A Kernel Classification Framework for Metric Learning

Faqiang Wang, Wangmeng Zuo, Member, IEEE, Lei Zhang, 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, which can not only generalize many popular metric learning methods such as LMNN and ITML, but 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 such as ITML and LMNN 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 their 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 plays a crucial role in improving 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 suggested for nonlinear distance metric learning [21], [22]. Extensions of metric learning have also been investigated for semi-supervised learning [5], [23], [24], multiple instance learning [25], and multi-task learning [26], [27], 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 unified 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 for metric learning, which can unify most state-of-the-art metric learning methods, such as large margin nearest neighbor (LMNN) [8], [28], [29], 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) [30]. 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 [31].

The remainder of the 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 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 [32]. Huq 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 [33], while Kulis et al. showed that in visual category recognition tasks, asymmetric transform would achieve better classification performance [34]. Cinbis et al. adapted LDML to unsupervised metric learning for face identification with uncontrolled video data [35]. Several relaxed pairwise metric learning methods have been developed for efficient Mahalanobis metric learning [36], [37].

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], [38]. Actually, such boosting-like approaches usually represent the metric matrix M as a linear combination of rank-one matrices [39]. Liu and Vemuri proposed a doubly regularized metric learning method by incorporating two regularization terms in the dual problem [40]. Shalev-Shwartz et al. proposed a pseudo-metric online learning algorithm (POLA) to learn distance metric in the kernel space [41]. Besides, a number of pairwise SVM methods have been proposed to learn distance metrics or nonlinear distance functions [42].

In this paper, we will see that most of the aforementioned metric learning approaches can be unified into the proposed 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.

III. A KERNEL CLASSIFICATION BASED METRIC LEARNING FRAMEWORK

Current metric learning models largely depend on convex or non-convex optimization techniques, some of which can be very inefficient to use in solving large-scale problems. In this section, we present a kernel classification framework which can unify many state-of-the-art metric learning methods, and make the metric learning task significantly more efficient. 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 cast as a conventional supervised learning problem. To build a connection between the two problems, we model metric learning as a kind of supervised learning problem operating on a set of doublets or triplets, as described below.

Let $D = \{(x_i, y_i) | i = 1, 2, \ldots, n\}$ be a training dataset, where vector $x_i \in \mathbb{R}^d$ represents the $i$th training sample, and scalar $y_i$ represents the class label of $x_i$. Any two samples extracted from $D$ can form a doublet $(x_i, x_j)$, 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 $x_i$, we find from $D$ its $m_1$ nearest similar neighbors, denoted by $\{x_{i,1}, \ldots, x_{i,m_1}\}$, and its $m_2$ nearest dissimilar neighbors, denoted by $\{x_{i,1}^d, \ldots, x_{i,m_2}^d\}$, and then construct $(m_1 + m_2)$ doublets $\{(x_i, x_{i,1}^d), \ldots, (x_i, x_{i,m_1}), (x_i, x_{i,1}), \ldots, (x_i, x_{i,m_2})\}$. By combining all such doublets constructed from all training samples, we build a doublet set, denoted by $\{z_{1}, \ldots, z_{N_d}\}$, where $z_i = (x_{i,1}, x_{i,2}), l = 1, 2, \ldots, N_d$. The label of doublet $z_l$ is denoted by $h_i$. 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 $(x_i, x_j, x_k)$ a triplet if three samples $x_i, x_j$, and $x_k$ are from $D$ 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 $x_i$, we find its $m_1$ nearest neighbors $\{x_{i,1}, \ldots, x_{i,m_1}\}$ which have the same class label as $x_i$, and $m_2$ nearest neighbors $\{x_{i,1}^d, \ldots, x_{i,m_2}^d\}$ which have different class labels from $x_i$. We can thus construct $m_1m_2$ triplets $\{(x_i, x_{i,j}, x_{i,k}) | j = 1, \ldots, m_1; k = 1, \ldots, m_2\}$ for each sample $x_i$. By combining all the triplets together, we form a triplet set $\{t_1, \ldots, t_{N_t}\}$, where $t_l = (x_{i,1}, x_{i,2}, x_{i,3}), l = 1, 2, \ldots, N_t$. Note that for the convenience of expression, here we remove the super-script “s” and “d” from $x_{i,2}$ and $x_{i,3}$, 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 $x_i$ and $x_j$, we define the following function:

$$K_p(x_i, x_j) = \text{tr}(x_i^T x_j^T x_j x_i^T) \quad (1)$$

where $\text{tr}(\bullet)$ represents the trace operator of a matrix. One can easily see that $K_p(x_i, x_j) = (x_i^T x_j)^2$ is a degree-2 polynomial kernel, and $K_p(x_i, x_j)$ satisfies the Mercer’s condition [43].

The kernel function defined in (1) can be extended to a pair of doublets or triplets. Given two doublets $z_i = (x_{i,1}, x_{i,2})$ and $z_j = (x_{j,1}, x_{j,2})$, we define the corresponding degree-2 polynomial kernel as

$$K_p(z_i, z_j) = \text{tr} \left( \begin{bmatrix} (x_{i,1} - x_{j,1})(x_{i,1} - x_{j,1})^T \\ (x_{j,1} - x_{j,2})(x_{j,1} - x_{j,2})^T \end{bmatrix} \right). \quad (2)$$

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 \( t_i = (x_{i1}, x_{i2}, x_{i3}) \) and \( t_j = (x_{j1}, x_{j2}, x_{j3}) \), we define the corresponding degree-2 polynomial kernel as

\[
K_p(t_i, t_j) = \text{tr} \left( T_i T_j \right)
\]

where

\[
T_i = (x_{i1} - x_{i3}) (x_{i1} - x_{i3})^T - (x_{i1} - x_{i2}) (x_{i1} - x_{i2})^T,
\]

\[
T_j = (x_{j1} - x_{j3}) (x_{j1} - x_{j3})^T - (x_{j1} - x_{j2}) (x_{j1} - x_{j2})^T.
\]

The kernel function in (3) defines an inner product of two triplets. 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

\[
g_d(z) = \text{sgn} \left( \sum_l h_l \alpha_l K_p(z_l, z) + b \right)
\]

\[
g_t(t) = \text{sgn} \left( \sum_l \alpha_l K_p(t_l, t) \right)
\]

where \( z_l, l = 1, 2, \cdots, N \), is the doublet constructed from the training dataset, \( h_l \) is the label of \( z_l \), \( t_l, l = 1, 2, \cdots, N \), is the triplet constructed from the training dataset, \( z = (x_{(i)}, x_{(j)}) \) is the test doublet, \( t \) is the test triplet, \( \alpha_l \) is the weight, and \( b \) is the bias.

For doublet, we have

\[
\sum_l h_l \alpha_l \text{tr} \left( \frac{(x_{i1} - x_{i2}) (x_{i1} - x_{i2})^T}{(x_{i1} - x_{i3}) (x_{i1} - x_{i3})^T} \right) + b = (x_{(i)} - x_{(j)})^T M (x_{(i)} - x_{(j)}) + b
\]

where

\[
M = \sum_l h_l \alpha_l (x_{i1} - x_{i2}) (x_{i1} - x_{i2})^T
\]

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

For triplet, the matrix \( M \) can be derived as follows.

\textbf{Theorem 1:} For the decision function defined in (5), the matrix \( M \) of the Mahalanobis distance metric is

\[
M = \sum_l \alpha_l T_l = \sum_l \alpha_l \left[ (x_{i1} - x_{i3}) (x_{i1} - x_{i3})^T - (x_{i1} - x_{i2}) (x_{i1} - x_{i2})^T \right]
\]

and then \( \sum_l \alpha_l K_p(t_l, t) \) denotes the relative difference of the Mahalanobis distance between \( x_{(i)} \) and \( x_{(k)} \) and the Mahalanobis distance between \( x_{(i)} \) and \( x_{(j)} \).

\textbf{Proof:} Let \( T_i = (x_{i1} - x_{i3}) (x_{i1} - x_{i3})^T - (x_{i1} - x_{i2}) (x_{i1} - x_{i2})^T \). Based on the definition of \( K_p(t_l, t) \), we have

\[
\sum_l \alpha_l K_p(t_l, t) = \sum_l \alpha_l \text{tr} \left( T_l T \right)
\]

\[
= \sum_l \alpha_l \text{tr} \left( T_l \left( \frac{(x_{(i)} - x_{(j)}) (x_{(i)} - x_{(j)})^T}{(x_{(i)} - x_{(j)}) (x_{(i)} - x_{(j)})^T} \right) \right)
\]

\[
= \sum_l \alpha_l \text{tr} \left( T_l \left( \frac{(x_{(i)} - x_{(j)}) (x_{(i)} - x_{(j)})^T}{(x_{(i)} - x_{(j)}) (x_{(i)} - x_{(j)})^T} \right) \right)
\]

\[
= (x_{(i)} - x_{(j)})^T \left( \sum_l \alpha_l T_l \right) (x_{(i)} - x_{(j)})
\]

\[
= (x_{(i)} - x_{(k)})^T \sum_l \alpha_l T_l (x_{(i)} - x_{(j)})
\]

\[
= (x_{(i)} - x_{(k)})^T M (x_{(i)} - x_{(j)})
\]

\[
= (x_{(i)} - x_{(k)})^T M (x_{(i)} - x_{(j)}),
\]

By setting \( M = \sum_l \alpha_l T_l \) as the matrix \( M \) in the Mahalanobis distance metric, we can see that \( \sum_l \alpha_l K_p(t_l, t) \) is the difference of the distance between \( x_{(i)} \) and \( x_{(k)} \) and that between \( x_{(i)} \) and \( x_{(j)} \).

Clearly, equations (4) \( \sim \) (9) provide us a new perspective to view and understand the distance metric matrix \( M \) 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.

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

\[
\min_{M, \xi} r(M) + \rho(\xi)
\]

s.t. \( f_1^{(d)} \left( (x_{i1} - x_{i2})^T M (x_{i1} - x_{i2}), b, \xi_1 \right) \geq 0 \) (doublet set)

or \( f_1^{(t)} \left( (x_{i1} - x_{i3})^T M (x_{i1} - x_{i3}), \xi_2 \right) \geq 0 \) (triplet set)

\[
\xi_i \geq 0
\]

where \( r(M) \) is the regularizer term, \( \rho(\xi) \) is the margin loss term, the constraint \( f_1^{(d)} \) can be any linear function of \( (x_{i1} - x_{i2})^T M (x_{i1} - x_{i2}) \), \( b \), and \( \xi_1 \), and the constraint \( f_1^{(t)} \)
can be any linear function of \((x_{i,1} - x_{i,3})^T M(x_{i,1} - x_{i,3}) - (x_{i,1} - x_{i,2})^T M(x_{i,1} - x_{i,2})\) and \(\xi_i\). To guarantee that (10) is convex, we can simply choose convex regularizer \(r(M)\) and convex margin loss \(\rho(\xi)\). By plugging (7) or (8) in the model in (10), we can employ the SVM and kernel methods to learn all \(\alpha_i\) to obtain the matrix \(M\).

If we adopt the \(l_2\)-norm to regularize \(M\) and the hinge loss penalty on \(\xi_i\), the model in (10) would become the standard SVM. SVM and its variants have been extensively studied [30], [44], [45], and various algorithms have been proposed for large-scale SVM training [46], [47]. Thus, the SVM-like modeling 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) [48], we let \(h_i = 1\) if the samples of doublet \(z_i\) belong to the same class and let \(h_i = 0\) if the samples of it belong to different classes. Meanwhile, suppose that the label of a doublet \(z_i\) is unknown, and we can calculate the probability that \(z_i\)'s label is 1 as follows:

\[
P(p_i = 1 | z_i) = \frac{1}{1 + \exp \left( \sum_i \alpha_i K_{p}(z_i, z_i) + b \right)}.
\]

(11)

The coefficient \(\alpha\) and the bias \(b\) can be obtained by maximizing the following log-likelihood function:

\[
(\alpha, b) = \arg \max_{\alpha, b} \left\{ \sum_{i} h_i \ln P(p_i = 1 | z_i) + (1 - h_i) \ln P(p_i = 0 | z_i) \right\}.
\]

(12)

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 [49], 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 \(\{(x_i, x_j, x_k)\}\), where \(x_i\) has the same label as \(x_j\) but has different label from \(x_k\). The minimization of LMNN can be stated as follows:

\[
\min_{M, \xi_{ijk}} \sum_{i,j,k} (x_i - x_j)^T M(x_i - x_j) + C \sum_{i,j,k} \xi_{ijk}
\]

\[
\text{s.t. } (x_i - x_k)^T M(x_i - x_k) - (x_i - x_j)^T M(x_i - x_j) \geq 1 - \xi_{ijk}. \tag{13}
\]

Since \(M\) is required to be positive semi-definite in LMNN, we introduce the following indicator function:

\[
t_{\rho}(M) = \begin{cases} 0, & \text{if } M \succ 0 \\ +\infty, & \text{otherwise} \end{cases}
\]

(14)

and choose the following regularizer and margin loss:

\[
r_{\text{LMNN}}(M) = \sum_{i,j} (x_i - x_j)^T M(x_i - x_j) + t_{\rho}(M) \tag{15}
\]

\[
\rho_{\text{LMNN}}(\xi) = C \sum_{i,j,k} \xi_{ijk}. \tag{16}
\]

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

\[
\min_{M, \xi} r_{\text{LMNN}}(M) + \rho_{\text{LMNN}}(\xi)
\]

\[
\text{s.t. } (x_i - x_k)^T M(x_i - x_k) - (x_i - x_j)^T M(x_i - x_j) \geq 1 - \xi_{ijk}.
\]

(17)

It is obvious that the SVM-like model in (17) is equivalent to the LMNN model in (13).

2) **ITML:** ITML [10] is operated on a set of doublets \(\{(x_i, x_j)\}\) by solving the following minimization problem:

\[
\min_{M, \xi} D_{\text{ld}}(M, M_0) + \gamma \cdot D_{\text{ld}}(\text{diag}(\xi), \text{diag}(\xi_0))
\]

\[
\text{s.t. } (x_i - x_j)^T M(x_i - x_j) \leq \xi_{u(i,j)} \ (i,j) \in S
\]

\[
(x_i - x_j)^T M(x_i - x_j) \geq \xi_{l(i,j)} \ (i,j) \in D \tag{18}
\]

(22)

\[
M \succ 0
\]

where \(M_0\) is the given prior of the metric matrix, \(\xi_0\) is the given prior on \(\xi\), \(S\) is the set of doublets where \(x_i\) and \(x_j\) have the same label, \(D\) is the set of doublets where \(x_i\) and \(x_j\) have different labels, and \(D_{\text{ld}}(\cdot, \cdot)\) is the LogDet divergence of two matrices defined as:

\[
D_{\text{ld}}(M, M_0) = \text{tr} \left( MM_0^{-1} \right) - \log \det \left( MM_0^{-1} \right) - n. \tag{19}
\]

By introducing the following regularizer and margin loss:

\[
r_{\text{ITML}}(M) = D_{\text{ld}}(M, M_0) + t_{\rho}(M) \tag{20}
\]

\[
\rho_{\text{ITML}}(\xi) = \gamma \cdot D_{\text{ld}}(\text{diag}(\xi), \text{diag}(\xi_0)) \tag{21}
\]

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

\[
\min_{M, \xi} r_{\text{ITML}}(M) + \rho_{\text{ITML}}(\xi)
\]

\[
\text{s.t. } (x_i - x_j)^T M(x_i - x_j) \leq \xi_{u(i,j)} \ (i,j) \in S
\]

\[
(x_i - x_j)^T M(x_i - x_j) \geq \xi_{l(i,j)} \ (i,j) \in D \tag{22}
\]

\[
\xi_{ij} \geq 0
\]

where \(z_{ij} = (x_i, x_j)\). One can easily see that the SVM-like model in (22) is equivalent to the ITML model in (18).
3) LDML: LDML [3] is a logistic discriminant based metric learning approach based on a set of doublets. Given a doublet \( z_l = (x_{i(l)}, x_{i(j)}) \) and its label \( y_{i(l)} \), LDML defines the probability that \( y_{i(l)} = y_{i(j)} \) as follows:

\[
p_l = P(y_{i(l)} = y_{i(j)} | x_{i(l)}, x_{i(j)}, M, b) = \sigma(b - d_M(x_{i(l)}, x_{i(j)}))
\]

where \( \sigma(z) \) is the sigmoid function, \( b \) is the bias, and \( d_M(x_{i(l)}, x_{i(j)}) = (x_{i(l)} - x_{i(j)})^T M (x_{i(l)} - x_{i(j)}) \). With the \( p_l \) defined in (23), LDML learns \( M \) and \( b \) by maximizing the following log-likelihood:

\[
\max_{M, b} \left\{ \sum_l h_l \ln p_l + (1 - h_l) \ln(1 - p_l) \right\} .
\]  

(24)

Note that \( M \) is not constrained to be positive-definite in LDML.

With the same doublet set, let \( \alpha \) be the solution obtained by the kernel logistic model in (12), and \( M \) be the solution of LDML in (24). It is easy to see that:

\[
M = \sum_l \alpha_l (x_{i(l)} - x_{i(j)}) (x_{i(l)} - x_{i(j)})^T.
\]

(25)

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 is able to suggest new metric learning models. Actually, for both ITML and LMNN, the positive semi-definite constraint is imposed on \( M \) 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 \( M \), and then map \( M \) onto the space of positive semi-definite matrices. 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 \( l_2 \)-norm to regularize \( M \) 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 \( l_2 \)-norm regularizer as \( r_{\text{SV}}(M) = \frac{1}{2} \| M \|^2_F \), and set \( \rho_{\text{SV}}(\xi_l) = C \sum_l \xi_l \) as the margin loss term, resulting in the following model:

\[
\min_{M, b, \xi} \frac{1}{2} \| M \|^2_F + C \sum_l \xi_l
\]

s.t. \( h_l \left( (x_{i,1} - x_{i,2})^T M (x_{i,1} - x_{i,2}) + b \right) \geq 1 - \xi_l \)

(26)

\( \xi_l \geq 0, \forall l \)

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

\[
\max_{\alpha} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j h_i h_j K_p(z_i, z_j) + \sum_i \alpha_i
\]

s.t. \( 0 \leq \alpha_i \leq C \)

(27)

\[ \sum_l \alpha_l h_l = 0, \forall l \]

which can be easily solved by many existing SVM solvers such as LibSVM [31]. The detailed deduction of the dual of doublet-SVM can be found in Appendix A.

B. Triplet-SVM

In triplet-SVM, we also choose \( r_{\text{SV}}(M) = \frac{1}{2} \| M \|^2_F \) as the regularization term, and choose \( \rho_{\text{SV}}(\xi) = C \sum_l \xi_l \) 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,

\[
\min_{M, \xi} \frac{1}{2} \| M \|^2_F + C \sum_l \xi_l
\]

s.t. \( (x_{i,1} - x_{i,3})^T M (x_{i,1} - x_{i,3}) \geq 1 - \xi_l \)

\( - (x_{i,1} - x_{i,2})^T M (x_{i,1} - x_{i,2}) \geq 1 - \xi_l \)

(28)

\( \xi_l \geq 0, \forall l \)

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 [45]. The dual problem of triplet-SVM is:

\[
\max_{\alpha} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K_p(t_i, t_j) + \sum_i \alpha_i
\]

s.t. \( 0 \leq \alpha_i \leq C, \forall l \)

(29)

which can also be efficiently solved by existing SVM solvers [31]. The detailed deduction of the dual of triplet-SVM can be found in Appendix B.

C. Discussions

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

The proposed doublet-SVM and triplet-SVM are easy to implement since the use of \( l_2 \)-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 [50], have been proposed for SVM training, making doublet-SVM and triplet-SVM very efficient to optimize. Moreover, using the large-scale SVM training algorithms [46], [47], [51], [53], 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 [43]. With the proposed framework, we can easily couple them with the degree-2 polynomial kernel to develop new metric learning approaches for various applications. Similarly, the kernel methods for semi-supervised learning [54], multiple instance learning [55], [56] and multitask learning [57] can also be adopted for metric learning with the proposed framework.

V. EXPERIMENTAL RESULTS

In the experiments, we evaluate the proposed doublet-SVM and triplet-SVM for k-NN classification 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. The source codes of LMNN, ITML, LDML, NCA, and MCML are also online available, and we tuned their parameters to get the best results.

A. UCI Dataset Classification

Ten datasets selected from the UCI machine learning repository [58] 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 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 these 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 the lowest error rates when \( m_2 = 2 \). Moreover, the error rates tend to be a little higher when \( m_2 > 3 \). Thus, we set \( m_2 = 1 \sim 3 \) in our experiments.

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.

![Fig. 1. Classification error rate (%) versus \( m_2 \) for (a) doublet-SVM and (b) triplet-SVM with \( m_1 = 1 \) and \( C = 1 \).](http://www.csie.ntu.edu.tw/~cjlin/lbosvm/)

![Fig. 2. Classification error rate (%) versus \( m_1 (= m_2) \) for (a) doublet-SVM and (b) triplet-SVM with \( C = 1 \).](http://www.csie.ntu.edu.tw/~cjlin/lbosvm/)

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 \( 10^3 \) for triplet-SVM. Thus, we set \( C < 10^4 \) for doublet-SVM and \( C < 10^3 \) for triplet-SVM in our experiments.

![Fig. 3. Classification error rate (%) versus \( C \) for (a) doublet-SVM and (b) triplet-SVM with \( m_1 = m_2 = 2 \).](http://www.csie.ntu.edu.tw/~cjlin/lbosvm/)

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 performances 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
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 | 2079                  | 231               | 19                | 7            |
| Breast Tissue        | 96                    | 10                | 9                 | 6            |
| ILPD                 | 525                   | 58                | 10                | 2            |
| Statlog Satellite    | 4435                  | 2000              | 36                | 6            |
| Blood Transfusion    | 674                   | 74                | 4                 | 2            |
| SPECTF Heart         | 80                    | 187               | 44                | 2            |
| Cardiotocography     | 1914                  | 212               | 21                | 10           |
| Letter               | 16000                 | 4000              | 16                | 26           |

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 |

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 [59].

From Table II, we can see that doublet-SVM achieves the best average rank and triplet-SVM achieves the fourth best 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 the large 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 2000 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 performances 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 Semeion datasets, we use 10-fold cross validation to evaluate the metric learning methods, and the error rate and training
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          |

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 2000 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.

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. As examples, two metric learning methods, i.e., doublet-SVM and triplet-SVM, were developed and they can be efficiently solved by the standard SVM solvers. Our experimental results on the UCI
TABLE IV

| 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 |
| Sonneon | 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 |

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.

APPENDIX A

THE DUAL OF DOUBLET-SVM

According to the original problem of doublet-SVM in (26), its Lagrangian can be defined as follows:

\[ L(M, b, \xi, \alpha, \beta) = \frac{1}{2} \| M \|_F^2 + C \sum_{l} \xi_l \]

\[- \sum_{l} \alpha_l \left[ h_l \left( (x_{l,1} - x_{l,2})^T M(x_{l,1} - x_{l,2}) + b \right) - 1 + \xi_l \right] \]

\[- \sum_{l} \beta_l \xi_l \]

(30)

where \( \alpha \) and \( \beta \) 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, let the derivative of the Lagrangian with respect to \( M, b \) and \( \xi \) be 0:

\[ \frac{\partial L(M, b, \xi, \alpha, \beta)}{\partial M} = 0 \Rightarrow M - \sum_{l} \alpha_l h_l (x_{l,1} - x_{l,2}) (x_{l,1} - x_{l,2})^T = 0 \]

(31)

\[ \frac{\partial L(M, b, \xi, \alpha, \beta)}{\partial b} = 0 \Rightarrow \sum_{l} \alpha_l h_l = 0 \]

(32)

\[ \frac{\partial L(M, b, \xi, \alpha, \beta)}{\partial \xi_l} = 0 \Rightarrow C - \alpha_l - \beta_l = 0 \Rightarrow 0 < \alpha_l < C, \forall l \]

(33)

Equation (31) implies the relationship between \( M \) and \( \alpha \) as follows:

\[ M = \sum_{l} \alpha_l h_l (x_{l,1} - x_{l,2}) (x_{l,1} - x_{l,2})^T \]

(34)

Substituting (31)~(33) back into the Lagrangian, we get the Lagrange dual problem of doublet-SVM as follows:

\[ \max_{\alpha} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j h_i h_j K_p(z_i, z_j) + \sum_{i} \alpha_i \]

s.t. \( 0 \leq \alpha_l \leq C \)

\[ \sum_{l} \alpha_l h_l = 0, \forall l \]

(35)

APPENDIX B

THE DUAL OF TRIPLET-SVM

According to the original problem of triplet-SVM in (28), its Lagrangian can be defined as follows:

\[ L(M, \xi, \alpha, \beta) = \frac{1}{2} \| M \|_F^2 + C \sum_{l} \xi_l \]

\[- \sum_{l} \alpha_l \left[ (x_{l,1} - x_{l,3})^T M(x_{l,1} - x_{l,3}) - (x_{l,1} - x_{l,2})^T M(x_{l,1} - x_{l,2}) \right] \]

\[ + \sum_{l} \alpha_l - \sum_{l} \alpha_l \xi_l - \sum_{l} \beta_l \xi_l \]

(36)

where \( \alpha \) and \( \beta \) 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 \( M \) and \( \xi \) to be 0:

\[ \frac{\partial L(M, \xi, \alpha, \beta)}{\partial M} = 0 \Rightarrow M - \sum_{l} \alpha_l \left[ (x_{l,1} - x_{l,3}) (x_{l,1} - x_{l,3})^T - (x_{l,1} - x_{l,2}) (x_{l,1} - x_{l,2})^T \right] = 0 \]

(37)

\[ \frac{\partial L(M, \xi, \alpha, \beta)}{\partial \xi_l} = 0 \Rightarrow C - \alpha_l - \beta_l = 0 \Rightarrow 0 < \alpha_l < C, \forall l \]

(38)

Equation (37) implies the relationship between \( M \) and \( \alpha \) as follows:

\[ M = \sum_{l} \alpha_l \left[ (x_{l,1} - x_{l,3}) (x_{l,1} - x_{l,3})^T - (x_{l,1} - x_{l,2}) (x_{l,1} - x_{l,2})^T \right] \]

(39)

Substituting (37) and (38) back into the Lagrangian, we get the Lagrange dual problem of triplet-SVM as follows:

\[ \max_{\alpha} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K_p(t_i, t_j) + \sum_{i} \alpha_i \]

s.t. \( 0 \leq \alpha_l \leq C, \forall l \]

(40)

ACKNOWLEDGMENT

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, and the Program of Ministry of Education for New Century Excellent Talents.
[54] 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, Dec. 2006.

[55] S. Andrews, I. Tsouchantaridis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in Proc. Adv. Neural Inf. Process. Syst., 2002, pp. 561-568.

[56] 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, Sep. 2013.

[57] 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.

[58] A. Frank and A. Asuncion. (2010) UCI machine learning repository [Online]. Available: [http://archive.ics.uci.edu/ml]

[59] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” J. Mach. Learn. Res., vol. 7, pp. 1-30, Dec. 2006.