based metric learning framework can lead to very competitive classification accuracy with state-of-the-art metric learning methods. It is interesting to see that, although doublet-SVM outperforms triplet-SVM on most datasets, triplet-SVM works better than doublet-SVM on large scale datasets like Statlog Segmentation, Statlog Satellite and Cardiotocography, and achieves very close error rate to doublet-SVM on the large dataset Letter. These results may indicate that doublet-SVM is more effective for small scale datasets, while triplet-SVM is more effective for large scale datasets, where each class has many training samples. Our experimental results on the three large scale handwritten digit datasets in Section V-B will further verify this.

Let's then compare the training time of the proposed methods and the competing methods. All the experiments are executed in a PC with 4 Intel Core i5-2410 CPUs (2.30 GHz) and 16 GB RAM. Note that in the training stage, doublet-SVM, ITML, LDML, MCML, and NCA are operated on the doublet set, while triplet-SVM and LMNN are operated on the triplet set. Thus, we compare the five doublet-based metric learning methods and the two triplet-based methods, respectively. Fig. 4 compares the training time of doublet-SVM, ITML, LDML, MCML, and NCA. Clearly, doublet-SVM is always the fastest algorithm and it is much faster than the other four methods. In average, it is 2,000 times faster than the second fastest algorithm, ITML. Fig. 5 compares the training time of triplet-SVM and LMNN. One can see that triplet-SVM is about 100 times faster than LMNN on the ten data sets.

B. Handwritten Digit Recognition

Apart from the UCI datasets, we also perform experiments on three widely used large scale handwritten digit sets, i.e., MNIST, USPS, and Semeion, to evaluate the performance of doublet-SVM and triplet-SVM. On the MNIST and USPS datasets, we use the defined training and test sets to train the models and calculate the classification error rates. On the

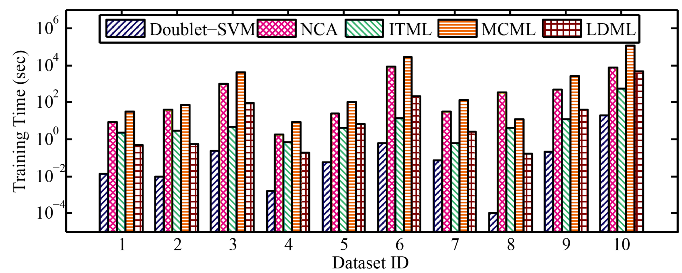


Fig. 4. Training time (sec.) of doublet-SVM, NCA, ITML, MCML and LDML. From 1 to 10, the Dataset ID represents Parkinsons, Sonar, Statlog Segmentation, Breast Tissue, ILPD, Statlog satellite, Blood Transfusion, SPECTF Heart, Cardiotocography, and Letter.

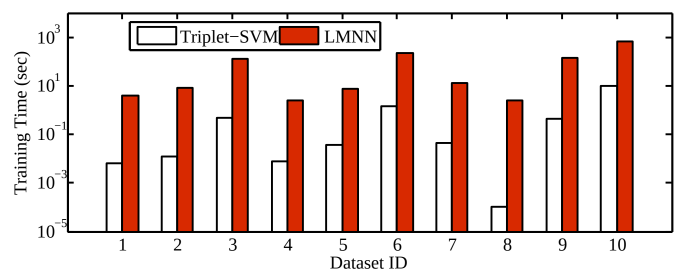


Fig. 5. Training time (sec.) of triplet-SVM and LMNN. From 1 to 10, the Dataset ID represents Parkinsons, Sonar, Statlog Segmentation, Breast Tissue, ILPD, Statlog satellite, Blood Transfusion, SPECTF Heart, Cardiotocography, and Letter.

Semeion datasets, we use 10-fold cross validation to evaluate the metric learning methods, and the error rate and training time are obtained by averaging over the 10 runs. Table III summarizes the basic information of the three handwritten digit datasets.

As the dimensions of digit images are relatively high, PCA is utilized to reduce the feature dimension. The metric learning models are trained in the PCA subspace. Table IV lists the classification error rates on the handwritten digit datasets. On the MNIST dataset, LMNN achieves the lowest error rate; on the USPS dataset, doublet-SVM achieves the lowest error rate; and on the Semeion dataset, triplet-SVM obtains the lowest

TABLE III
THE HANDWRITTEN DIGITS DATASETS USED IN THE EXPERIMENTS

Dataset	# of training samples	# of test samples	Feature dimension	PCA dimension	# of classes
MNIST	60,000	10,000	784	100	10
USPS	7,291	2,007	256	100	10
Semeion	1,434	159	256	100	10

TABLE IV
THE CLASSIFICATION ERROR RATES (%) AND AVERAGE RANKS OF THE COMPETING METHODS ON THE HANDWRITTEN DIGIT DATASETS

Dataset	Doublet-SVM	Triplet-SVM	NCA	LMNN	ITML	MCML	LDML
MNIST	3.19	2.92	5.46	**2.28**	2.89	-	6.05
USPS	**5.03**	5.23	5.68	5.38	6.63	5.08	8.77
Semeion	5.09	**4.71**	8.60	6.09	5.71	11.23	11.98
Average Rank	*2.33*	***2.00***	*4.67*	*2.67*	*3.33*	-	*6.00*

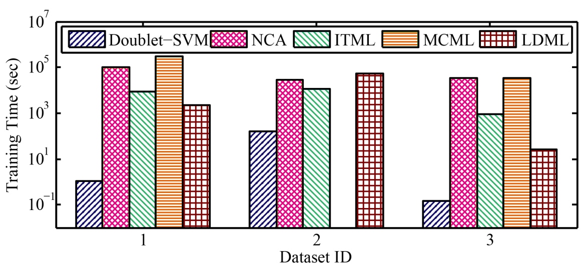


Fig. 6. Training time (sec.) of doublet-SVM, NCA, ITML, MCML and LDML. From 1 to 3, the Dataset ID represents USPS, MNIST and Semeion.

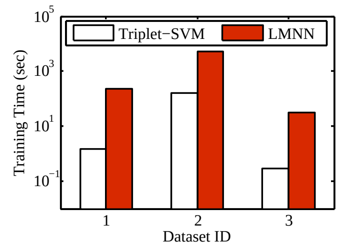


Fig. 7. Training time (sec.) of triplet-SVM and LMNN. From 1 to 3, the Dataset ID represents USPS, MNIST and Semeion.

error rate. We do not report the error rate of MCML on the MNIST dataset because MCML requires too large memory space (more than 30 GB) on this dataset and cannot be run in our PC.

The last row of Table IV lists the average ranks of the seven metric learning models. We can see that triplet-SVM can achieve the best average rank, and doublet-SVM achieves the second best average rank. The results further validate that on large scale datasets where each class has sufficient number of training samples, triplet-SVM would be superior to doublet-SVM and the competing methods.

We then compare the training time of these metric learning methods. All the experiments are executed in the same PC as the experiments in Section V-A. We compare the five doublet-based metric learning methods and the two triplet-based methods, respectively. Fig. 6 shows the training time of doublet-SVM, ITML, LDML, MCML, and NCA. We can see that doublet-SVM is much faster than the other four methods. In average it is 2,000 times faster than the second fastest algorithm, ITML. Fig. 7 shows the training time of triplet-SVM and LMNN. One can see that triplet-SVM is about 100 times faster than LMNN on the three datasets.

C. Doublets/Triplets Construction

Let's first compare the classification performance by using different strategies to construct the doublet set. Using Doublet-SVM as an example, we consider the following two strategies to construct the doublet set:

(i) *Nearest neighbor (NN) selection*: As described in Section III.A, for each training sample $\mathbf{x}_i$, we construct $m_1 + m_2$ doublets $\{(\mathbf{x}_i, \mathbf{x}^s_{i,1}), \cdots, (\mathbf{x}_i, \mathbf{x}^s_{i,m_1}), (\mathbf{x}_i, \mathbf{x}^d_{i,1}), \cdots, (\mathbf{x}_i, \mathbf{x}^d_{i,m_2})\}$, where $\mathbf{x}^s_{i,k}$ denotes the k^{th} similar nearest neighbor of $\mathbf{x}_i$, and $\mathbf{x}^d_{i,k}$ denotes the k^{th} dissimilar nearest neighbor of $\mathbf{x}_i$. By constructing all such doublets from the training samples, we build a doublet set using the NN strategy.

(ii) *Random selection*: Given a training set of n samples, we randomly select $(m_1 + m_2)\, n$ doublets from all the $n(n-1)$ possible doublets.

Tables V and VI list the classification error rates of doublet-SVM by using the NN and the random selection strategies to construct the doublet set. The NN selection outperforms the random selection on 7 out of the 10 UCI datasets and on all the three handwritten digit datasets. One can conclude that for doublet-SVM, the NN selection is better than the random selection to construct doublet set.

We then compare the classification performance by using different strategies to construct the triplet set. Using Triplet-SVM as an example, we also consider the NN selection and random selection strategies to construct triplet set:

(i) *Nearest neighbor (NN) selection*: For each training sample $\mathbf{x}_i$, we construct $m_1 m_2$ triplets $\{(\mathbf{x}_i, \mathbf{x}_{i,j}^s, \mathbf{x}_{i,k}^d) | j = 1, \ldots, m_1, k = 1, \ldots, m_2\}$, where $\mathbf{x}_{i,j}^s$ denotes the j^{th} similar nearest neighbor of $\mathbf{x}_i$, and $\mathbf{x}_{i,k}^d$ denotes the k^{th} dissimilar nearest neighbor of $\mathbf{x}_i$. By constructing all such triplets from the training samples, we build a triplet set using the NN strategy.

(ii) *Random selection*: Given a training set of n samples, we randomly select $(m_1 m_2)n$ triplets from all the $n(n-1)(n-2)$ possible triplets.

Tables V and VI list the classification error rates of triplet-SVM by using the NN and the random selection strategies. The NN selection outperforms the random selection on 9 out of the 10 UCI datasets and 2 out of the 3 handwritten digit datasets. One can conclude that the NN selection strategy is also a better choice than the random selection strategy for triplet-SVM to construct triplet sets.

D. Statistical Tests

Based on the classification error rates listed in Tables II and IV and following the statistical test setting in [61], we perform the Bonferroni-Dunn test [62] at the significance level $p = 0.05$. The results are shown in Fig. 8. The Bonferroni-Dunn test results indicate that the classification performance of Doublet-SVM and Triplet-SVM is statistically better than that of LDML at $p = 0.05$, but there is no statistically significant difference between the classification performance of Doublet-SVM, Triplet-SVM and the other 4 methods.

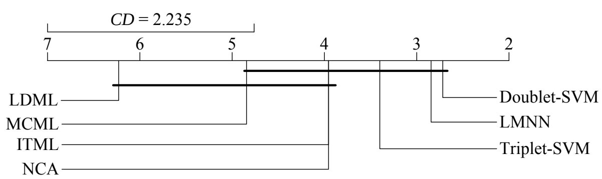


Fig. 8. Performance comparison of the seven metric learning methods using the Bonferroni-Dunn test. Groups of methods that are not significantly different (at $p = 0.05$) are connected. CD refers to the critical difference between the average ranks of two methods.

VI. Conclusion

In this paper, we proposed a general kernel classification framework for distance metric learning. By coupling a degree-2 polynomial kernel with some kernel methods, the proposed framework can unify many representative and state-of-the-art metric learning approaches such as LMNN, ITML and LDML. The proposed framework also provides a good platform for developing new metric learning algorithms. Two metric learning methods, i.e., doublet-SVM and triplet-SVM, were developed and they can be efficiently implemented by the standard SVM solvers. Our experimental results on the UCI datasets and handwritten digit datasets showed that doublet-SVM and triplet-SVM are much faster than state-of-the-art methods in terms of training time, while they achieve very competitive results in terms of classification error rate.

The proposed kernel classification framework provides a new perspective on developing metric learning methods via kernel classifiers. By incorporating the kernel learning methods for semi-supervised learning, multiple instance learning, etc., the proposed framework can be adopted to develop metric learning approaches for many other applications. By replacing the degree-2 polynomial kernel with nonlinear kernel functions which satisfy the Mercer's condition [45], the proposed framework can also be extended to nonlinear metric learning.

One limitation of the proposed doublet-SVM and triplet-SVM is that a two-step greedy strategy is used to learn the positive semi-definite matrix $\mathbf{M}$, and the solution is not globally optimal. In the future, we will study global optimization algorithms for the proposed kernel classification framework, and develop nonlinear metric learning methods.

Appendix A
The dual of doublet-SVM

According to the original problem of doublet-SVM in (40), its Lagrangian can be defined as follows:

$$\begin{aligned} L(\mathbf{M}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2}\|\mathbf{M}\|_F^2 + C\sum\nolimits_l \xi_l \\ &- \sum\nolimits_l \alpha_l \left[h_l \left((\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T \mathbf{M} (\mathbf{x}_{l,1} - \mathbf{x}_{l,2}) + b \right) - 1 + \xi_l \right] \\ &- \sum\nolimits_l \beta_l \xi_l, \end{aligned} \tag{51}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the Lagrange multipliers which satisfy $\alpha_l \geq 0$ and $\beta_l \geq 0$, $\forall l$. To convert the original problem to its dual, we let the derivative of the Lagrangian with respect to $\mathbf{M}$, b and $\boldsymbol{\xi}$ to be $\mathbf{0}$:

$$\begin{aligned} &\frac{\partial L(\mathbf{M}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{M}} = \mathbf{0} \Rightarrow \\ &\mathbf{M} - \sum\nolimits_l \alpha_l h_l (\mathbf{x}_{l,1} - \mathbf{x}_{l,2})(\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T = \mathbf{0}, \end{aligned} \tag{52}$$

$$\frac{\partial L(\mathbf{M}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = 0 \Rightarrow \sum\nolimits_l \alpha_l h_l = 0, \tag{53}$$

$$\begin{aligned} &\frac{\partial L(\mathbf{M}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_l} = 0 \Rightarrow \\ &C - \alpha_l - \beta_l = 0 \Rightarrow 0 < \alpha_l < C, \ \forall l. \end{aligned} \tag{54}$$

Equation (52) implies the relationship between $\mathbf{M}$ and $\boldsymbol{\alpha}$ as follows:

$$\mathbf{M} = \sum\nolimits_l \alpha_l h_l (\mathbf{x}_{l,1} - \mathbf{x}_{l,2})(\mathbf{x}_{l,1} - \mathbf{x}_{l,2})^T. \tag{55}$$

Substituting (52)~(54) back into the Lagrangian, we get the Lagrange dual problem of doublet-SVM as follows:

$$\max_{\boldsymbol{\alpha}} \ -\frac{1}{2}\sum\nolimits_{i,j} \alpha_i \alpha_j h_i h_j K_D(\mathbf{z}_i, \mathbf{z}_j) + \sum\nolimits_i \alpha_i \tag{56}$$

$$\text{s.t.} \quad 0 \leq \alpha_l \leq C, \quad \forall l, \tag{57}$$

$$\sum\nolimits_l \alpha_l h_l = 0. \tag{58}$$

TABLE V
THE CLASSIFICATION ERROR RATES (%) BY USING THE RANDOM SELECTION STRATEGY AND THE NN SELECTION STRATEGY FOR SELECTING DOUBLETS/TRIPLETS ON THE UCI DATASETS

Method	Doublet-SVM (Random)	Doublet-SVM (NN)	Triplet-SVM (Random)	Triplet-SVM (NN)
Parkinsons	6.92	**5.68**	8.48	**7.89**
Sonar	15.46	**13.07**	15.04	**13.93**
Statlog Segmentation	2.46	**2.12**	2.80	**2.29**
Breast Tissue	**38.03**	38.38	36.78	**33.37**
ILPD	**33.49**	34.48	33.20	**31.57**
Statlog Satellite	11.07	**11.05**	11.17	**10.80**
Blood Transfusion	34.40	**33.63**	**32.73**	33.91
SPECTF Heart	29.52	**27.27**	30.27	**28.34**
Cardiotocography	20.01	**18.59**	20.41	**19.63**
Letter	8.24	**2.80**	6.38	**2.95**

TABLE VI
THE CLASSIFICATION ERROR RATES (%) BY USING THE RANDOM SELECTION STRATEGY AND THE NN SELECTION STRATEGY TO SELECT DOUBLETS/TRIPLETS ON THE HANDWRITTEN DIGIT DATASETS

Method	Doublet-SVM (Random)	Doublet-SVM (NN)	Triplet-SVM (Random)	Triplet-SVM (NN)
MNIST	3.41	**3.19**	3.84	**2.92**
USPS	5.49	**5.43**	**5.65**	5.78
Semeion	6.43	**5.09**	6.96	**4.71**

APPENDIX B
THE DUAL OF TRIPLET-SVM

According to the original problem of triplet-SVM in (46), its Lagrangian can be defined as follows:

$$\begin{aligned} & L(\mathbf{M}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{M}\|_F^2 + C\sum\nolimits_l \xi_l \\ & -\sum\nolimits_l \alpha_l \Big[(\mathbf{x}_{l,1}-\mathbf{x}_{l,3})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,3}) \\ & -(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\mathbf{M}(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})\Big] \\ & +\sum\nolimits_l \alpha_l - \sum\nolimits_l \alpha_l\xi_l - \sum\nolimits_l \beta_l\xi_l, \end{aligned} \tag{59}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the Lagrange multipliers, which satisfy $\alpha_l \geq 0$ and $\beta_l \geq 0, \ \forall l$. To convert the original problem to its dual, we let the derivative of the Lagrangian with respect to $\mathbf{M}$ and $\boldsymbol{\xi}$ to be $\mathbf{0}$:

$$\begin{aligned} & \frac{\partial L(\mathbf{M}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{M}} = \mathbf{0} \Rightarrow \\ & \mathbf{M} - \sum\nolimits_l \alpha_l \Big[(\mathbf{x}_{l,1}-\mathbf{x}_{l,3})(\mathbf{x}_{l,1}-\mathbf{x}_{l,3})^T \\ & -(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\Big] = \mathbf{0}, \end{aligned} \tag{60}$$

$$\begin{aligned} & \frac{\partial L(\mathbf{M}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_l} = 0 \Rightarrow \\ & C - \alpha_l - \beta_l = 0 \Rightarrow 0 < \alpha_l < C, \ \forall l. \end{aligned} \tag{61}$$

Equation (60) implies the relationship between $\mathbf{M}$ and $\boldsymbol{\alpha}$ as follows:

$$\begin{aligned} & \mathbf{M} = \sum\nolimits_l \alpha_l \Big[(\mathbf{x}_{l,1}-\mathbf{x}_{l,3})(\mathbf{x}_{l,1}-\mathbf{x}_{l,3})^T \\ & -(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})(\mathbf{x}_{l,1}-\mathbf{x}_{l,2})^T\Big]. \end{aligned} \tag{62}$$

Substituting (60) and (61) back into the Lagrangian, we get the Lagrange dual problem of triplet-SVM as follows:

$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\sum\nolimits_{i,j} \alpha_i\alpha_j K_T(\mathbf{t}_i, \mathbf{t}_j) + \sum\nolimits_i \alpha_i \tag{63}$$

$$\text{s.t.} \quad 0 \leq \alpha_l \leq C, \quad \forall l. \tag{64}$$

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous reviewers for their constructive suggestions.

REFERENCES

[1] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2002, pp. 505–512.

[2] A. Globerson and S. Roweis, "Metric learning by collapsing classes," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 451–458.

[3] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? metric learning approaches for face identification," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV 2009)*, 2009, pp. 498–505.

[4] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 1–26, 2012.

[5] S. Hoi, W. Liu, and S. Chang, "Semi-supervised distance metric learning for collaborative image retrieval," in *Proc. 2008 IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR 2008)*, 2008, pp. 1–7.

[6] W. Bian and D. Tao, "Constrained empirical risk minimization framework for distance metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1194–1205, 2012.

[7] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *Proc. 10th Eur. Conf. Comput. Vis. (ECCV 2008)*, 2008, pp. 548–561.

[8] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 1473–1480.

[9] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2004, pp. 513–520.

[10] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML 2007)*, 2007, pp. 209–216.

[11] K. Huang, Y. Ying, and C. Campbell, "Gsml: A unified framework for sparse metric learning," in *Proc. 9th IEEE Int. Conf. Data Mining*, 2009, pp. 189–198.

[12] C. Shen, J. Kim, L. Wang, and A. Hengel, "Positive semidefinite metric learning with boosting," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 1651–1659.

[13] Q. Qian, R. Jin, J. Yi, L. Zhang, and S. Zhu, "Efficient distance metric learning by adaptative sampling and mini-batch stochastic gradient descent (sdd)," *arXiv: 1304.1192v1*, 2013.

[14] J. Wang, A. Woznica, and A. Kalousis, "Learning neighborhoods for metric learning," *arXiv: 1206.6883v1*, 2012.

[15] C. Shen, J. Kim, and L. Wang, "Scalable large-margin mahalanobis distance metric learning," *IEEE Trans. Neural Netw.*, vol. 21, no. 9, pp. 1524–1530, 2010.

[16] M. F. Balcan, A. Blum, and N. Srebro, "A theory of learning with similarity functions," *Mach. Learn.*, vol. 72, no. 1-2, pp. 89–112, Aug. 2008.

[17] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "An online algorithm for large scale image similarity learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 306–314.

[18] A. Bellet, A. Habrard, and M. Sebban, "Good edit similarity learning by loss minimization," *Mach. Learn.*, vol. 89, no. 1–2, pp. 5–35, Oct. 2012.

[19] Y. Fu, S. Yan, and T. S. Huang, "Correlation metric for generalized feature extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 12, pp. 2229–2235, Dec. 2008.

[20] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proc. 10th Asian Conf. Comp. Vis.*, 2010, pp. 709–720.

[21] J. Wang, H. Do, A. Woznica, and A. Kalousis, "Metric learning with multiple kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1170–1178.

[22] D. Kedem, S. Tyree, K. Weinberger, F. Sha, and G. Lanckriet, "Non-linear metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2582–2590.

[23] P. Jain, B. Kulis, J. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 519–547, Jan. 2012.

[24] G. Niu, B. Dai, M. Yamada, and M. Sugiyama, "Information-theoretic semi-supervised metric learning via entropy regularization," in *Proc. 29th Int. Conf. Mach. Learn. (ICML 2012)*, 2012, pp. 89–96.

[25] Q. Wang, P. C. Yuen, and G. Feng, "Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions," *Pattern Recognition*, vol. 46, no. 9, pp. 2576–2587, Mar. 2013.

[26] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," in *Proc. 11th European Conf. Comp. Vis.*, 2010, pp. 634–647.

[27] S. Parameswaran and K. Weinberger, "Large margin multi-task metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1867–1875.

[28] P. Yang, K. Huang, and C. L. Liu, "A multi-task framework for metric learning with common subspace," *Neural Computing Applicat.*, vol. 22, no. 7–8, pp. 1337–1347, June 2012.

[29] K. Q. Weinberger and L. K. Saul, "Fast solvers and efficient implementations for distance metric learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML 2008)*, 2008, pp. 1160–1167.

[30] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[31] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag, 1995.

[32] C. C. Chang and C. J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1–27, 2011.

[33] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "An invariant large margin nearest neighbour classifier," in *Proc. 11th Int. Conf. Comp. Vis.*, 2007, pp. 1–8.

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th European Conf. Comp. Vis.*, 2010, pp. 213–226.

[35] B. Kulis, K. Saenko, and T. Darrel, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Int. Conf. Comp. Vis. Patt. Recogn.*, 2011, pp. 1785–1792.

[36] R. G. Cinbis, J. Verbeek, and C. Schmid, "Unsupervised metric learning for face identification in tv video," in *Proc. 2011 IEEE Int. Conf. Comp. Vis.*, 2011, pp. 1559–1566.

[37] M. Kostinger, M. Hirzer, P. Wohlhart, P. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. 2012 IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR 2012)*, 2012, pp. 2288–2295.

[38] M. Hirzer, P. M. Roth, M. Kostinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," in *Proc. 12th European Conf. Comp. Vis.*, 2012, pp. 780–793.

[39] C. Shen, J. Kim, and L. Wang, "A scalable dual approach to semidefinite metric learning," in *Proc. 2011 IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR 2011)*, 2011, pp. 2601–2608.

[40] C. Shen, J. Kim, L. Wang, and A. Hengel, "Positive semidefinite metric learning using boosting-like algorithms," *J. Mach. Learn. Res.*, vol. 13, pp. 1007–1036, 2012.

[41] M. Liu, B. C. Vemuri, S. I. Amari, and F. Nielsen, "Shape retrieval using hierarchical total bregman soft clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2407–2419, Dec. 2012.

[42] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng, "Online and batch learning of pseudo-metrics," in *Proc. 21th Int. Conf. Mach. Learn.*, 2004, pp. 94–101.

[43] C. Brunner, A. Fischer, K. Luig, and T. Thies, "Pairwise support vector machines and their applications to large scale problems," *J. Mach. Learn. Res.*, vol. 13, pp. 2279–2292, 2012.

[44] A. Perez-Suay, F. J. Ferri, M. Arevalillo-Herraez, and J. V. Albert, "Comparative evaluation of batch and online distance metric learning approaches based on margin maximization," in *Proc. IEEE Int. Conf. Syst. Man. Cybern.*, 2013, pp. 3511–3515.

[45] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge Univ. Press, 2004.

[46] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, 2001.

[47] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[48] R. Collobert, S. Bengio, and Y. Bengio, "A parallel mixture of svms for very large scale problems," *Neural Computation*, vol. 14, no. 5, pp. 1105–1114, May 2002.

[49] I. W. Tsang, J. T. Kwok, and P. M. Cheung, "Core vector machines: Fast svm training on very large data sets," *J. Mach. Learn. Res.*, vol. 3, pp. 363–392, 2005.

[50] S. S. Keerthi, K. B. Duan, S. K. Shevade, and A. N. Poo, "A fast dual algorithm for kernel logistic regression," *Mach. Learn.*, vol. 61, no. 1-3, pp. 151–165, Nov. 2005.

[51] K. Koh, S. Kim, and S. Boyd, "An interior-point method for large-scale l1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, no. 8, pp. 1519–1555, Dec. 2007.

[52] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp. 185–208.

[53] J. Dong, A. Krzyzak, and C. Y. Suen, "Fast svm training algorithm with decomposition on very large data sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 603–618, Apr. 2005.

[54] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, *Large-scale kernel machines*. Cambridge, MA: MIT Press, 2007.

[55] M. S. Baghshah and S. B. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1217–1222.

[56] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[57] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 561–568.

[58] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "Learning sparse kernel classifiers for multi-instance classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1377–1389, 2013.

[59] T. Evgeniou, F. France, and M. Pontil, "Regularized multi-task learning," in *Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2004, pp. 109–117.

[60] A. Frank and A. Asuncion, "Uci machine learning repository [http://archive.ics.uci.edu/ml]," 2010.

[61] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[62] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Statist. Assoc.*, vol. 56, pp. 52–64, 1961.