Journal of Machine Learning Research 13 (2012) 1-26 Submitted 2/11; Revised 10/11; Published 1/12

Distance Metric Learning with Eigenvalue Optimization

Yiming Ying Y.YING@EXETER.AC.UK
College of Engineering, Mathematics and Physical Sciences
University of Exeter
Harrison Building, North Park Road
Exeter, EX4 4QF, UK

Peng Li LIPENG@IEEE.ORG
Department of Engineering Mathematics
University of Bristol
Merchant Venturers Building, Woodland Road
Bristol, BS8 1UB, UK

Editor: Sören Sonnenburg, Francis Bach and Cheng Soon Ong

Abstract

The main theme of this paper is to develop a novel eigenvalue optimization framework for learning a Mahalanobis metric. Within this context, we introduce a novel metric learning approach called *DML-eig* which is shown to be equivalent to a well-known eigenvalue optimization problem called minimizing the maximal eigenvalue of a symmetric matrix (Overton, 1988; Lewis and Overton, 1996). Moreover, we formulate *LMNN* (Weinberger et al., 2005), one of the state-of-the-art metric learning methods, as a similar eigenvalue optimization problem. This novel framework not only provides new insights into metric learning but also opens new avenues to the design of efficient metric learning algorithms. Indeed, first-order algorithms are developed for DML-eig and LMNN which only need the computation of the largest eigenvector of a matrix per iteration. Their convergence characteristics are rigorously established. Various experiments on benchmark data sets show the competitive performance of our new approaches. In addition, we report an encouraging result on a difficult and challenging face verification data set called Labeled Faces in the Wild (LFW).

Keywords: metric learning, convex optimization, semi-definite programming, first-order methods, eigenvalue optimization, matrix factorization, face verification

1. Introduction

Distance metrics are fundamental concepts in machine learning since a proper choice of a metric has crucial effects on the performance of both supervised and unsupervised learning algorithms. For example, the k-nearest neighbor (k-NN) classifier depends on a distance function to identify the nearest neighbors for classification. The k-means algorithm depends on the pairwise distance measurements between examples for clustering, and most information retrieval methods rely on a distance metric to identify the data points that are most similar to a given query. Recently, learning a distance metric from data has been actively studied in machine learning (Bar-Hillel et al., 2005; Davis et al., 2007; Goldberger et al., 2004; Rosales and Fung, 2006; Shen et al., 2009; Torresani and Lee, 2007; Weinberger et al.,

2005; Weinberger and Saul, 2008; Xing et al., 2002; Ying et al., 2009). These methods have been successfully applied to many real-world application domains including information retrieval, face verification, image recognition (Chopra et al., 2005; Guillaumin et al., 2009; Hoi et al., 2006) and bioinformatics (Kato and Nagano, 2010; Vert et al., 2007).

Most metric learning methods attempt to learn a distance metric from side information which is often available in the form of pairwise constraints, that is, pairs of *similar* data points and pairs of *dissimilar* data points. The information of similarity or dissimilarity between a pair of examples can easily be collected from the label information in supervised classification. For example, we can reasonably let two samples in the same class be a similar pair and samples in the distinct classes be a dissimilar pair. In semi-supervised clustering, a small amount of knowledge is available concerning pairwise (must-link or cannot-link) constraints between data items. This side information delivers the message that a must-link pair of samples is a similar pair and a cannot-link one is a dissimilar pair. A common theme in metric learning is to learn a distance metric such that the distance between similar examples should be relatively smaller than that between dissimilar examples. Although the distance metric can be a general function, the most prevalent one is the Mahalanobis metric defined by $d_M(x_i, x_j) = \sqrt{(x_i - x_j)^\top M(x_i - x_j)}$ where M is a positive semi-definite (p.s.d.) matrix.

In this work we restrict our attention to learning a Mahalanobis metric for k-nearest neighbor (k-NN) classification. However, the proposed methods below can easily be adapted to metric learning for semi-supervised k-means clustering. Our main contribution is summarized as follows. Firstly, we introduce a novel approach called *DML-eig* mainly inspired by the original work of Xing et al. (2002). Although our ultimate target is similar to theirs, our methods are essentially different. In particular, we can show our approach is equivalent to a well-known eigenvalue optimization problem called *minimizing the maximal eigenvalue of a symmetric matrix* (Lewis and Overton, 1996; Overton, 1988). We further show that the above novel optimization formulation can also be extended to LMNN (Weinberger et al., 2005) and low-rank matrix factorization for collaborative filtering (Srebro et al., 2004). Secondly, in contrast to the full eigen-decomposition used in many existing approaches to metric learning, we will develop novel approximate semi-definite programming (SDP) algorithms for DML-eig and LMNN which only need the computation of the largest eigenvector of a matrix per iteration. The algorithms combine and develop the Frank-Wolfe algorithm (Frank and Wolfe, 1956; Hazan, 2008) and Nesterov's smoothing techniques (Nesterov, 2005). Finally, its rigorous convergence characteristics will also be established, and experiments on various UCI data sets and benchmark face data sets show the competitiveness of our new approaches. In addition, we report an encouraging result on a challenging face verification data set called Labeled Faces in the Wild (Huang et al., 2007).

The paper is organized as follows. In Section 2, we propose our new approach (DML-eig) for distance metric learning and show its equivalence to the well-known eigenvalue optimization problem. In addition, a generalized eigenvalue-optimization formulation will be established for LMNN and low-rank matrix factorization for collaborative filtering (Srebro et al., 2004). In Section 3, based on eigenvalue optimization formulations of DML-eig and LMNN, we develop novel first-order algorithms. Their convergence rates are successfully established. Section 4 discusses the related work. In Section 5, our proposed methods

are compared with the state-of-the-art methods through extensive experiments. The last section concludes the paper.

2. Metric Learning Model and Equivalent Formulation

We begin by introducing useful notations. Let $\mathbb{N}_n = \{1, 2, \ldots, n\}$ for any $n \in \mathbb{N}$. The space of symmetric d times d matrices will be denoted by $\mathbb{S}^d$ and the cone of p.s.d. matrices is denoted by $\mathbb{S}^d_+$. For any $X, Y \in \mathbb{R}^{d\times n}$, we denote the inner product in $\mathbb{S}^d$ by $\langle X, Y\rangle := \mathbf{Tr}(X^\top Y)$ where $\mathbf{Tr}(\cdot)$ denotes the trace of a matrix. The standard norm in Euclidean space is denoted by $\|\cdot\|$.

Throughout the paper, the training data is given by $\mathbf{z} := \{(x_i, y_i) : i \in \mathbb{N}_n\}$ with input $x_i = (x_i^1, x_i^2, \ldots, x_i^d) \in \mathbb{R}^d$, class label y_i (not necessary binary) and later on we use the convention $X_{ij} = (x_i - x_j)(x_i - x_j)^\top$. Then, for any $M \in \mathbb{S}^d_+$, the associated Mahalanobis distance between x_i and x_j can be written as $d^2_M(x_i, x_j) = (x_i - x_j)^\top M(x_i - x_j) = \langle X_{ij}, M\rangle$. Let $\mathcal{S}$ index the similar pairs and $\mathcal{D}$ index the dissimilar pairs. For instance, if (x_i, x_j) is a similar pair we denote it by $\tau = (i, j) \in \mathcal{S}$, and write X_{ij} as X_τ for simplicity.

Given a set of pairwise distance constraints, the target of metric learning is to find a distance matrix M such that the distance between the dissimilar pairs is large and the distance between the similar pairs is small. There are many possible criteria to realize this intuition. Our model is mainly inspired by Xing et al. (2002) where the authors proposed to maximize the sum of distances between dissimilar pairs, while maintaining an upper bound on the sum of squared distances between similar pairs. Specifically, the following criterion was used in Xing et al. (2002):

$$\begin{array}{ll} \max_{M\in\mathbb{S}^d_+} & \sum_{(i,j)\in\mathcal{D}} d_M(x_i, x_j) \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}} d^2_M(x_i, x_j) \le 1. \end{array} \tag{1}$$

An iterative projection method was proposed to solve the above problem. However, it usually takes a long time to converge and the algorithm needs the computation of the full eigen-decomposition of a matrix in each iteration.

In this paper, we propose to maximize the minimal squared distances between dissimilar pairs while maintaining an upper bound for the sum of squared distances between similar pairs, that is,

$$\begin{array}{ll} \max_{M\in\mathbb{S}^d_+} & \min_{(i,j)\in\mathcal{D}} d^2_M(x_i, x_j) \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}} d^2_M(x_i, x_j) \le 1. \end{array} \tag{2}$$

Now, let $X_\mathcal{S} = \sum_{(i,j)\in\mathcal{S}} X_{ij}$ we can rewrite problem (2) as follows:

$$\begin{array}{ll} \max_{M\in\mathbb{S}^d_+} & \min_{\tau\in\mathcal{D}} \langle X_\tau, M\rangle \\ \text{s.t.} & \langle X_\mathcal{S}, M\rangle \le 1. \end{array} \tag{3}$$

This problem is obviously a semi-definite programming (SDP) since it is equivalent to

$$\begin{array}{ll} \max_{M\in\mathbb{S}^d_+} & t \\ \text{s.t.} & \langle X_\tau, M\rangle \ge t,\ \forall \tau = (i, j) \in \mathcal{D}, \\ & \langle X_\mathcal{S}, M\rangle \le 1. \end{array}$$

In contrast to problem (1), the objective function and the constraints in (3) are linear with respect to (w.r.t.) M. As shown in the next subsection, this simple but important property[1] plays a critical role in formulating problem (2) as an eigenvalue optimization problem. This equivalent formulation is key to the design of efficient algorithms in Section 3.

The generation of the pairwise constraints plays an important role in learning a metric. If labels are known then the learning setting is often referred to as supervised metric learning which can further be divided into two categories: the *global metric learning* and the *local metric learning*. The global approach learns the distance metric in a global sense, that is, to satisfy all the pairwise constraints simultaneously. The original model in Xing et al. (2002) is a global method which used all the similar pairs (same labels) and dissimilar pairs (distinct labels). The local approach is to learn a distance metric only using local pairwise constraints which usually outperforms the global methods as observed in many previous studies. This is reasonable in the case of learning a metric for the k-NN classifiers since k-NN classifiers are influenced most by the data items that are close to the test/query examples. Since we are mainly concerned with learning a metric for k-NN classifier, the pairwise constraints for DML-eig are generated locally, that is, the similar/dissimilar pairs are k-nearest neighbors. The details can be found in the experimental section.

2.1 Equivalent Formulation as Eigenvalue Optimization

In this section we establish a min-max formulation of problem (3), which is finally shown to be equivalent to an eigenvalue optimization problem called *minimizing the maximal eigenvalue of symmetric matrices* (Lewis and Overton, 1996; Overton, 1988).

For simplicity of notation, for any $X \in \mathbb{S}^d$, we denote its maximum eigenvalue of $X \in \mathbb{S}^d$ by $\lambda_{\max}(X)$. Let D be the number of dissimilar pairs and the simplex is denoted by

$$\triangle = \{u \in \mathbb{R}^D : u_\tau \geq 0, \quad \sum_{\tau \in \mathcal{D}} u_\tau = 1\}.$$

We also denote the *spectrahedron* by

$$\mathcal{P} = \{M \in \mathbb{S}^d_+ : \mathbf{Tr}(M) = 1\}.$$

Now we can show problem (3) is indeed an eigenvalue optimization problem.

Theorem 1 *Assume that $X_{\mathcal{S}}$ is invertible and, for any $\tau \in \mathcal{D}$, let $\widetilde{X}_\tau = X_{\mathcal{S}}^{-1/2} X_\tau X_{\mathcal{S}}^{-1/2}$. Then, problem (3) is equivalent to the following problem*

$$\max_{S \in \mathcal{P}} \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \langle \widetilde{X}_\tau, S \rangle, \tag{4}$$

which can further be written as an eigenvalue optimization problem:

$$\min_{u \in \triangle} \max_{S \in \mathcal{P}} \langle \sum_{\tau \in \mathcal{D}} u_\tau \widetilde{X}_\tau, S \rangle = \min_{u \in \triangle} \lambda_{max} \Big(\sum_{\tau \in \mathcal{D}} u_\tau \widetilde{X}_\tau \Big). \tag{5}$$

1. One might consider replacing the objective $\sum_{(i,j) \in \mathcal{D}} d_M(x_i, x_j)$ in problem (1) by $\sum_{(i,j) \in \mathcal{D}} d^2_M(x_i, x_j)$. This would also lead to a simple linear constraint and linear objective function. However, as mentioned in Xing et al. (2002), it would result in M always being rank 1 (i.e., the data are always projected onto a line).

Proof Let M^* be an optimal solution of problem (3) and $\widetilde{M}^* = \frac{M^*}{\langle X_{\mathcal{S}}, M^* \rangle}$. Then, we have $\langle X_{\mathcal{S}}, \widetilde{M}^* \rangle = 1$ and

$$\min_{\tau \in \mathcal{D}} \langle X_\tau, \widetilde{M}^* \rangle = \min_{\tau \in \mathcal{D}} \langle X_\tau, M^* \rangle / \langle X_{\mathcal{S}}, M^* \rangle \geq \min_{\tau \in \mathcal{D}} \langle X_\tau, M^* \rangle,$$

since $\langle X_{\mathcal{S}}, M^* \rangle \leq 1$. This implies that $\widetilde{M}^*$ is also an optimal solution. Consequently, problem (3) is equivalent to, up to a scaling constant,

$$\arg \max_{M \in \mathbb{S}_+^d} \{ \min_{\tau \in \mathcal{D}} \langle X_\tau, M \rangle : \langle X_{\mathcal{S}}, M \rangle = 1 \}. \tag{6}$$

Noting that $\min_{\tau \in \mathcal{D}} \langle X_\tau, M \rangle = \min_{u \in \triangle} \sum_{\tau \in \mathcal{D}} u_\tau \langle X_\tau, M \rangle$, the desired equivalence between (4) and (3) follows by changing variable $S = X_{\mathcal{S}}^{1/2} M X_{\mathcal{S}}^{1/2}$ in formulation (6).

Also, note from Overton (1988) that $\max_{M \in \mathcal{P}} \langle X, M \rangle = \lambda_{\max}(X)$. By the min-max theorem, problem (4) can further be written by a well-known eigenvalue optimization problem:

$$\min_{u \in \triangle} \max_{M \in \mathcal{P}} \langle \sum_{\tau \in \mathcal{D}} u_\tau \widetilde{X}_\tau, M \rangle = \min_{u \in \triangle} \lambda_{\max} \Big(\sum_{\tau \in \mathcal{D}} u_\tau \widetilde{X}_\tau \Big).$$

This completes the proof of the theorem. ■

The problem of minimizing the maximal eigenvalue of a symmetric matrix is well-known which has important applications in engineering design, see Overton (1988); Lewis and Overton (1996). Hereafter, we refer to metric learning formulation (3) (equivalently (4) or (5)) as **DML-eig**.

We end this subsection with two remarks. Firstly, Theorem 1 assumes that $X_{\mathcal{S}}$ is invertible. In practice, this can be achieved by enforcing a small ridge to the diagonal of the matrix $X_{\mathcal{S}}$, that is, $X_{\mathcal{S}} \longleftarrow X_{\mathcal{S}} + \delta \, \mathbf{I}_d$ where $\mathbf{I}_d$ is the identity matrix and $\delta > 0$ is a very small ridge constant. Without loss of generality, we assume that $X_{\mathcal{S}}$ is positive definite throughout the paper. Secondly, when the dimension d of the input space is very large, the computation of $X_{\mathcal{S}}^{-1/2}$ could be time-consuming. Instead of directly inverting the matrix, one can use the Cholesky decomposition which is faster and numerically more stable. Indeed, the Cholesky decomposition tells us that $X_{\mathcal{S}} = LL^\top$ where L is a lower triangular matrix with strictly positive diagonal entries. Hence, in (6) we can let $S = L^\top M L$ (i.e., $M = (L^{-1})^\top S L^{-1}$) and Theorem 1 still holds true if we redefine, for any $\tau = (i, j) \in \mathcal{D}$, that $\widetilde{X}_\tau = (L^{-1}(x_i - x_j))(L^{-1}(x_i - x_j))^\top$. Therefore, it suffices to compute $\{L^{-1} x_i : i \in \mathbb{N}_n\}$ which can efficiently be obtained by solving linear system of equations (e.g., using the operation $L \setminus x_i$ in MATLAB).

2.2 Eigenvalue Optimization for LMNN

Weinberger et al. (2005) proposed the large margin nearest neighbor classification (LMNN) which is one of the state-of-the-art metric learning methods. In analogy to the above argument for DML-eig, we can also formulate LMNN as a generalized eigenvalue optimization problem.

Formulation (3) used the pairwise constraints in the form of similar/dissimilar pairs. In contrast, LMNN aims to learn a metric using the relative distance constraints which

are presented in the form of triplets. With a little abuse of notation, we denote a triplet by $\tau = (i, j, k)$ which means that x_i is similar to x_j and x_j is dissimilar to x_k. Then, denote the set of triplets by $\mathcal{T}$ which can be specified based on label information (e.g., see Section 5). Given a set $\mathcal{S}$ of similar pairs and a set $\mathcal{T}$ of triplets, the target of LMNN is to learn a distance metric such that k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. In particular, let $X_{\mathcal{S}} = \sum_{(i,j)\in\mathcal{S}}(x_i - x_j)(x_i - x_j)^\top$ and $C_\tau = X_{jk} - X_{ij}$, then LMNN can be rewritten as

$$\begin{array}{ll} \min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau + \gamma\mathbf{Tr}(X_{\mathcal{S}}M) \\ \text{s.t.} & 1 - \langle C_\tau, M\rangle \le \xi_\tau, \\ & M \in \mathbb{S}_+^d, \xi_\tau \ge 0,\ \forall \tau = (i,j,k) \in \mathcal{T}, \end{array} \tag{7}$$

where $\gamma \in [0, 1]$ is a trade-off parameter.

Let T be the number of triplets, that is, the cardinality of the triplet set $\mathcal{T}$. We can establish the following equivalent min-max formulation of LMNN. A similar result with a quite different proof has also been given in Baes and Bürgisser (2009) for a certain class of SDP problems.

Lemma 2 *LMNN formulation (7) is equivalent to*

$$\max_{M\in\mathbb{S}_+^d,\xi\ge 0}\big\{\min_\tau(\xi_\tau + \langle C_\tau, M\rangle) : \gamma\mathbf{Tr}(X_{\mathcal{S}}M) + (1-\gamma)\sum_\tau \xi_\tau = 1\big\}. \tag{8}$$

Proof Write the LMNN formulation (7) as

$$\begin{array}{ll} \min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau + \gamma\mathbf{Tr}(X_{\mathcal{S}}M) \\ \text{s.t.} & \langle C_\tau, M\rangle + \xi_\tau \ge 1, \\ & M \in \mathbb{S}_+^d, \xi_\tau \ge 0,\ \forall \tau = (i,j,k) \in \mathcal{T}. \end{array} \tag{9}$$

The condition that $\langle C_\tau, M\rangle + \xi_\tau \ge 1$ for any $\tau \in \mathcal{T}$ is identical to $\min_{\tau\in\mathcal{T}}\langle C_\tau, M\rangle + \xi_\tau \ge 1$. Hence, problem (7) is further equivalent to

$$\begin{array}{ll} \min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau + \gamma\mathbf{Tr}(X_{\mathcal{S}}M) \\ \text{s.t.} & \min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle + \xi_\tau \ge 1, \\ & M \in \mathbb{S}_+^d, \xi \ge 0. \end{array} \tag{10}$$

Since the objective function and the constraints are linear w.r.t. variable (M, ξ), the optimal solution for (10) must be attained on the boundary of the feasible domain, that is, $\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle + \xi_\tau = 1$. Consequently, problem (10) is identical to

$$\begin{array}{ll} \min_{M,\xi} & (1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau + \gamma\mathbf{Tr}(X_{\mathcal{S}}M) \\ \text{s.t.} & \min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle + \xi_\tau = 1, \\ & M \in \mathbb{S}_+^d, \xi \ge 0. \end{array} \tag{11}$$

Let $\Omega = \big\{(M, \xi) : \min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle + \xi_\tau \ge 0, M \in \mathbb{S}_+^d, \xi \ge 0\big\}$. We first claim that (11) is equivalent to

$$\min_{M,\xi}\left\{\frac{(1-\gamma)\sum_{\tau\in\mathcal{T}}\xi_\tau + \gamma\mathbf{Tr}(X_{\mathcal{S}}M)}{\min_{\tau=(i,j,k)\in\mathcal{T}}\langle C_\tau, M\rangle + \xi_\tau} : (M,\xi)\in\Omega\right\}. \tag{12}$$

To see this equivalence, let ϕ_1 be the optimal value of problem (11) and ϕ_2 be the optimal value of problem (12). Suppose that (M^*, ξ^*) be an optimal solution of problem (12). Let $\delta^* = \min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M^*\rangle + \xi^*_\tau$ and denote $(\widetilde{M}^*, \widetilde{\xi}^*) = (M^*/\delta^*, \xi^*/\delta^*)$. Then, for any $M \in \mathbb{S}^d_+$ and $\xi \geq 0$ satisfying $\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau = 1$,

$$(1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M) = \frac{(1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M)}{\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau} \geq \phi_2 = \frac{(1-\gamma)\sum_{\tau\in\mathcal{T}} \xi^*_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M^*)}{\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M^*\rangle + \xi^*_\tau}$$
$$= (1-\gamma)\sum_{\tau\in\mathcal{T}} \widetilde{\xi}^*_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} \widetilde{M}^*) \geq \phi_1,$$

where the last inequality follows from the fact that $\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, \widetilde{M}^*\rangle + \widetilde{\xi}^*_\tau = 1$. Since the above inequality holds true for any $M \in \mathbb{S}^d_+$ and $\xi \geq 0$ satisfying $\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau = 1$, we finally get that $\phi_1 \geq \phi_2 \geq \phi_1$, that is, $\phi_1 = \phi_2$ and, moreover $(\widetilde{M}^*, \widetilde{\xi}^*)$ is an optimal solution of problem (11). This completes the equivalence between (11) and (12).

Now, rewrite problem (12) as

$$\min_{M,\xi} \left\{ \Big(\frac{\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau}{(1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M)} \Big)^{-1} : (M,\xi) \in \Omega \right\},$$

which is further equivalent to

$$\max_{M,\xi} \left\{ \frac{\min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau}{(1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M)} : (M,\xi) \in \Omega \right\}. \tag{13}$$

Using exactly the same argument of proving the equivalence between (11) and (12), one can show that the above problem (13) is equivalent to

$$\max_{M,\xi} \Big\{ \min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau : (1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M) = 1, (M,\xi) \in \Omega \Big\}. \tag{14}$$

Now consider problem (14) without the restriction $(M,\xi) \in \Omega$, that is,

$$\max_{M,\xi} \Big\{ \min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau : (1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M) = 1, M \in \mathbb{S}^d_+,\ \xi \geq 0 \Big\}. \tag{15}$$

Let $\tilde{M} = \mathbf{0}$ and $\tilde{\xi}_\tau = \frac{1}{(1-\gamma)T}$ for any τ which obviously satisfies the restriction condition of problem (15), that is, $(1-\gamma)\sum_{\tau\in\mathcal{T}} \tilde{\xi}_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} \tilde{M}) = 1$. Then,

$$\max_{M,\xi} \Big\{ \min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, M\rangle + \xi_\tau : (1-\gamma)\sum_{\tau\in\mathcal{T}} \xi_\tau + \gamma \mathbf{Tr}(X_{\mathcal{S}} M) = 1, M \in \mathbb{S}^d_+,\ \xi \geq 0 \Big\}$$
$$\geq \min_{\tau=(i,j,k)\in\mathcal{T}} \langle C_\tau, \tilde{M}\rangle + \tilde{\xi}_\tau = \frac{1}{(1-\gamma)T} > 0,$$

which means that any optimal solution for problem (15) automatically satisfies $(M,\xi) \in \Omega$. Consequently, problem (15) is equivalent to (14). Combining this with the equivalence between (11), (12), (13) and (14) finally yields the equivalence between problem (15) and the primal formulation (7) of LMNN. This completes the proof of the lemma. ■

Using the above min-max representation for LMNN, it is now easy to reformulate LMNN as a generalized eigenvalue optimization as we will do below. With a little abuse of notation, denote the simplex by $\triangle = \{u \in \mathbb{R}^T : \sum_{\tau\in\mathcal{T}} u_\tau = 1,\ u_\tau \geq 0\}$.

Theorem 3 *Assume that $X_{\mathcal{S}}$ is invertible and, for any $\tau \in \mathcal{T}$, let $\widetilde{C}_\tau = X_{\mathcal{S}}^{-1/2} C_\tau X_{\mathcal{S}}^{-1/2}$. Then, LMNN is equivalent to the following problem*

$$\max_{S,\xi}\Big\{\min_{u\in\triangle}\sum_{\tau\in\mathcal{T}} u_\tau\big(\xi_\tau + \langle\widetilde{C}_\tau, S\rangle\big) : (1-\gamma)\xi^\top\mathbf{1} + \gamma\mathbf{Tr}(S) = 1, S\in\mathbb{S}_+^d,\ \xi\geq 0\Big\}, \tag{16}$$

where $\mathbf{1}$ is a column vector with all entries one. Moreover, it can further be written as a generalized eigenvalue optimization problem:

$$\min_{u\in\triangle}\max\Big(\frac{1}{1-\gamma}u_{max}, \frac{1}{\gamma}\lambda_{max}\big(\sum_{\tau\in\mathcal{T}} u_\tau\widetilde{C}_\tau\big)\Big), \tag{17}$$

where u_{max} is the maximum element of the vector $(u_\tau : \tau\in\mathcal{T})$.

Proof Note that $\min_{\tau=(i,j,k)\in\mathcal{T}}\big(\langle C_\tau, M\rangle + \xi_\tau\big) = \min_{u\in\triangle} u_\tau\big(\langle C_\tau, M\rangle + \xi_\tau\big)$. Combing this with Lemma 2 implies that LMNN is equivalent to

$$\max_{M,\xi}\Big\{\min_{u\in\triangle}\sum_{\tau\in\mathcal{D}} u_\tau\big(\xi_\tau + \langle C_\tau, M\rangle\big) : (1-\gamma)\xi^\top\mathbf{1} + \gamma\mathbf{Tr}(X_{\mathcal{S}}M) = 1, M\in\mathbb{S}_+^d,\ \xi\geq 0\Big\}.$$

Letting $S = X_{\mathcal{S}}^{1/2} M X_{\mathcal{S}}^{1/2}$ yields the equivalence between (8) and (16).

By the min-max theorem, problem (16) is equivalent to

$$\min_{u\in\triangle}\Big\{\max_{S,\xi}\sum_{\tau\in\mathcal{D}} u_\tau\big(\xi_\tau + \langle\widetilde{C}_\tau, S\rangle\big) : (1-\gamma)\xi^\top\mathbf{1} + \gamma\mathbf{Tr}(S) = 1, S\in\mathbb{S}_+^d,\ \xi\geq 0\Big\}. \tag{18}$$

To see the equivalence between (18) and (17), observe that

$$\begin{aligned}
&\max\Big\{\textstyle\sum_{\tau\in\mathcal{D}} u_\tau\big(\xi_\tau + \langle\widetilde{C}_\tau, S\rangle\big) : (1-\gamma)\xi^\top\mathbf{1} + \gamma\mathbf{Tr}(S) = 1, S\in\mathbb{S}_+^d,\ \xi\geq 0\Big\}\\
&= \max\Big\{\tfrac{1}{1-\gamma}\sum_{\tau\in\mathcal{D}} u_\tau\xi_\tau + \tfrac{1}{\gamma}\langle\sum_{\tau\in\mathcal{D}} u_\tau\widetilde{C}_\tau, S\rangle : \xi^\top\mathbf{1} + \mathbf{Tr}(S) = 1, S\in\mathbb{S}_+^d,\ \xi\geq 0\Big\}\\
&= \max\Big(\tfrac{1}{1-\gamma}u_{\max}, \tfrac{1}{\gamma}\lambda_{\max}\big(\sum_{\tau\in\mathcal{D}} u_\tau\widetilde{C}_\tau\big)\Big),
\end{aligned}$$

where the last equality follows from the fact that the above maximization problem is a linear programming w.r.t. (S,ξ) and, for any $A\in\mathbb{S}^d$, $\max\{\langle A, B\rangle : B\in\mathbb{S}_+^d,\ \mathbf{Tr}(B)\leq 1\} = \lambda_{\max}(A)$. This completes the proof of the theorem. ■

Since we have formulated LMNN as an eigenvalue optimization problem in the above theorem, hereafter we refer to formulation (16) (equivalently (17)) as **LMNN-eig**. The above eigenvalue optimization formulation is not restricted to metric learning problems. It can be extended to other machine learning tasks if their SDP formulation is similar to that of LMNN. Maximum-margin matrix factorization (Srebro et al., 2004) is one of such examples. Its eigenvalue optimization formulation can be found in Appendix A.

3. Eigenvalue Optimization Algorithms

In this section we develop efficient algorithms for solving DML-eig and LMNN-eig. We can directly employ the entropy smoothing techniques (Nesterov, 2007; Baes and Bürgisser, 2009) for eigenvalue optimization which, however, needs the computation of the full eigen-decomposition per iteration. Instead, we propose a new first-order method by developing and combining the smoothing techniques (Nesterov, 2005) and Frank-Wolfe algorithm (Frank and Wolfe, 1956; Hazan, 2008), which will only involve the computation of the largest eigenvector of a matrix.

3.1 Approximate Frank-Wolfe Algorithm for DML-eig

By Theorem 1, DML-eig is identical to problem:

$$\max_{S\in\mathcal{P}} f(S) = \max_{S\in\mathcal{P}} \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau \langle \widetilde{X}_\tau, S\rangle. \tag{19}$$

To this end, for a smoothing parameter $\mu > 0$, define

$$f_\mu(S) = \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau \langle \widetilde{X}_\tau, S\rangle + \mu \sum_{\tau\in\mathcal{D}} u_\tau \ln u_\tau.$$

We use the smoothed problem $\max_{S\in\mathcal{P}} f_\mu(S)$ to approximate problem (19).

It is easy to see that

$$f_\mu(S) = -\mu \ln\big(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\big),$$

and

$$\nabla f_\mu(S) = \frac{\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu} \widetilde{X}_\tau}{\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}}.$$

Since f_μ is a smooth function, we can prove that its gradient is Lipschitz continuous.

Lemma 4 *For any $S_1, S_2 \in \mathcal{P}$, then*

$$\|\nabla f_\mu(S_1) - \nabla f_\mu(S_2)\| \le C_\mu \|S_1 - S_2\|,$$

where $C_\mu = 2\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2/\mu$.

Proof It suffices to see $\|\nabla^2 f_\mu(S)\| \le 2\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2/\mu$. To this end,

$$\nabla^2 f_\mu(S) = \frac{(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu} \widetilde{X}_\tau) \bigotimes (\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu} \widetilde{X}_\tau)}{\mu\big(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\big)^2} - \frac{\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu} \widetilde{X}_\tau \bigotimes \widetilde{X}_\tau}{\mu \sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}} := I + II,$$

where $X \bigotimes S$ denotes the tensor product of matrices X and S. We can estimate the term I as follows:

$$\|I\| \le \frac{\big(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu} \|\widetilde{X}_\tau\|\big)\big(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu} \|\widetilde{X}_\tau\|\big)}{\mu\big(\sum_{\tau\in\mathcal{D}} e^{-\langle \widetilde{X}_\tau, S\rangle/\mu}\big)^2} \le \frac{1}{\mu} \max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2,$$

Input:
· smoothing parameter $\mu > 0$ (e.g., 10^{-5})
· tolerance value *tol* (e.g., 10^{-5})
· step sizes $\{\alpha_t \in (0,1) : t \in \mathbb{N}\}$
Initialization: $S_1^\mu \in \mathbb{S}_+^d$ with $\mathbf{Tr}(S_1^\mu) = 1$
for $t = 1, 2, 3, \ldots$ **do**
· $Z_t^\mu = \arg\max \{f_\mu(S_t) + \langle Z, \nabla f_\mu(S_t^\mu)\rangle : Z \in \mathbb{S}_+^d,\ \mathbf{Tr}(Z) = 1\}$, that is, $Z_t^\mu = vv^\top$ where v is the maximal eigenvector of matrix $\nabla f_\mu(S_t^\mu)$
· $S_{t+1}^\mu = (1-\alpha_t)S_t^\mu + \alpha_t Z_t^\mu$
· if $|f_\mu(S_{t+1}^\mu) - f_\mu(S_t^\mu)| < tol$ then **break**
Output: $d \times d$ matrix $S_t^\mu \in \mathbb{S}_+^d$

Table 1: Approximate Frank-Wolfe Algorithm for DML-eig

where, in the above inequality, we used the fact that $\|S \bigotimes X\| \leq \|X\|\|S\|$ for any $X, S \in \mathbb{S}^d$. The second term II can be similarly estimated:

$$\|II\| \leq \max_{\tau \in \mathcal{D}} \|\widetilde{X}_\tau\|^2/\mu.$$

Putting them together yields the desired result. ■

The pseudo-code to solve DML-eig is described in Table 1 which is a generalization of Frank-wolfe algorithm (Frank and Wolfe, 1956) which originally applies to the context of minimizing a convex function over a feasible polytope. Hazan (2008) first extended the original Frank-Wolfe algorithm to solve SDP over the spectrahedron $\mathcal{P} = \{M : M \in \mathbb{S}_+^d, \mathbf{Tr}(M) = 1\}$. Recall that D is the cardinality of $\mathcal{D}$, that is, the number of dissimilar pairs. Then, we have the following convergence result.

Lemma 5 *For any $0 < \mu \leq 1$, let $\{S_t^\mu : t \in \mathbb{N}\}$ be generated by the algorithm in Table 1 and C_μ be defined in Lemma 4. Then we have that*

$$\max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S_{t+1}^\mu) \leq C_\mu \alpha_t^2 + (1-\alpha_t)\big(\max_{S \in \mathcal{P}} f_\mu(S) - f(S_t^\mu)\big).$$

Proof By the definition of C_μ in Lemma 4, we have

$$f_\mu(S_{t+1}^\mu) \geq f_\mu(S_t^\mu) + \alpha_t \langle \nabla f_\mu(S_t^\mu), Z_t - S_t^\mu\rangle - C_\mu \alpha_t^2. \tag{20}$$

Since f is concave, for any $S \in \mathcal{P}$ there holds

$$\langle \nabla f_\mu(S_t^\mu), Z_t - S_t^\mu\rangle \geq \langle \nabla f_\mu(S_t^\mu), S - S_t^\mu\rangle \geq f_\mu(S) - f(S_t^\mu),$$

which implies that

$$\langle \nabla f_\mu(S_t^\mu), Z_t - S_t^\mu\rangle \geq \max_{S \in \mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu).$$

Substituting the above inequality into (20) yields the desired result. ■

For simplicity, let $R_t = \max_{S\in\mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu)$. If $\alpha_t \in (0,1]$ for any $t \geq t_0$ with some $t_0 \in \mathbb{N}$, then by Lemma 5 and a simple induction, for any $t \geq t_0$ there holds

$$R_{t+1} \leq C_\mu \sum_{j=t_0}^{t} \prod_{k=j+1}^{t} (1-\alpha_k)\alpha_j^2 + \prod_{j=t_0}^{t}(1-\alpha_j) R_{t_0}. \tag{21}$$

Combining this inequality and some ideas in Ying and Zhou (2006), one can establish sufficient conditions on the stepsizes $\{\alpha_t : t \in \mathbb{N}\}$ such that $\lim_{t\to\infty} f_\mu(S_t^\mu) = \min_{S\in\mathcal{P}} f_\mu(S)$.

Theorem 6 *For any fixed $\mu > 0$, let $\{S_t^\mu : t \in \mathbb{N}\}$ be generated by the algorithm in Table 1. If the step sizes satisfy that*

$$\sum_{t\in\mathbb{N}} \alpha_t = \infty, \qquad \lim_{t\to\infty} \alpha_t = 0, \tag{22}$$

then

$$\lim_{t\to\infty} f_\mu(S_t^\mu) = \max_{S\in\mathcal{P}} f_\mu(S).$$

The detailed proof of the above theorem is given in Appendix B. Typical examples of step sizes satisfying condition (22) are $\{\alpha_t = t^{-\theta} : t \in \mathbb{N}\}$ with $0 < \theta \leq 1$. For the particular case $\theta = 1$, by Lemma 5 we can prove the following result.

Theorem 7 *For any $0 < \mu \leq 1$, let $\{S_t^\mu : t \in \mathbb{N}\}$ be generated by Table 1 with step sizes given by $\{\alpha_t = 2/(t+1) : t \in \mathbb{N}\}$. Then, for any $t \in \mathbb{N}$ we have that*

$$\max_{S\in\mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu) \leq \frac{8\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2}{\mu t} + \frac{4\ln D}{t}. \tag{23}$$

Furthermore,

$$\max_{S\in\mathcal{P}} f(S) - f(S_t^\mu) \leq 2\mu \ln D + \frac{8\max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2}{\mu t} + \frac{8\ln D}{t}.$$

Proof It is easy to see, for any $S \in \mathcal{P}$ that

$$|f(S) - f_\mu(S)| \leq \mu \max_{u\in\triangle} \sum_{\tau\in\mathcal{D}} (-u_\tau \ln u_\tau) \leq \mu \ln D.$$

Let $S_* = \arg\max_{S\in\mathcal{P}} f(S)$ and $S_*^\mu = \arg\max_{S\in\mathcal{P}} f_\mu(S)$. Then, for any $t \in \mathbb{N}$,

$$\begin{aligned}
\max_{S\in\mathcal{P}} f(S) - f(S_t^\mu) &= [f(S_*) - f_\mu(S_*)] + [f_\mu(S_*) - \max_{S\in\mathcal{P}} f_\mu(S)] \\
&\quad + [f_\mu(S_*^\mu) - f_\mu(S_t^\mu)] + [f_\mu(S_t^\mu) - f(S_t^\mu)] \\
&\leq [f(S_*) - f_\mu(S_*)] + [f_\mu(S_*^\mu) - f_\mu(S_t^\mu)] + [f_\mu(S_t^\mu) - f(S_t^\mu)] \\
&\leq 2\mu \ln D + [f_\mu(S_*^\mu) - f_\mu(S_t^\mu)] \\
&= 2\mu \ln D + [\max_{S\in\mathcal{P}} f_\mu(S) - f_\mu(S_t^\mu)].
\end{aligned}$$

Hence, it suffices to prove (23) by induction. Indeed, for $t = 1$, we have that

$$\begin{aligned}\max_{S\in\mathcal{P}} f_\mu(S) - f_\mu(S_1^\mu) &\le f_\mu(S_*^\mu) + \mu \sup_{u\in\triangle}\big(\sum_{\tau\in\mathcal{D}}(-u_\tau \ln u_\tau)\big)\\ &\le \max_{S\in\mathcal{P}} \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau \langle \widetilde{X}_\tau, S\rangle + \mu \ln D\\ &\le \max_{S\in\mathcal{P}} \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau \|\widetilde{X}_\tau\|\|S\| + \mu \ln D\\ &\le \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau \|\widetilde{X}_\tau\| + \mu \ln D\\ &\le \min_{u\in\triangle} \big[\sum_{\tau\in\mathcal{D}} u_\tau + \sum_{\tau\in\mathcal{D}} u_\tau \|\widetilde{X}_\tau\|^2\big] + \mu \ln D\\ &\le 1 + \max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2 + \mu \ln D,\end{aligned}$$

which obviously satisfies (23) with $t = 1$. Suppose the inequality (23) holds true for some $t > 1$. Now by Lemma 5,

$$\begin{aligned}R_{t+1} &\le C_\mu \alpha_t^2 + (1-\alpha_t) R_t\\ &\le \tfrac{4C_\mu}{(t+1)^2} + \tfrac{t-1}{t+1}\big(\tfrac{4C_\mu}{t} + \tfrac{4\ln D}{t}\big)\\ &\le 4(C_\mu + \ln D)\big(\tfrac{1}{(t+1)^2} + \tfrac{t-1}{(t+1)t}\big) \le \tfrac{4(C_\mu + \ln D)}{t+1},\end{aligned}$$

where the second inequality follows from the induction assumption. This proves the inequality (23) for all $t \in \mathbb{N}$ which completes the proof of the theorem. ■

By the above theorem, for any $\varepsilon > 0$, then $\mu = \frac{\varepsilon}{4\ln D}$ and the iteration number $t \ge 64(1 + \max_{\tau\in\mathcal{D}} \|\widetilde{X}_\tau\|^2) \ln D/\varepsilon^2$ yields that $\max_{S\in\mathcal{P}} f(S) - f(S_t^\mu) \le \varepsilon$. The time complexity of the approximate first-order method for DML-eig is of $\mathcal{O}\big(d^2/\varepsilon^2\big)$.

3.2 Approximate Frank-Wolfe Algorithm for LMNN-eig

We can easily extend the above approximate Frank-Wolfe algorithm to solve the eigenvalue optimization formulation of LMNN-eig (formulation (16) or (17)). To this end, let

$$f(S,\xi) = \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau\big(\xi_\tau + \langle \widetilde{C}_\tau, S\rangle\big).$$

Then, problem (16) is identical to

$$\max\big\{f(S,\xi) : (1-\gamma)\sum_\tau \xi_\tau + \gamma \mathbf{Tr}(S) = 1, S \in \mathbb{S}_+^d,\ \xi \ge 0\big\}.$$

In analogy to the smooth techniques applied to DML-eig, we approximate $f(S,\xi)$ by the following smooth function:

$$f_\mu(S,\xi) = \min_{u\in\triangle} \sum_{\tau\in\mathcal{D}} u_\tau(\xi_\tau + \langle \widetilde{C}_\tau, S\rangle) + \mu \sum_{\tau\in\mathcal{D}} u_\tau \ln u_\tau.$$

One can easily see that

$$f_\mu(S,\xi) = -\mu \ln\big(\sum_{\tau\in\mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}\big).$$

Input:
· smoothing parameter $\mu > 0$ (e.g., 10^{-5})
· tolerance value *tol* (e.g., 10^{-5})
· step sizes $\{\alpha_t \in (0,1) : t \in \mathbb{N}\}$
Initialization: $S_1^\mu \in \mathbb{S}_+^d$ with $\mathbf{Tr}(S_1^\mu) = 1$ and $\xi_1^\mu \geq 0$
for $t = 1, 2, 3, \ldots$ **do**
· $(Z_t^\mu, \beta_t^\mu) = \arg\max \big\{ \langle Z, \partial_S f_\mu(S_t^\mu, \xi_t^\mu)\rangle + \xi^\top \partial_\xi f_\mu(S_t^\mu, \xi_t^\mu) : Z \in \mathbb{S}_+^d,\ \xi \geq 0$
$(1-\gamma)\xi^\top \mathbf{1} + \gamma \mathbf{Tr}(Z) = 1 \big\}$
· $(S_{t+1}^\mu, \xi_{t+1}^\mu) = (1-\alpha_t)(S_t^\mu, \xi_t^\mu) + \alpha_t (Z_t^\mu, \beta_t^\mu)$
· if $|f_\mu(S_{t+1}^\mu, \xi_{t+1}^\mu) - f_\mu(S_t^\mu, \xi_t^\mu)| < tol$ then **break**
Output: $d \times d$ matrix $S_t^\mu \in \mathbb{S}_+^d$ and slack variables ξ_t^μ

Table 2: Approximate Frank-Wolfe Algorithm for LMNN-eig

and its gradient function is given by

$$\nabla_S f_\mu(S, \xi) = \frac{\sum_{\tau \in \mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu} \widetilde{C}_\tau}{\sum_{\tau \in \mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}},$$

and

$$\frac{\partial f_\mu(S, \xi)}{\partial \xi_\tau} = \frac{e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}}{\sum_{\tau \in \mathcal{T}} e^{-(\langle \widetilde{C}_\tau, S\rangle + \xi_\tau)/\mu}}.$$

The approximate Frank-Wolfe algorithm for LMNN-eig is exactly the same as DML-eig in Table 1. The pseudo-code is listed in Table 2. The key step of the algorithm is to compute the following problem:

$$\begin{aligned}(Z_t^\mu, \beta_t^\mu) \quad = \arg\max \big\{ &\langle Z, \partial_S f_\mu(S_t^\mu, \xi_t^\mu)\rangle + \xi^\top \partial_\xi f_\mu(S_t^\mu, \xi_t^\mu) : Z \in \mathbb{S}_+^d,\ \xi \geq 0 \\ &(1-\gamma)\xi^\top \mathbf{1} + \gamma \mathbf{Tr}(Z) = 1 \big\}.\end{aligned}$$

Equivalently, one needs to solve, for any $A \in \mathbb{S}^d$ and $\beta \in \mathbb{R}^T$, the following problem:

$$(Z^*, \xi^*) \quad = \arg\max \big\{ \langle Z, A\rangle + \xi^\top \beta : Z \in \mathbb{S}_+^d,\ \xi \geq 0, (1-\gamma)\xi^\top \mathbf{1} + \gamma \mathbf{Tr}(Z) = 1 \big\}. \tag{24}$$

Let $\beta_{\max} = \beta_{\tau^*}$ with $\tau^* \in \mathcal{T}$ and v^* is the largest eigenvector of A. Then, problem (24) is a linear programming and its optimal value is either

$$\max \big\{ \xi^\top \beta : (1-\gamma)\xi^\top \mathbf{1} = 1, \xi \geq 0 \big\} = \frac{\beta_{\max}}{1-\gamma},$$

or

$$\max \big\{ \langle Z, A\rangle : \gamma \mathbf{Tr}(Z) = 1, Z \in \mathbb{S}_+^d \big\} = \frac{\lambda_{\max}(A)}{\gamma}.$$

The optimal solution of problem (24) is given as follows. If $\frac{\lambda_{\max}(A)}{\gamma} >= \frac{\beta_{\max}}{1-\gamma}$, then $Z^* = \frac{v^*(v^*)^\top}{\gamma}$ where v^* is the largest eigenvector of matrix A and $\xi^* = \mathbf{0}$. Otherwise, $Z^* = \mathbf{0}$ and the τ^*-th element of ξ^* equals $\frac{1}{1-\gamma}$, that is, $(\xi^*)_{\tau^*} = \frac{1}{1-\gamma}$ and the other entries of ξ^* all zeros. In analogy to the arguments for Theorem 7, for step sizes $\{\alpha_t = \frac{2}{t+1} : t \in \mathbb{N}\}$ one can exactly prove the time complexity of LMNN-eig is $\mathcal{O}(d^2/\varepsilon^2)$.

4. Related Work and Discussion

There is a large amount of work on metric learning including distance metric learning for k-means clustering (Xing et al., 2002), relevant component analysis (RCA) (Bar-Hillel et al., 2005), maximally collapsing metric learning (MCML) (Goldberger et al., 2004), neighborhood component analysis (NCA) (Goldberger et al., 2004) and an information-theoretic approach to metric learning (ITML) (Davis et al., 2007) etc. We refer the readers to Yang and Jin (2007) for a nice survey on metric learning. Below we discuss some specific metric learning models which are closely related to our work.

Xing et al. (2002) developed the metric learning model (2) to learn a Mahalanobis metric for k-means clustering. The main idea is to maximize the distance between points in the dissimilarity set under the constraint that the distance between points in the similarity set is upper-bounded. A projection gradient method is employed to obtain the optimal solution. Specifically, at each iteration the algorithm takes a gradient ascent step of the objective function and then projects it back to the set of constraints and the cone of the p.s.d. matrices. The projection to the p.s.d. cone needs the computation of the full eigen-decomposition with time complexity $\mathcal{O}(d^3)$. The projection gradient method usually takes a large number of iterations to become convergent. It is worth mentioning that the metric learning model proposed in Xing et al. (2002) is a global method in the sense that the model aggregates all similarity constraints together as well as all dissimilarity constraints. In contrast to Xing et al. (2002), DML-eig aims to maximize the minimal distance between dissimilar pairs instead of maximizing the summation of their distances. Consequently, DML-eig would intuitively force the dissimilar samples to be far more separated from similar samples. This intuition may account for the superior performance of DML-eig which will be shown soon in the experimental section.

Weinberger et al. (2005) developed a large margin framework to learn a Mahalanobis distance metric for k-nearest neighbor (k-NN) classification (LMNN). The main intuition behind LMNN is that k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. In contrast to the global method (Xing et al., 2002), LMNN is a local method in the sense that only triplets from the k-nearest neighbors are used. Our method DML-eig is a local method which only uses the similar pairs and dissimilar pairs from k-nearest neighbors.

Since every $M \in \mathbb{S}_+^d$ can be factored as $M = AA^\top$ for some $A \in \mathbb{R}^{d\times d}$, LMNN becomes an unconstrained optimization problem with an unconstrained variable A. Weinberger et al. (2005) used this idea and proposed to use the sub-gradient method to obtain the optimal solution. Since the modified problem w.r.t. variable A is generally not convex, the sub-gradient method would lead to local minimizers. For some special SDP problems, it was shown in Burer and Monteiro (2003) that such dilemma will not happen. Specifically, Burer and Monteiro (2003) considered the following SDPs:

$$\min\Big\{\mathbf{Tr}(CM) : \mathbf{Tr}(A_iM) = b_i, i = 1,\ldots,m, M \in \mathbb{S}_+^d\Big\}. \tag{25}$$

It was proved that if A^* is a local minimum of the modified problem:

$$\min\Big\{\mathbf{Tr}(CAA^\top) : \mathbf{Tr}(A_iAA^\top) = b_i, i = 1,\ldots,m, A \in \mathbb{R}^{d\times d}\Big\},$$

then $M^* = A^*(A^*)^\top$ is a global minimum of the primal problem (25). However, since the hinge loss is not smooth, it is unclear how their proof can be adapted to the case of LMNN.

Rosales and Fung (2006) proposed the following element-sparse metric learning for high-dimensional data sets

$$\min_{M\in\mathbb{S}^d_+} \sum_{t=(i,j,k)\in\mathcal{T}} (1 + x_{ij}^\top M x_{ij} - x_{kj}^\top M x_{kj})_+ + \gamma \sum_{\ell,k\in\mathbb{N}_d} |M_{\ell k}|. \tag{26}$$

In order to solve the optimization problem, they further proposed to restrict M to the space of *diagonal dominance* matrices which reduces formulation (26) to a linear programming problem. Such a restriction would only result in a sub-optimal solution.

Shalev-Shwartz et al. (2004) developed an appealing online learning model for learning a Mahalanobis distance metric. In each time, given a pair of examples the p.s.d. distance matrix is updated by a rank-one matrix which only needs the time complexity $\mathcal{O}(d^2)$. However, since the pairs of similarly labeled and differently labeled examples are usually of order $\mathcal{O}(n^2)$, the online learning procedure takes many rank-one matrix updates. Jin et al. (2009) established generalization bounds for large margin metric learning and proposed an adaptive way to adjust the step sizes of the online metric learning method in order to guarantee the output matrix in each step is positive semi-definite. Since the pairs of similarity and dissimilarity are usually of order $\mathcal{O}(n^2)$ where n is the sample number, the online learning procedure generally needs many matrix updates.

Shen et al. (2009) recently employed the exponential loss for metric learning which can be written by

$$\min_{M\in\mathbb{S}^d_+} \sum_{\tau=(i,j,k)\in\mathcal{T}} e^{\langle C_\tau, M\rangle} + \mathbf{Tr}(M),$$

where $\mathcal{T}$ is the triplet set and $C_\tau = (x_i - x_j)(x_i - x_j)^\top - (x_j - x_k)(x_j - x_k)^\top$ for any $\tau = (i, j, k) \in \mathcal{T}$. A boosting-based algorithm called BoostMetric was developed which is based on the idea that each p.s.d. matrix can be decomposed into a linear positive combination of trace-one and rank-one matrices. The algorithm is essentially a column-generation scheme which iteratively finds the linear combination coefficients of the current basis set of rank-one matrices and then update the basis set of trace-one and rank-one matrices. The updating of rank-one and trace-one matrix only involves the computation of the largest eigenvector which is of time complexity $\mathcal{O}(d^2)$. However, the number of linear combination for the p.s.d. matrix can be infinite and the convergence rate of this column-generation algorithm is not clear.

Recently, Guillaumin et al. (2009) proposed a metric learning model with logistic regression loss which is referred to as LDML. Promising results were reported in its application to face verification problems. LDML employed the gradient descent algorithm to obtain the optimal solution. However, in order to reduce the computational time, the algorithm ignored the positive semi-definiteness of the distance matrix which would only lead to a suboptimal solution.

5. Experiments

In this section we compare our proposed method **DML-eig** and **LMNN-eig** with a few methods: the method proposed in Xing et al. (2002) denoted by **Xing**, **LMNN** (Weinberger

Data	No.	n	d	♯class	$\sharp\,\mathcal{T}$	$\sharp\mathcal{D}$
Wine	1	178	13	3	1134	378
Iris	2	150	4	3	954	315
Breast	3	569	30	2	3591	1197
Diabetes	4	768	8	2	4842	1614
Waveform	5	5000	21	3	3150	1050
Segment	6	2310	19	7	14553	4851
Optdigits	7	2680	64	10	24120	8040
Face	8	400	2576	40	2520	840
USPS	9	9298	256	10	58626	19542

Table 3: Description of data sets n is the number of samples and d is the dimensionality. For AT&T face data set, we use PCA to reduce its dimension to 64.

et al., 2005) and its accelerated version **mLMNN** (Weinberger and Saul, 2008), **ITML** (Davis et al., 2007), **BoostMetric** (Shen et al., 2009) and the baseline algorithm that uses the standard Euclidean distance denoted by **Euc.** For all the data sets we have set $k = 3$ for nearest neighbor classification. The trade-off parameters in ITML, LMNN and LMNN-eig are tuned via three-fold cross validation. The smoothing parameter for DML-eig and LMNN-eig is set to be $\mu = 10^{-4}$ and the maximum iteration for DML-eig, BoostMetric, LMNN-eig is set to be 10^3.

We first run experiments on 9 data sets, that is, 1) wine, 2) iris, 3) Breast-Cancer, 4) the Indian Pima Diabetes, 5) Waveform, 6) Segment, 7) Optdigits, 8) AT&T Face data set [2] and 9) USPS. The statistics of data sets summarized in Table 3. All experimental results are obtained by averaging over 10 runs (except 1 run for USPS due to its large size). For each run, we randomly split the data sets 70% for training and 30% for test validation. We have used the same mechanism in Weinberger et al. (2005) to generate training triplets. Briefly speaking, for each training point x_i, k nearest neighbors that have same labels as y_i (targets) as well as k nearest neighbors that have different labels from y_i (imposers) are found. From x_i and its corresponding targets and imposers, we then construct the set of similar pairs $\mathcal{S}$ (same labels) and the set of dissimilar pairs $\mathcal{D}$ (distinct labels), and the set of triplets $\mathcal{T}$. As mentioned above, the original formulation in Xing et al. (2002) used all pairwise constraints. We emphasize here, for fairness of comparison (especially the running time comparison), that all methods including the Xing's method used the same set of similar/dissimilar pairs generated locally as above.

Finally we will apply the developed models and algorithms on a large and challenging face verification data set called *Labeled Faces in the Wild* (LFW).[3] It contains 13233 labeled faces of 5749 people, for 1680 people there are two or more faces. Furthermore, the data is challenging and difficult due to face variations in scale, pose, lighting, background, expression, hairstyle, and glasses, as the faces are detected in images in the wild, taken from Yahoo! News.

2. Data sets can be found at `http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html`.
3. Data set can be found at `http://vis-www.cs.umass.edu/lfw/index.html`.

	Euc.	Xing	LMNN	ITML	BoostMetric	DML-eig	LMNN-eig
1	3.46(3.60)	4.04(4.00)	3.08(2.07)	**1.15(2.07)**	2.31(2.18)	1.35(1.30)	2.88(1.87)
2	5.11(2.58)	6.67(3.11)	4.22(1.95)	4.44(2.57)	3.56(2.52)	**3.11(1.15)**	4.00(2.30)
3	6.47(1.33)	8.18 (1.58)	5.35(1.43)	6.82(1.57)	3.82(1.55)	**3.53(0.88)**	4.94(1.28)
4	31.09(2.03)	32.09 (3.56)	29.70(3.20)	29.96(2.97)	**26.78(2.12)**	27.71(3.93)	31.13(2.24)
5	18.87(0.65)	16.43(1.00)	18.61(0.72)	15.94(0.83)	16.86(0.90)	**15.33(0.80)**	18.49(0.21)
6	5.61(0.92)	5.26(0.60)	3.69(0.70)	5.02(0.70)	4.21(0.48)	**2.97(0.55)**	3.61(0.83)
7	1.67(0.24)	1.57(0.28)	**1.37(0.25)**	1.46(0.29)	1.38(0.33)	1.45(0.22)	1.43(0.42)
8	6.67(1.67)	7.75(0.69)	2.08(1.53)	2.42(2.17)	2.25(1.25)	**1.67(1.24)**	**1.67(1.76)**
9	3.05	-	**2.98**	3.92	3.34	3.66	3.13

Table 4: Average test error (%) of different metric learning methods (standard deviation are in parentheses). The best performance is denoted in bold type. The notation "–" means that the method does not converge in a reasonable time.

data	Xing	LMNN/mLMNN	ITML	BoostMetric	LMNN-eig	DML-eig
1	1.00	0.87/1.01	4.63	0.49	0.30	0.23
2	2.41	0.57/0.62	3.56	0.10	0.92	0.43
3	3.08	2.71/0.75	4.54	2.04	3.71	3.18
4	2.45	1.73/1.03	3.95	0.20	6.78	0.03
5	231.33	8.83/5.54	7.83	11.36	36.95	1.45
6	109.13	1.73/4.25	61.55	9.06	5.06	1.76
7	59.24	24.81/15.92	37.42	93.73	86.38	2.67
8	182.56	5.54/1.50	40.38	60.31	18.42	2.58
9	–	723.49/454.21	726.88	694.84	572.04	52.48

Table 5: Average running time (seconds) of different methods. The notation "–" means that the method does not converge in a reasonable time.

5.1 Generalization and Running Time

As we can see from Table 4, DML-eig consistently improves k-NN classification using Euclidean distance on most data sets. Hence, learning a Mahalanobis metric from training data does lead to improvements in k-NN classification. Also, we can see that DML-eig is competitive with the state-of-the-art methods: LMNN, ITML and BoostMetric. Indeed, DML-eig outperforms other algorithms on 5 out of 9 data sets. As expected, LMNN-eig performs similarly or slightly better than LMNN since these two models are essentially the same. In Table 5, we list the average CPU time of different algorithms. We can see that the method proposed in Xing et al. (2002) generally needs more time since it needs the full eigen-decomposition of a matrix per iteration. DML-eig, BoostMetric and LMNN are among the fastest algorithms while LMNN-eig is slower than LMNN and mLMNN in most cases. The accelerated version mLMNN is faster than LMNN.

On the left-hand side of Figure 1, we plot the running time versus the reduced dimension by principal component analysis (PCA) for AT&T data set. We can observe that

LMNN, BoostMetric, LMNN-eig and DML-eig are faster than ITML and Xing's method. When the dimension is low, LMNN, BoostMetric, LMNN-eig and DML-eig are similar. As the dimension increases, DML-eig and mLMNN are faster. On this data set, LMNN-eig runs slower than mLMNN. The reason could be that mLMNN used the techniques of ball trees and employed only an active set of triplets per iteration. Our algorithms have not been combined with the techniques of ball trees and are implemented in MATLAB and better improvements are expected if used in C/C++. On the right-hand side of Figure 1, we also plot the test errors of various methods across different PCA dimensions. Almost every method performs better than the baseline method using the standard Euclidean distance metric. DML-eig performs slightly better than other methods. We observe that, with increasing PCA dimensions, DML-eig, BoostMetric and ITML yield relatively stable performance across different PCA dimensions. In contrast, the performance of other baseline methods such as LMNN and Xing's method varied as the PCA dimensions changed.

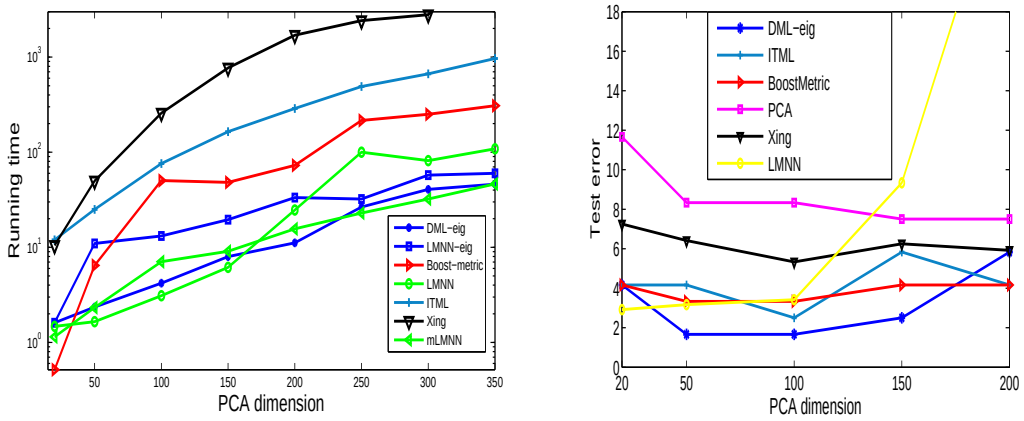

Figure 1: Performance on AT&T Face data set. Left figure: running time (seconds) versus PCA dimension. Right figure: test error (%) versus PCA dimension; the pink line is the performance of k-NN classifier ($k = 3$) using the standard Euclidean distance.

5.2 Application to Face Verification

In this experiment we investigate our proposed method (DML-eig) for face verification. The task of face verification is to determine whether two face images are from the same identity or not. It is a highly active area of research and finds application in access control, image search, security and many other areas. The large variation in lighting, pose, expression etc. of the face images poses great challenges to the face verification algorithms. Inference that is based on the raw pixels of the image data or features extracted from the images is usually unreliable as the data show large variation and are high-dimensional.

Metric learning provides a viable solution by comparing the image pairs based on the metric learnt from the face data. Here we evaluate our new metric learning method using a large scale face database—Labeled Faces in the Wild (LFW) (Huang et al., 2007). There are

a total of 13233 images and 5749 people in the database. These face images are automatically captured from news articles on the web. Recently it has become a benchmark to test new face verification algorithms (Wolf et al., 2008; Guillaumin et al., 2009; Wolf et al., 2009; Taigman et al., 2009; Pinto et al., 2011).

The images we used are in gray scale and aligned in two ways. One is "funneled" (Huang et al., 2007) and the other is "aligned" using a commercial face alignment software by Taigman et al. (2009). These images are divided into ten folds where the subject identities are mutually exclusive. In each fold, there are 300 pairs of images from the same identity and another 300 pairs of images from different identities. We followed the standard procedure for training and test in the technical report of Huang et al. (2007). The performance of the algorithms is evaluated by average (and standard error of) correct verification rate and the ROC curve of the 10-fold cross validation test.

We investigated several descriptors (features) from face images in this experiment. As for the "funneled" images, we used SIFT descriptors computed at the fixed facial key-points (e.g., corners of eyes and nose). These data are available from Guillaumin et al. (2009). We focus on the SIFT descriptor to evaluate our algorithm as it provides a fair comparison to Guillaumin et al. (2009). To compare with the state-of-the-art methods in face verification, we further investigated three types of features for the "aligned" images: 1) raw pixel data by concatenating the intensity value of each pixel in the image; 2) Local Binary Patterns (LBP) (Ojala et al., 2002); and 3) LBP's variation, three-Patch Local Binary Patterns (TPLBP) (Wolf et al., 2008). The original dimensionality of the features is quite high ($3456 \sim 12000$) so we reduced the dimension using PCA. These descriptors were tested with both their original value and the square root of them (Wolf et al., 2008, 2009; Guillaumin et al., 2009).

There are two configuration for forming the training sets. One is "restricted configuration": only same/not-same labels are used during training and no information about the actual names of the people (class labels) in the image pairs should be used. In the past, most of the published work on this data set using the restricted protocol (e.g., Guillaumin et al., 2009; Wolf et al., 2009; Pinto et al., 2011). Another is "unrestricted configuration": all available information including the names of the people in the images can be used for training. So far there are only two published results on the unrestricted configuration (Guillaumin et al., 2009; Taigman et al., 2009). Here we mainly focus on the restricted configuration.

LMNN and BoostMetric are not applicable in this restricted configuration setting since they need label information to generate the triplet set. Therefore, we only compared our DML-eig method with LDML (Guillaumin et al., 2009) and ITML (Davis et al., 2007). For each of the ten-fold cross-validation test, we use the data from 2700 pairs of images from the same identities and another 2700 pairs of images from the different identities to learn a metric. Then test it using the other 600 image pairs. The performance is evaluated using accurate verification rate .

Table 6 illustrates the performances of our algorithm and ITML and LDML. The best verification rate of DML-eig is 81.27%. It outperforms LDML (77.50%) and ITML (76.20%) in their best settings. Note that the performance of DML-eig is consistently better than LDML and ITML in each PCA dimension.

By varying the dimension of principal components of the SIFT descriptor, the performance of DML-eig of the 10-fold cross validation test is plotted in Figure 2. The best

Method	PCA Dim.	Original	Square Root
ITML	35	0.7537 ± 0.0158	0.7627 ± 0.0161
LDML	35	0.7660 ± 0.0070	0.7750 ± 0.0050
DML-eig	35	0.7742 ± 0.0213	0.7793 ± 0.0214
ITML	40	0.7618 ± 0.0125	0.7643 ± 0.0121
LDML	40	–	–
DML-eig	40	0.7752 ± 0.0198	0.7838 ± 0.0195
ITML	55	0.7530 ± 0.0185	0.7557 ± 0.0187
LDML	55	0.7280 ± 0.0060	0.7280 ± 0.0040
DML-eig	55	0.7900 ± 0.0189	0.7938 ± 0.0163
ITML	100	0.7340 ± 0.0250	0.7403 ± 0.0216
LDML	100	–	–
DML-eig	100	$\mathbf{0.8055 \pm 0.0171}$	$\mathbf{0.8127 \pm 0.0230}$

Table 6: Performance comparison on LFW database in the restricted configuration (mean verification accuracy and standard error of the mean of 10-fold cross validation test) with only SIFT descriptors. "Square Root" means the features preprocessed by taking square root before fed into metric learning method. The result of LDML is cited from Guillaumin et al. (2009) where it was reported that the best result of LDML is achieved with PCA dimension 35. Our result of ITML is very similar to that reported in Guillaumin et al. (2009).

Method	Accuracy
High-Throughput Brain-Inspired Features, aligned (Pinto et al., 2011)	0.8813 ± 0.0058
LDML + Combined, funneled (Guillaumin et al., 2009)	0.7927 ± 0.0060
DML-eig + Combining four descriptors (this work)	0.8565 ± 0.0056

Table 7: Performance comparison of DML-eig and other state-of-the-art methods in the restricted configuration (mean verification rate and standard error of the mean of 10-fold cross validation test) based on combination of different types of descriptors. The descriptors vary in different study. The best result up to date is achieved using sophisticated large scale feature search (Pinto et al., 2011).

performance is achieved when the dimension of principal components is 100. So we fix this dimension for SIFT feature in the following experiment. As mentioned in Guillaumin et al. (2009), the peak performance in a specific PCA dimension is due to the limit of training samples. The PCA dimension achieving the best performance is 35 for LDML and 55 for ITML. This number for DML-eig is 100 which is larger than that of both LDML and ITML. It shows that the DML-eig metric is less prone to overfitting than both LDML and ITML.

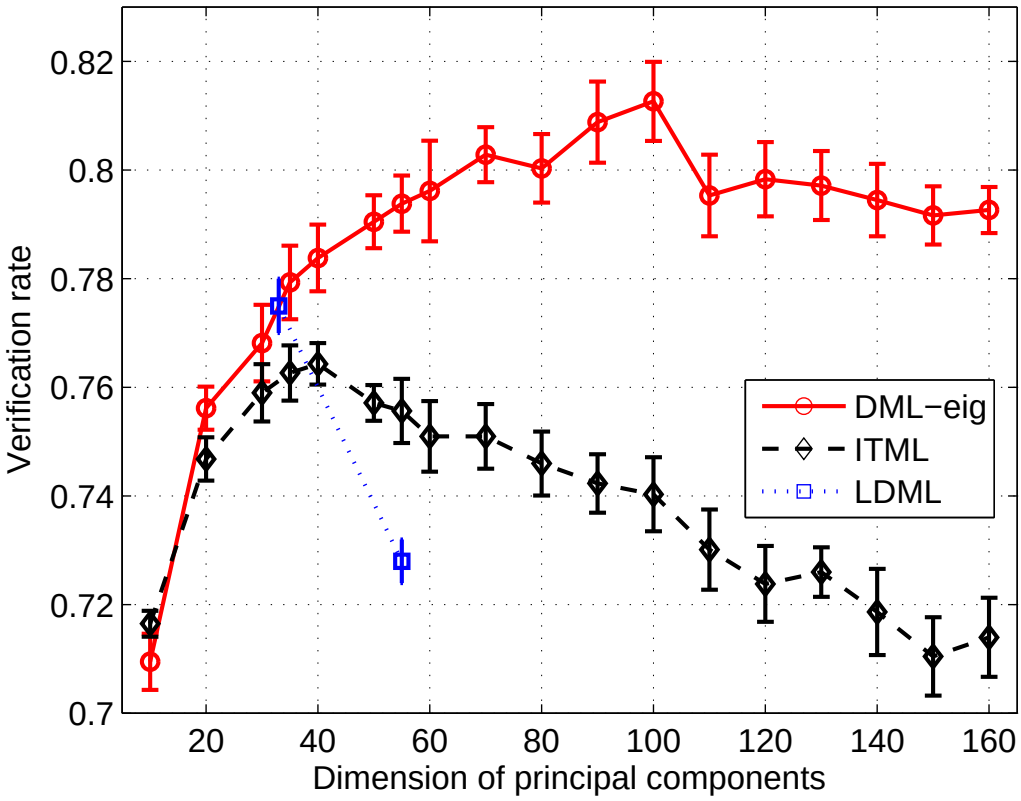

Figure 2: Performance of DML-eig, ITML and LDML metric by varying the dimension of the principal components using SIFT descriptor. The result of LDML is copied from Guillaumin et al. (2009).

Besides the SIFT descriptor, we also investigated to combine it with other three types of descriptors aforementioned. Following Wolf et al. (2008); Guillaumin et al. (2009), we combine the distance scores from 4 different descriptors using a linear Support Vector Machine (SVM). The performance of DML-eig is compared to the other state-of-the-art methods in Table 7 and Figure 3. Note that each of these published results use its own learning technique and different feature extraction approaches which makes the conclusion hard to draw.

The best result reported to date is 88.13% in restricted configuration which performs sophisticated large scale feature search (Pinto et al., 2011). This work used multiple complimentary representations which are derived through training set augmentation, alternative face comparison functions, and feature set searches with a varying number of model layers. These individual feature representations are then combined using kernel techniques. The results by other state-of-the-art methods are also based on different descriptors (Guillaumin et al., 2009; Wolf et al., 2009). The best result achieved by DML-eig is 85.65%, which is close to the other state-of-the-art approaches. In addition, we note that the performance

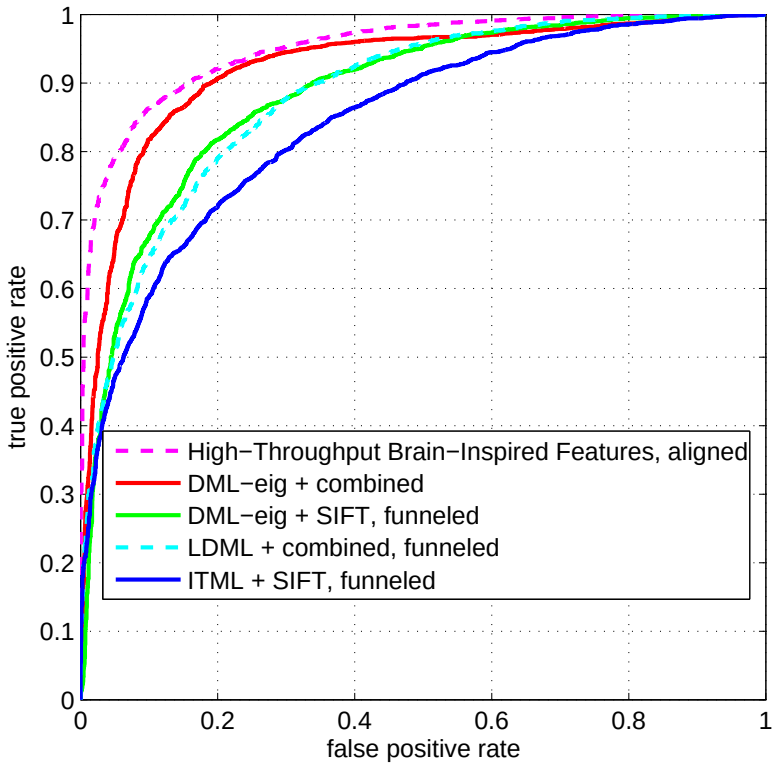

Figure 3: ROC curve of DML-eig and other the state of arts methods for face verification on LFW data set.

of DML-eig based on the single SIFT descriptor (81.27% in Table 6) is better than that of LDML based on 4 types of descriptors (79.27% in Table 7). The ROC curves of different methods are depicted in Figure 3. We can see that DML-eig outperforms ITML and LDML while it is suboptimal to the best up-to-date method (Pinto et al., 2011) which, however, employed sophisticated feature search method.

Finally, the performance of DML-eig metric may be further improved by exploring different number of nearest neighbors and different types of descriptors such as those used in Pinto et al. (2011), making it a competitive candidate for the task of face verification.

6. Conclusion

The main theme of this paper is to develop a new eigenvalue-optimization framework for metric learning. Within this context, we first proposed a novel metric learning model which was shown to be equivalent to a well-known eigenvalue optimization problem (Overton, 1988; Lewis and Overton, 1996). This appealing optimization formulation was further extended to LMNN (Weinberger et al., 2005) and maximum margin matrix factorization (Srebro et al., 2004). Then, we developed efficient first-order algorithms for metric learning which

only involve the computation of the largest eigenvector of a matrix. Their convergence rates were rigorously established. Finally, experiments on various data sets have shown that our proposed approach is competitive with state-of-the-art metric learning methods. In particular, we reported promising results on the Labeled Faces in the Wild (LFW) data set.

In future we will exploit the extension of the above eigenvalue optimization framework to other machine learning tasks such as spectral graph cuts and semi-definite embedding (Weinberger et al., 2004). Another direction for investigation is to develop a kernelized version of DML-eig using the techniques in Jain et al. (2010). Finally, we will also investigate the performance of our methods on the LFW data set in the unrestricted configuration setting, and embed the technique of ball trees (Weinberger and Saul, 2008) into our algorithms to further increase the computational speed.

Acknowledgments

The authors would like to thank Colin Campbell, Massimiliano Pontil, and Charles Micchelli for stimulating discussion and invaluable comments on the preliminary version of this paper. The authors also sincerely thank the anonymous reviewers for their comments and suggestions which have led to valuable improvements of this paper. This work is supported by the EPSRC under grant EP/J001384/1. The second author would like to thank Cancer Research UK for the research grant.

Appendix A. Eigenvalue Optimization for Maximum-margin Matrix Factorization

Another important problem is low-rank matrix completion which recently has attracted much attention. This line of research involves computing a large matrix with a nuclear-norm (summation of singular values) regularization and the optimization problem here also consists of an SDP. Such tasks include multi-task feature learning (Argyriou et al., 2006) and low-rank matrix completion (Bach, 2008; Candes and Recht, 2008; Srebro et al., 2004). It has successful applications to collaborative filtering for predicting customers' preferences to products, where the matrix's rows and columns respectively identify the "customers" and "products", and a matrix entry encodes customers' preference of a product (e.g., Netflix data set, `http://www.netflixprize.com/`).

Similar eigenvalue optimization formulation can be developed for maximum-margin matrix factorization (MMMF) for collaborative filtering (Srebro et al., 2004). Given a partially labeled $Y_{ia} \in \{\pm 1\}$ with $ia \in S$, the target of MMMF is to learn a large matrix $X \in \mathbb{R}^{m \times n}$ where each entry X_{ia} indicates the preference of the customer i for product a. The following large margin model was proposed in Srebro et al. (2004) to learn X:

$$\begin{array}{ll} \min_X & \sum_{ia \in S} \xi_{ia} + \gamma \|X\|_* \\ \text{s.t.} & 1 - Y_{ia} X_{ia} \leq \xi_{ia}, \\ & \xi_{ia} \geq 0, \ \forall ia \in S, \end{array}$$

where $\|X\|_*$ is the nuclear norm of X, that is, the summation of its singular values. The above model was further formulated as an SDP problem:

$$\begin{array}{ll} \min_M & \gamma \mathbf{Tr}(M) + \sum_{ia\in S} \xi_{ia} \\ & M = \begin{pmatrix} A & X \\ X^\top & B \end{pmatrix} \in \mathcal{S}_+^{(m+n)}, \\ & Y_{ia} X_{ia} + \xi_{ia} \geq 1, \ \forall ia \in S. \end{array} \tag{27}$$

Let e_i be a column vector with its i-th element one and all others zero, then we have $M_{i(m+a)} = X_{ia} = \langle C_{ia}, M\rangle$ with $C_{ia} = e_i e_{(m+a)}^\top$. Consequently, the constraint condition in problem (27) can be written as $\min_{ia\in S}\langle Y_{ia}, C_{ia}\rangle + \xi_{ia} \geq 1$. Using exact arguments for proving Theorem 3, we can formulate MMMF as an eigenvalue optimization problem.

Theorem 8 *MMMF formulation (27) is equivalent to*

$$\max\Big\{\min_{u\in\triangle} \sum_{ia\in S} u_{ia}\big(\xi_{ia} + \langle Y_{ia}C_{ia}, M\rangle\big) : \xi^\top \mathbf{1} + \gamma \mathbf{Tr}(M) = 1, M \in \mathcal{S}_+^{(m+n)}, \ \xi \geq 0\Big\}.$$

In particular it is equivalent to the following eigenvalue optimization problem:

$$\min_{u\in\triangle} \max\Big(u_{max}, \frac{1}{\gamma}\lambda_{max}\big(\sum_{ia\in S} u_{ia} Y_{ia} C_{ia}\big)\Big). \tag{28}$$

As mentioned above, MMMF (27) is a standard SDP. Indeed, Srebro et al. (2004) proposed to employ standard SDP solvers (e.g., CSDP Borchers, 1999) to obtain the optimal solution. However, such generic solvers are only able to handle problems with about a hundred users and a hundred items. The eigenvalue-optimization formulation potentially provides more efficient algorithms for MMMF. Since the paper mainly focuses on metric learning, we leave its empirical implementation for future study.

Appendix B. Proof of Theorem 6

In this appendix we give the proof of Theorem 6. The spirit of the proof is very close to that of Theorem 1 in Ying and Zhou (2006) where similar conditions on step sizes were derived to guarantee the convergence of stochastic online learning algorithms in reproducing kernel Hilbert spaces.

Proof [Proof of Theorem 6] According to the assumption (22) on the step size, we can assume that, for any $t \geq t_0$, that $\alpha_t \leq 1/2$. Hence, the inequality (21) holds true. We will estimate the terms on the left-hand side of (21) one by one.

For the second term on the righthand side of (21), observe that $\prod_{j=t_0}^t (1-\alpha_j) \leq \exp\{-\sum_{j=t_0}^t \alpha_j\} \to 0$ as $t \to \infty$. Therefore, for any $\varepsilon > 0$ there exists some $t_1 \in \mathbb{N}$ such that the second term on the righthand side of (21) is bounded by ε whenever $t \geq t_1$.

To deal with the first term on the righthand side of (21), we use the assumption $\lim_{j\to\infty} \alpha_j = 0$ and know that there exists some $j(\varepsilon)$ such that $\alpha_j \leq \varepsilon$ for every $j \geq j(\varepsilon)$. Write

$$\sum_{j=t_0}^t \alpha_j^2 \prod_{k=j+1}^t (1-\alpha_k) = \sum_{j=t_0}^{j(\varepsilon)} \alpha_j^2 \prod_{k=j+1}^t (1-\alpha_k) + \sum_{j=j(\varepsilon)+1}^t \alpha_j^2 \prod_{k=j+1}^t (1-\alpha_k). \tag{29}$$

Since $j(\varepsilon)$ is fixed, we can find some $t_2 \in \mathbb{N}$ such that for each $t \geq t_2$, there holds $\sum_{j=t(\varepsilon)+1}^{t} \alpha_j \geq \sum_{j=j(\varepsilon)+1}^{t_2} \alpha_j \geq \log \frac{j(\varepsilon)}{4\varepsilon}$. It follows that for each $1 \leq j \leq j(\varepsilon)$, there holds $\prod_{k=j+1}^{t}(1 - \alpha_k) \leq \exp\{-\sum_{k=j+1}^{t} \alpha_k\} \leq \exp\{-\sum_{k=j(\varepsilon)+1}^{t} \alpha_k\} \leq \frac{4\varepsilon}{j(\varepsilon)}$. This in connection with the bound $\alpha_j \leq 1/2$ for each $j \geq t_0$ tells us that the first term of (29) is bounded as

$$\sum_{j=t_0}^{t(\varepsilon)} \alpha_j^2 \prod_{k=j+1}^{t} (1 - \alpha_k) \leq \frac{4\varepsilon}{j(\varepsilon)} \sum_{j=t_0}^{j(\varepsilon)} \alpha_j^2 \leq \varepsilon.$$

The second term on the righthand side of (29) is dominated by $\varepsilon \sum_{j=j(\varepsilon)+1}^{t-1} \alpha_j \prod_{k=j+1}^{t}(1-\alpha_k)$. Noting the fact that $\alpha_j = 1 - (1 - \alpha_j)$ implies

$$\begin{aligned}\sum_{j=j(\varepsilon)+1}^{t} \alpha_j \prod_{k=j+1}^{t} (1 - \alpha_k) &= \sum_{j=j(\varepsilon)+1}^{t} \Big[\prod_{k=j+1}^{t} (1 - \alpha_k) - \prod_{k=j}^{t} (1 - \alpha_k) \Big] \\ &= \Big[1 - \prod_{k=j(\varepsilon)+1}^{t} (1 - \alpha_k) \Big] \leq 1.\end{aligned}$$

Therefore, when $t \geq \max\{t_1, t_2\}$, combining the estimation with inequality (21), we have $R_{t+1} \leq (1 + C_\mu)\varepsilon$. This proves the theorem. ■

References

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*, 2006.

F. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9:1019-1048, 2008.

M. Baes and M. Bürgisser. Smoothing techniques for solving semi-definite programs with many constraints. IFOR Internal report, ETH, 2009. Available electronically via `http://www.optimization-online.org/DB-FILE/2009/10/2414.pdf`.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937-965, 2005.

B. Borchers. CSDP, a C library for semi-definite programming. *Optimization Methods and Software*, 11:613-623, 1999.

S. Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semi-definite programs via low-rank factorization. *Mathematical Programming*, 95:329-357, 2003.

E.J. Candes and B. Recht. Exact matrix completion via convex optimisation. *arXiv:0805.4471*, 2008. Available electronically via `http://arxiv.org/abs/0805.4471`.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively with application to face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 539–546, 2005.

J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 209–216, 2007.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quaterly*, 3:149–154, 1956.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems 17*, 2004.

M. Guillaumin, J. Verbeek and C. Schmid. Is that you? Metric learning approaches for face identification. In *IEEE 12th International Conference on Computer Vision*, pages 498–505, 2009.

E. Hazan. Sparse approximation solutions to semi-definite programs. *LATIN: Proceedings of the 8th Latin American Conference on Theoretical informatics*, 2008.

C. Helmberg and F. Rendl. A spectral bundle method for semi-definite programming. *SIAM Journal on Optimization*, 10(3):673-696, 1999.

S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2072–2078, 2006.

G. B. Huang, M. Ramesh, T. Berg and E. Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report* 07–49, October, 2007.

R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

P. Jain, B. Kulis, I. S. Dhillon. Inductive regularized learning of kernel functions. In *Advances in Neural Information Processing Systems 23*, 2010.

R. Jin, S. Wang and Y. Zhou. Regularized distance metric learning: theory and algorithm. In *Advances in Neural Information Processing Systems 22*, 2009.

T. Kato and N. Nagano. Metric learning for enzyme active-site search. *Bioinformatics*, 26:2698-2704, 2010.

A. S. Lewis and M. L. Overton. Eigenvalue optimization. *Acta Numerica*, 5:149–190, 1996.

A. Nemirovski. *Efficient methods in convex programming*. Lecture Notes, 1994.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2003.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127-152, 2005.

Y. Nesterov. Smoothing technique and its applications in semi-definite optimization. *Mathematical Programming*, 110:245–259, 2007.

T. Ojala, M. Pietikainen and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

M. L. Overton. On minimizing the maximum eigenvalue of a symmetric matrix. *SIAM. J. Matrix Anal. & Appl.* 9:256–268, 1988.

N. Pinto and D. Cox. Beyond simple features: a large-scale feature search approach to unconstrained face recognition. In *International Conference on Automatic Face and Gesture Recognition*, 2011.

R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

C. Shen, J. Kim, L. Wang and A. Hengel. Positive semi-definite metric learning with boosting. In *Advances in Neural Information Processing Systems 22*, 2009.

S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

N. Srebro, J.D.M. Rennie, and T.S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, 2004.

Y. Taigman and L. Wolf and T. Hassner. Multiple one-shots for utilizing class label information. In *The British Machine Vision Conference*, Sept. 2009.

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems 19*, 2007.

J.-P. Vert, J. Qiu and W. S. Noble. A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, 8(Suppl 10), 2007.

K. Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbour classification. In *Advances in Neural Information Processing Systems 18*, 2005.

K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, pages 1160–1167, 2008.

K. Q. Weinberger, F. Sha and L. K. Saul. Learning the kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

L. Wolf, T. Hassner and Y. Taigman. Similarity scores based on background samples. In *Asian Conference on Computer Vision*, 2009.

L. Wolf, T. Hassner and Y. Taigman. Descriptor based methods in the wild. In *Real-Life Images workshop at the European Conference on Computer Vision*, October, 2008.

E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. In *Advances in Neural Information Processing Systems 15*, 2003.

L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *Technical report, Department of Computer Science and Engineering, Michigan State University*, 2007.

L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. Hoi, and M. Satyanarayanan. A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32:30–44, 2010.

Y. Ying, K. Huang and C. Campbell. Sparse metric learning via smooth optimization. In *Advances in Neural Information Processing Systems 22*, 2009.

Y. Ying and D.X. Zhou, Online regularized classification algorithms. *IEEE Trans. Inform. Theory*, 11:4775-4788, 2006.