Distance Metric Learning Revisited

Qiong Cao[1], Yiming Ying[1], and Peng Li[2]

[1] College of Engineering, Mathematics and Physical Sciences,
University of Exeter, Harrison Building, Exeter, EX4 4QF, UK
{qc218,y.ying}@exeter.ac.uk
[2] Department of Engineering Mathematics,
University of Bristol, Bristol, BS8 1UB, UK
lipeng@ieee.org

Abstract. The success of many machine learning algorithms (e.g. the nearest neighborhood classification and k-means clustering) depends on the representation of the data as elements in a metric space. Learning an appropriate distance metric from data is usually superior to the default Euclidean distance. In this paper, we revisit the original model proposed by Xing et al. [25] and propose a general formulation of learning a Mahalanobis distance from data. We prove that this novel formulation is equivalent to a convex optimization problem over the spectrahedron. Then, a gradient-based optimization algorithm is proposed to obtain the optimal solution which only needs the computation of the largest eigenvalue of a matrix per iteration. Finally, experiments on various UCI datasets and a benchmark face verification dataset called Labeled Faces in the Wild (LFW) demonstrate that the proposed method compares competitively to those state-of-the-art methods.

Keywords: Metric learning, convex optimization, Frank-Wolfe algorithm, face verification.

1 Introduction

Many machine learning algorithms critically depend on the quality of the chosen distance metric. For instance, k-nearest neighbor classification needs the identification of nearest neighbors and k-means clustering depends on the distance measurements for clustering. The default distance is the Euclidean distance, which, however, does not reflect the given data representation. Recent advances in metric learning [1,2,4,6,19,20,22,23,25,27] make it possible to learn an effective distance metric which is more suitable for a given learning problem. These methods have demonstrated the successful applications of metric learning to various real-world problems including information retrieval and face verification.

Given some partial information of constraints, the goal of metric learning is to learn a distance metric which reports small distances for *similar* examples and large distances for *dissimilar* examples. The partial information can be presented in the form of constraints such as similarity or dissimilarity between a pair of examples. These constraints can be collected either from the label information

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 283–298, 2012.

in supervised classification or the side information in semi-supervised clustering such as must-links and cannot-links. Most of metric learning methods focus on learning a 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. Many metric learning methods for learning Mahalanobis distances are therefore formulated as semi-definite programs [21].

Depending on the generation of constraints information, metric learning can be supervised or unsupervised. Unsupervised metric learning is closely related to dimension reduction. To see this, observe that any positive semi-definite M can be rewritten as $A^\top A$, and hence, $d_M(x_i, x_j) = \sqrt{(x_i - x_j)^\top M(x_i - x_j)} = \|A(x_i - x_j)\|$. This simple observation implies that learning an appropriate M is equivalent to learning an appropriate projection map A. From this perspective, dimension reduction methods (e.g. [3,16,17]) can be regarded as unsupervised metric learning. In supervised metric learning, the available labels can be used to create the information of constraints. Supervised metric learning can be further divided into two categories: the global method and the local method. The global methods learn the distance metric which satisfies all the pairwise constraints simultaneously. The original model proposed by Xing et al. [25] is a global method which used all the similar pairs (same labels) and dissimilar pairs (distinct labels). Local methods only use local pairwise constraints which usually outperform the global ones as observed in many previous studies. This is particularly reasonable in the case of learning a metric for the kNN classifiers since kNN classifiers are influenced mostly by the data items that are close to the test/query examples. Since we are mainly concerned with metric learning for kNN classifier, the pairwise constraints are generated locally, that is, the similar/dissimilar pairs are k-nearest neighbors. The details can be found in the experimental section.

In this paper, we revisit the original model proposed by Xing et al. [25], where the authors proposed to learn a metric by maximizing the distance between dissimilar samples whilst keeping the distance between similar points upper-bounded. However, the projection gradient method employed there usually takes a large number of iterations to become convergent, and also it needs the full eigen-decomposition per iteration. The first contribution of this paper is to extend the methods in [25,28] and propose a general formulation for metric learning. We prove the convexity of this general formulation and illustrate it with various examples. Our second contribution is to show, by exploring its special structures, that the proposed formulation is further equivalent to a convex optimization over the spectrahedron. This equivalent formulation enables us to directly employ the Frank-Wolfe algorithm [5] to obtain the optimal solution. In contrast to the algorithm in [25], our proposed algorithm only needs to compute the largest eigenvalue of a matrix per iteration and is guaranteed to converge with a time complexity $\mathcal{O}(1/t)$ where t is the iteration number.

The paper is organized as follows. The next section presents the proposed model and proves its convexity. Section 3 establishes its equivalent formulation from which an efficient algorithm is proposed. In Section 4, we review and discuss

some related work on metric learning. Section 5 reports experimental results on UCI datasets and a benchmark face verification dataset called Labeled Faces in the Wild (LFW). The last section concludes the paper.

2 Convex Metric Learning Model

We begin by introducing some useful notations. For any $n \in \mathbb{N}$, denote $\mathbb{N}_n = \{1, 2, \ldots, n\}$. The space of symmetric $d \times d$ matrices is denoted by $\mathbb{S}^d$ and $\mathbb{S}^d_+$ denotes the cone of positive semi-definite matrices. For any $X, Y \in \mathbb{R}^{d\times n}$, the inner product in $\mathbb{S}^d$ is denoted by $\langle X, Y\rangle := \mathbf{Tr}(X^\top Y)$ where $\mathbf{Tr}(\cdot)$ is the trace of a matrix.

For simplicity, we focus on learning a distance metric for kNN classification, although the proposed methods below can easily be adapted to metric learning for k-means clustering. Now we denote the training data 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). Later on, we use the convention $X_{ij} = (x_i - x_j)(x_i - x_j)^\top$ and let $\mathcal{S}$ index the similarity pairs, $\mathcal{D}$ index the dissimilarity pairs. For instance, $\tau = (i, j) \in \mathcal{S}$ means that (x_i, x_j) is a similar pair and rewrite X_{ij} as X_τ. One can follow the mechanism in [22] to extract local information of similarity or dissimilarity for kNN classification; see the experimental section for more details.

Given a set of similar samples and a set of dissimilar samples, we aim to find a good distance matrix M such that the distance between the dissimilar pair is large while keeping the distance between the similar pairs small. There are many formulations to achieve this goal. In particular, the following formulation was proposed in [25]:

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

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

In this paper, we propose a more general formulation:

$$\begin{array}{cc} \max_{M\in\mathbb{S}^d_+} & \left[\sum_{(i,j)\in\mathcal{D}} [d_M(x_i, x_j)]^{2p}/D\right]^{\frac{1}{p}} \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}} [d_M(x_i, x_j)]^2 \le 1, \end{array} \tag{2}$$

where $p \in (-\infty, \infty)$ and D is the number of dissimilarity pairs. We refer to the above formulation as $\mathbf{DML}_p$. The above formulation is well defined even for the limiting case $p = 0$ as discussed in the examples below.

- $\mathbf{p = 1/2}$: In this case, problem (2) can be written as

$$\begin{array}{cc} \max_{M\in\mathbb{S}^d_+} & \left[\sum_{(i,j)\in\mathcal{D}} d_M(x_i, x_j)/D\right]^2 \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}} [d_M(x_i, x_j)]^2 \le 1, \end{array} \tag{3}$$

  which is equivalent to formulation (1) proposed in [25].

– $\mathbf{p} \to -\infty$: Observe, for any positive sequence $\{\alpha_i > 0 : i \in \mathbb{N}_n\}$, that

$$\lim_{p\to-\infty} \big(\sum_{i\in\mathbb{N}_n} a_i^p/n\big)^{\frac{1}{p}} = \min_{i\in\mathbb{N}_n} a_i.$$

Hence, in the limiting case $p \to -\infty$, problem (2) is reduced to the metric learning model called DML-eig [28]:

$$\begin{array}{ll} \max_{M\in\mathbb{S}_+^d} & \min_{(i,j)\in\mathcal{D}}[d_M(x_i,x_j)]^2 \\ \text{s.t.} & \sum_{(i,j)\in\mathcal{S}}[d_M(x_i,x_j)]^2 \le 1. \end{array} \tag{4}$$

– $\mathbf{p} \to \mathbf{0}$: Note, for any sequence $\{\alpha_i > 0 : i \in \mathbb{N}_n\}$, that

$$\lim_{p\to 0}\big[\sum_{i\in\mathbb{N}_n} a_i^p/n\big]^{\frac{1}{p}} = \prod_{i=1}^{n} \alpha_i^{\frac{1}{n}}.$$

Hence, in the limiting case $p \to 0$, problem (2) becomes

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

where D is the number of dissimilar pairs in the set $\mathcal{D}$.

The following theorem investigates the convexity/concavity of the objective function in problem (2).

Theorem 1. *Let function $\mathcal{L} : \mathbb{S}_+^d \to \mathbb{R}$ be the objective function of DML$_p$, i.e., for any $M \in \mathbb{S}_+^d$, $\mathcal{L}(M) = \big[\sum_{(i,j)\in\mathcal{D}}\langle X_{ij}, M\rangle^p/D\big]^{\frac{1}{p}}$ for $p \neq 0$, and $\mathcal{L}(M) = \prod_{(i,j)\in\mathcal{D}}[d_M(x_i,x_j)]^{\frac{2}{D}}$ for $p = 0$. Then, we have that $\mathcal{L}(\cdot)$ is concave for $p < 1$ and otherwise convex.*

Proof. First we prove the concavity of $\mathcal{L}(\cdot)$ when $p < 1$ and $p \neq 0$. It suffices to prove, for any $n \in \mathbb{N}$ and for any $\{\mathbf{a} = (a_1, a_2, \ldots, a_n) : a_i > 0, \quad i \in \mathbb{N}_n\}$, that function $(\sum_{j\in N_n} a_j^p)^{1/p}$ is concave w.r.t. variable $\mathbf{a}$. To this end, let f be a function defined, for any $x > 0$ and $y > 0$, by $f(x, y) = -x^{1-p}y^p/p$. We can easily prove that f is jointly convex w.r.t. (x, y), since its Hessian matrix

$$(1-p)\begin{pmatrix} x^{-p-1}y^p & -x^{-p}y^{p-1} \\ -x^{-p}y^{p-1} & x^{1-p}y^{p-2} \end{pmatrix} \in \mathbb{S}_+^d.$$

Consequently, for any $i \in \mathbb{N}_n$, $-x^{1-p}a_i^p/p$ is jointly convex, which implies that its summation $\sum_{i\in\mathbb{N}_n} -x^{1-p}a_i^p/p = -x^{1-p}(\sum_{i\in\mathbb{N}_n} a_i^p)/p$ is jointly convex. Hence, the function defined by $E(x, \mathbf{a}) = (1-p)x/p - x^{1-p}(\sum_{i\in\mathbb{N}_n} a_i^p)/p$ is also jointly convex w.r.t. $(x, \mathbf{a})$. Clearly,

$$-\big(\sum_{j\in N_n} a_j^p\big)^{1/p} = \min\{E(x, \mathbf{a}) : x \ge 0\}. \tag{5}$$

Recalling that the partial minimum of a jointly convex function is convex [9, Sec.IV.2.4], we obtain the concavity of $(\sum_{j\in N_n} a_j^p)^{1/p}$ when $p < 1$ and $p \neq 0$. The concavity of $\mathcal{L}$ for $p = 0$ follows from the fact that the limit function of a sequence of concave functions is concave.

The convexity of $\mathcal{L}$ for $p \geq 1$ can be proved similarly by observing that $E(x, \mathbf{a})$ is jointly concave if $p \geq 1$. Consequently, equation (5) should be replaced by $(\sum_{j\in N_n} a_j^p)^{1/p} = \min\{-E(x, \mathbf{a}) : x \geq 0\}$. This completes the proof of the theorem.

We conclude this section with two remarks. Firstly, we exclude the extreme case $p = 1$ since, in this case, the optimal solution of DML$_p$ will be always a rank-one matrix (i.e. the data is projected to the line), as argued in [25]. Secondly, when $p \in (1, \infty)$, by Theorem 1 we know that formulation (2) is indeed a problem of *maximizing a convex function*, which is a challenging task to get a global solution. In this paper we will only consider the case $p \in (-\infty, 1)$ which guarantees that formulation (2) is a convex optimization problem.

3 Equivalent Formulation and Optimization

We turn our attention to an equivalent formulation of problem (2), which is critical to designing its efficient algorithms. For notational simplicity, denote the *spectrahedron* by $\mathcal{P} = \{M \in \mathbb{S}_+^d : \mathbf{Tr}(M) = 1\}$ and let $X_\mathcal{S} = \sum_{(i,j)\in\mathcal{S}} X_{ij}$. Then, DML$_p$ (i.e. formulation (2)) can be rewritten as the following problem:

$$\begin{array}{cc} \max_{M\in\mathbb{S}_+^d} & \left[\sum_{\tau\in\mathcal{D}} \langle X_\tau, M\rangle^p / D\right]^{\frac{1}{p}} \\ \text{s.t.} & \langle X_\mathcal{S} + \delta \mathbf{I}_d, M\rangle \leq 1. \end{array} \tag{6}$$

Without loss of generality, we assume that $X_\mathcal{S}$ is invertible throughout the paper. This can be achieved by adding a small ridge term, i.e. $X_\mathcal{S} \longleftarrow X_\mathcal{S} + \delta\, \mathbf{I}_d$ where $\mathbf{I}_d$ is the identity matrix and $\delta > 0$ is a small ridge constant. In this case, we can apply the Cholesky decomposition to get that $X_\mathcal{S} = LL^\top$, where L is a lower triangular matrix with strictly positive diagonal entries.

Equipped with the above preparations, we are now ready to show that problem (2) is equivalent to an optimization problem over the spectrahedron $\mathcal{P} = \{M \in \mathbb{S}_+^d : \mathbf{Tr}(M) = 1\}$. Similar ideas have been used in [28].

Theorem 2. *For any $\tau = (i, j) \in \mathcal{D}$, let $\widetilde{X}_\tau = L^{-1}(x_i - x_j)(L^{-1}(x_i - x_j))^\top$. Then, problem (2) is equivalent to*

$$\max_{S\in\mathcal{P}} \Big[\sum_{\tau\in\mathcal{D}} \langle \widetilde{X}_\tau, S\rangle^p\Big]^{\frac{1}{p}}, \tag{7}$$

Proof. Let M^* be an optimal solution of problem (2) and $\widetilde{M}^* = \frac{M^*}{\langle X_\mathcal{S}, M^*\rangle}$. Then, $\langle X_\mathcal{S}, \widetilde{M}^*\rangle = 1$ and $\Big[\sum_{\tau\in\mathcal{D}} \frac{\langle X_\tau, \widetilde{M}^*\rangle^p}{D}\Big]^{\frac{1}{p}} = \Big[\sum_{\tau\in\mathcal{D}} \frac{\langle X_\tau, M^*\rangle^p}{D}\Big]^{\frac{1}{p}} / \langle X_\mathcal{S}, M^*\rangle \geq$

Table 1. Pseudo-code of the Frank-Wolfe algorithm to solve DML_p where f denotes the objective function of formulation (7)

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

$\left[\sum_{\tau \in \mathcal{D}} \frac{\langle X_\tau, M^* \rangle^p}{D}\right]^{\frac{1}{p}}$ since $\langle X_\mathcal{S}, M^* \rangle \leq 1$. This implies that $\widetilde{M}^*$ is also an optimal solution. Consequently, problem (2) is equivalent to, up to a scaling constant,

$$\begin{array}{ll} \max_{M \in \mathbb{S}_+^d} & \left[\sum_{(i,j) \in \mathcal{D}} \langle X_\tau, M \rangle^p / D\right]^{\frac{1}{p}} \\ \text{s.t.} & \langle X_\mathcal{S}, M \rangle = 1. \end{array} \tag{8}$$

Recall that $X_\mathcal{S} = LL^\top$ by Cholesky decomposition. Now the desired equivalence between (2) and (7) follows from changing variable $S = L^\top M L$ in (8). This completes the proof of the theorem.

By Theorem 2, the original metric learning problem (2) is reduced to a maximization problem on the spectrahedron. Therefore, we can apply the Frank-Wolfe (FW) algorithm [5,8] to obtain the optimal solution: the pseudo-code of the algorithm is given in Table 1 where f denotes the objective function of formulation (7). We conclude this section with a final remark. The objective function $\left[\sum_{\tau \in \mathcal{D}} \langle \widetilde{X}_\tau, S \rangle^p\right]^{\frac{1}{p}}$ in formulation (7) is not smooth since p can be negative. In order to avoid the numerical instability, we can add a small positive number inside so that it becomes a smooth function, i.e. $\left[\sum_{\tau \in \mathcal{D}} (\langle \widetilde{X}_\tau, S \rangle)^p\right]^{\frac{1}{p}}$ is replaced by $\left[\sum_{\tau \in \mathcal{D}} (\langle \widetilde{X}_\tau, S \rangle + \varepsilon)^p\right]^{\frac{1}{p}}$ where ε is a small positive number (e.g. $\varepsilon = 10^{-8}$). If the objective function has a Lipschitz-continuous gradient, then, by choosing $\alpha_t = \frac{2}{t+1}$, the FW algorithm is guaranteed to converge with a time complexity $\mathcal{O}(1/t)$. One can refer to [8,27] for a detailed proof.

4 Related Work

In recent years, distance metric learning has received a lot of attention in machine learning, see e.g. [1,2,4,6,15,19,20,22,25,27] and the references therein. It will be a difficult task to give a comprehensive review on related work. Below we only

briefly discuss some methods which are closely related to our work. We refer the readers to [26] for more related work on metric learning.

Xing et al. [25] presented metric-learning formulation (1) for k-means clustering. The method aims to maximize the distances between dissimilar samples subject to the constraint that distances between similar samples are upper-bounded. Ying et al. [28] proposed to maximize the minimal distance between dissimilar pairs while maintaining an upper bound for the distances between similar pairs. The proposed method (4) was shown to be equivalent to an eigenvalue optimization, which was solved by the Frank-Wolfe algorithm after smoothing the objective function. Our method DML_p is mainly motivated by the above two methods and provides a more general framework by recovering [25,28] as special cases. In contrast to the alternating projection method [25], we show that DML_p is reduced to a convex optimization problem over the spectrahedron. This new optimization formulation enables the direct application of the Frank-Wolfe algorithm which only needs the computation of the largest eigenvector of a matrix per iteration.

Weinberger et al. [22] developed the method called LMNN to learn a Mahalanobis distance metric in kNN classification settings. LMNN, as one of the state-of-the-art metric learning methods, aims to enforce k-nearest neighbors always belonging to the same class while examples from different classes being separated by a large margin. LMNN is a local method as it only used triplets from k-nearest neighbors. Similar to LMNN, our method focuses on similar pairs and dissimilar pairs generated from k-nearest neighbors. Davis et al. [4] proposed an information theoretic approach (ITML) to learning a Mahalanobis distance function by minimizing the Kullbach-Leibler divergence between two multivariate Gaussians subject to pairwise constraints.

Shen et al. [19] recently employed the exponential loss for metric learning named as BoostMetric and a boosting-based algorithm was developed. The rationale behind this algorithm is that each p.s.d. matrix can be decomposed into a linear positive combination of trace-one and rank-one matrices. This algorithm is very similar to the Frank-Wolfe algorithm employed for DML_p since both of them iteratively find a linear combination of rank-one matrices to approximate the desired solution. However, the method is a general column-generation algorithm and its convergence rate is not clear. The Frank-Wolfe algorithm for DML_p is theoretically guaranteed to have a convergence rate $\mathcal{O}(1/t)$ and it is relatively easy to be implemented by using just a few lines of MATLAB codes.

Guillaumin et al. [7] presented a metric learning model based on a logistic regression loss function called LDML. The method aims to learn robust distance measures for face identification using a logistic discriminant. In order to reduce the computational time, the authors proposed to remove the positive semi-definiteness constraint on the distance matrix. This would only lead to a sub-optimal solution.

Table 2. Description of datasets used in the experiments: n and d respectively denote the number of samples and attributes (feature elements) of the data; T is the number of triplets and D is the number of dissimilar pairs

Data	No.	n	d	class	T	D
Balance	1	625	4	3	3951	1317
Breast-Cancer	2	569	30	2	3591	1197
Diabetes	3	768	8	2	4842	1614
Image	4	2310	19	2	14553	4851
Iris	5	150	4	3	954	315
Waveform	6	5000	21	3	31509	10503
Wine	7	178	13	3	1134	378

5 Experiments

In this section, we compare the empirical performance of our proposed method DML_p with six other methods: the method proposed in [25] denoted by *Xing*, *LMNN* [22], *ITML* [4], *BoostMetric* [19], *DML-eig* [28] and the baseline algorithm using the standard Euclidean distance denoted by *Euclidean*. The model parameters in ITML, LMNN, BoostMetric and DML_p are tuned via three-fold cross validation. In addition, the maximum iteration number for DML_p is 1000 and the algorithm is terminated when the relative change of the objective function value is less than 10^{-5}.

We first run the experiments on UCI datasets to compare the kNN classification performance ($k = 3$) of different metric learning methods, where the kNN classifier is constructed using the Mahalanobis distance learned by metric learning methods. Then, we investigate the application of our method to the problem of face verification. In particular, we evaluate our new metric learning method using a large scale face database called Labeled Faces in the Wild (LFW) [10]. The LFW dataset is very 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. Recently it has become a benchmark to test new face verification algorithms [10,24,7,18].

5.1 Convergence and Generalization on UCI Datasets

To investigate the convergence and generalization of DML_p , we run experiments on seven UCI datasets: i.e. 1) Balance; 2) Breast-Cancer; 3) Diabetes; 4) Image segmentation; 5) Iris; 6) Waveform; 7) Wine. The statistics of the datasets are summarized in Table 2. All the experimental results are obtained by averaging over 10 runs and, for each run, the data is randomly split into 70% for training and 30% for testing. To generate relative constraints and pairwise constraints, we adopt a similar mechanism in [22]. More specifically, for each training point x_i, k nearest neighbors that have the same labels as y_i (targets) as well as k nearest neighbors that have different labels from y_i (imposers) are found. According

to x_i and its corresponding targets and imposers, we then construct the set of similar pairs $\mathcal{S}$, the set of dissimilar pairs $\mathcal{D}$ and the set of relative constraints in the form of triplets denoted by $\mathcal{T}$ required by LMNN and BoostMetric. As mentioned above, the original formulation in [25] used all pairwise constraints. For fairness of comparison, all methods including *Xing* used the same set of similar/dissimilar pairs generated locally as above.

Firstly, we study the convergence of algorithm DML_p with varying values of p. In Figure 1, we plot the objective function value of DML_p versus the number of iteration on Balance (subfigure (a)); Iris (subfigure (b)); Diabetes (subfigure (c)); and Image (subfigure (d)). We can see from Figure 1 that the algorithm converges quickly. The smaller the value of p is and the more iterations algorithm DML_p needs.

Secondly, we investigate the performance of DML_p against different values of p. Figure 2 depicts the test error of DML_p versus the value of p on Balance (subfigure (a)); Iris (subfigure (b)); Diabetes (subfigure (c)); and Image (subfigure (d)). We can observe from Figure 2 that the test error varies on different values of p and the best performance of DML_p is superior to those of DML-eig [28] and Xing [25] which are the special cases of DML_p with $p \to -\infty$ and $p = 1/2$ respectively. This observation validates the value of the general formulation DML_p and suggests the importance of choosing an appropriate value of p. In the following experiments, we will tune the value of p by three cross-validation.

Finally, we study the generalization performance of kNN classifiers where the distance metric to measure nearest neighbors is learned by metric learning methods. To this end, we compare DML_p with other metric learning methods including Xing [25], LMNN [22,23], ITML [4] and BoostMetric [19] as mentioned above. Figure 3 depicts the performance of different methods. It shows that almost all metric learning methods improve kNN classification using Euclidean distance on most datasets. Our proposed method DML_p delivers competitive performance with other state-of-the-art algorithms such as ITML, LMNN and BoostMetric. Indeed, DML_p outperforms other methods on 4 out of 7 datasets and shows competitive performance against the best one on the rest 3 datasets. From Figure 3, it is reasonable to see that the test errors of $\text{DML}_{1/2}$ are consistent with those of Xing since they are essentially the same model implemented by different algorithms. The only exception is the performance on Waveform dataset: the test error of Xing is much worse than $\text{DML}_{1/2}$. The reason could be that the alternating projection method proposed in [25] does not converge in a reasonable time due to the relatively large number of samples in Waveform dataset.

5.2 Application to Face Verification

The task of face verification is to determine whether two face images are from the same identity or not. Metric learning provides a very natural solution by comparing the image pairs based on the metric learnt from the face data. In this experiment, we investigate the performance of DML_p on the LFW dataset [10] – a benchmark dataset for face verification. It contains a total of 13233 labeled

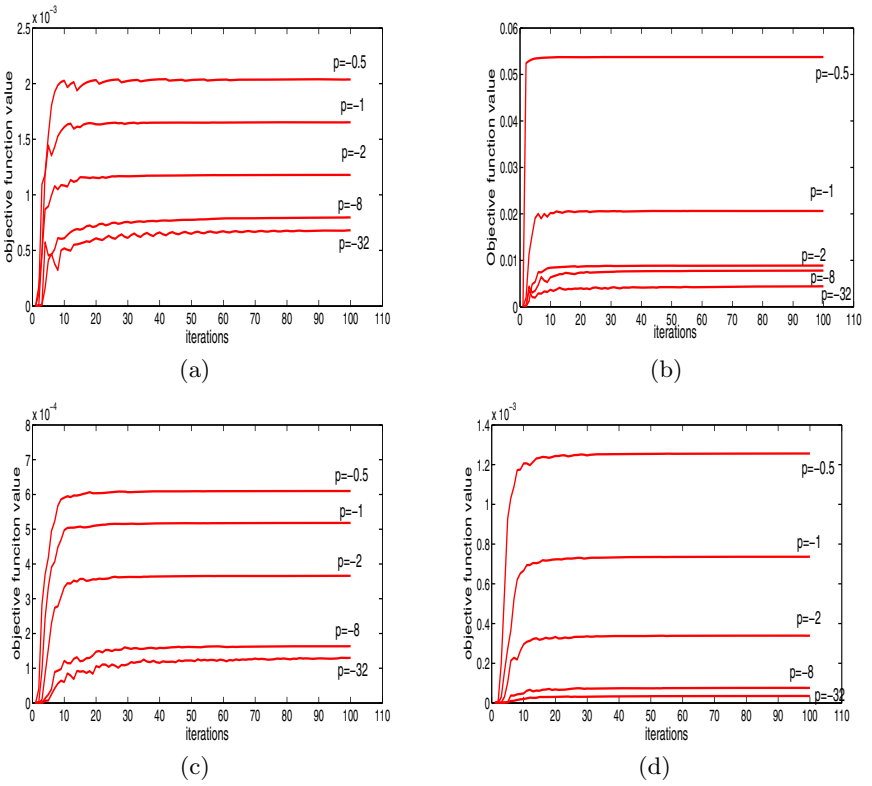

Fig. 1. Evolution of the objective function value of DML_p versus the number of iteration with varying p on Balance (a), Iris (b), Diabetes (c) and Image (d)

face images of 5749 people, 1680 of them appear in more than two images. There are two separate settings for forming training data: image-restricted and image-unrestricted setting. In the image-restricted paradigm, only the information whether a pair of images belongs to the same person (same class) is available and no information of actual names (class labels) in the pair of images is given. In the unrestricted setting, all available data including the identity of the people in the image is known. In this paper, we mainly consider the image-restricted setting.

The images we used are in gray scale and aligned in two ways. One is "funneled" [10] and the other is "aligned" using a commercial face alignment software [14]. We investigated several facial descriptors (features extracted from face images): 1) raw pixel data by concatenating the intensity value of each pixel in the image denoted by *Intensity*; 2) Local Binary Patterns (LBP) [13]; 3) Three-Patch Local Binary Patterns (TPLBP) [24]. For a fair comparison with [7], we

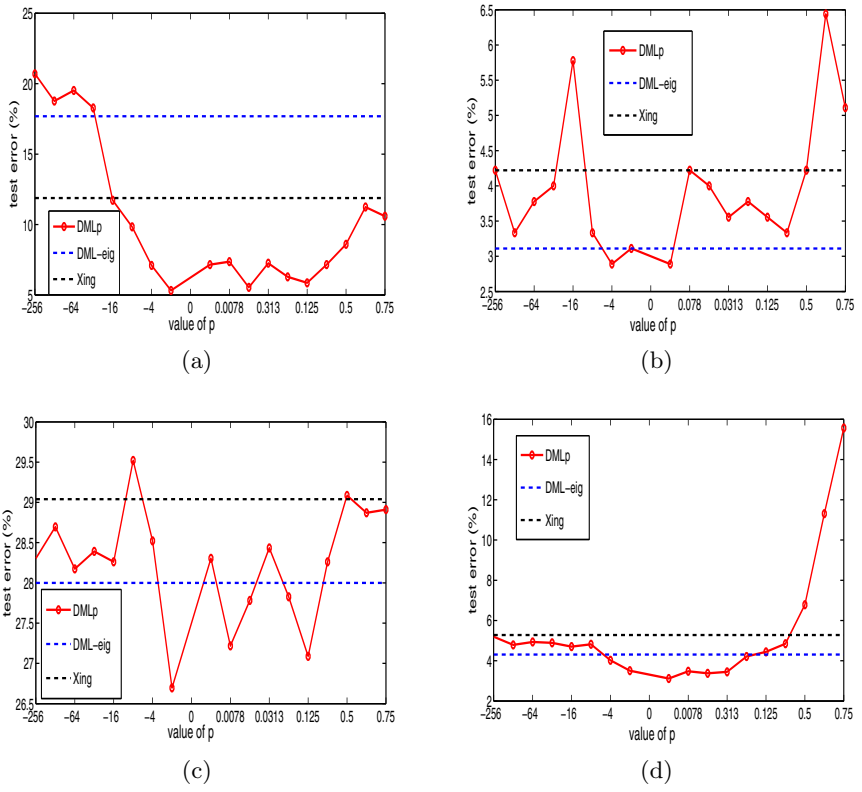

Fig. 2. Test error (%) of DML$_p$ versus different values of p on Balance (a), Iris (b), Diabetes (c) and Image (d). Red circled line is the result of DML$_p$ across different values of p (log-scaled); blue dashed line is the result of DML-eig and black dashed line represents the result of Xing.

also used SIFT descriptors [1]computed at the fixed facial key-points (e.g., corners of eyes and nose). Since the original dimensionality of the features is quite high (from 3456 to 12000), we reduced the dimension using principal component analysis (PCA). These descriptors were tested with both their original values and the square root of them [24,7].

In the image-restricted protocol, only pairwise constraints are given. LMNN and BoostMetric are not applicable to this setting since they require relative constraints in the form of triplets. Hence, we only compared our DML$_p$ method with ITML [4] and LDML [7]. The performance of our method is measured by the 10-fold cross-validation test. In each repeat, nine folds containing 2700 similar pairs of images and 2700 dissimilar pairs of images are used to learn

[1] `http://lear.inrialpes.fr/people/guillaumin/data.php`

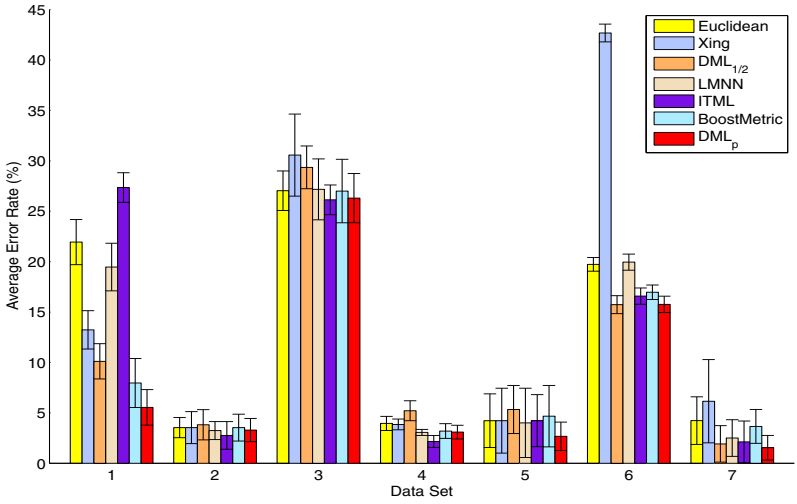

Fig. 3. Average test error (%) of DML_p against other methods

a metric and the remaining fold containing 600 image pairs is used to evaluate the performance of the metric learning method using accurate verification rate.

Firstly, we investigate the performance of DML_p on the SIFT descriptor by varying the dimension of principal components. Figure 4 depicts the verification accuracy versus the dimension of PCA. We can see that, compared to the ITML and LDML algorithms, our DML_p method using only SIFT descriptor delivers relatively stable performance as the PCA dimension varies. In particular, the performance of DML_p becomes stable after the dimension of PCA reaches around 100 and it consistently outperforms ITML across different PCA dimensions. We also observed similar results for other descriptors. Hence, for simplicity we set the PCA dimension to be 100 for the SIFT descriptor and other descriptors. According to [7], the best performances of LDML and ITML on the SIFT descriptor are 77.50% and 76.20% respectively. The best performance of DML_p reaches around 80% which outperforms ITML and LDML. We also note that the performance of ITML we got here is consistent with that reported in [7].

Secondly, we test the performance of our method using different descriptors and their combinations. Table 3 summarizes the results. In Table 3, the notation "Above combined" means that we combine the distance scores from the above listed (six) descriptors in the table using a linear Support Vector Machine (SVM), following the procedure in [7]. "All combined" means that all eight distance scores are combined. We observe that combining 4 descriptors (Intensity, SIFT, LBP and TFLBP) and their square-root ones yields 86.07% which outperforms

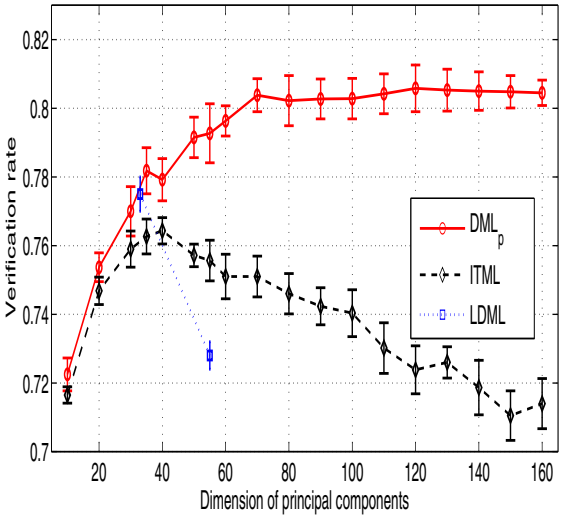

Fig. 4. Average verification rate of DML_p , ITML, and LDML on LFW by varying PCA dimension using the SIFT descriptor. The result of LDML is copied from Guillaumin et al. [7]: the best performance of LDML and ITML on the SIFT descriptor are respectively 77.50% and 76.20%.

Table 3. Performance of DML_p on LFW database with different descriptors (average verification accuracy and standard error). "DML_p SQRT" means DML_p uses the square root of the descriptor. "Intensity" means the raw pixel data by concatenating the intensity value of each pixel in the image. For all feature descriptors, the dimension is reduced to 100 using PCA. See more details in the text.

	DML_p	DML_p SQRT
SIFT	0.8015 ± 0.0055	0.8028 ± 0.0059
LBP	0.7972 ± 0.0062	0.8005 ± 0.0081
TPLBP	0.7790 ± 0.0058	0.7822 ± 0.0061
Above combined	0.8572 ± 0.0055	
Intensity	0.7335 ± 0.0054	0.7348 ± 0.0051
All combined	$\mathbf{0.8607 \pm 0.0058}$	

85.65% of DML-eig [28]. As mentioned above, DML-eig can be regarded as a limiting case of DML_p as $p \to -\infty$. This observation also validates the value of the general formulation DML_p . From Table 3, we can see that, although the individual performance of Intensity is inferior to those of other descriptors, combining it with other descriptors slightly increases the overall performance from 85.72% to 86.07%.

Finally, we summarize the performance of DML_p and other state-of-the-art methods in Table 4 and plot the ROC curve of our method compared to other

Table 4. Comparison of DML_p with 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

Method	Accuracy
High-Throughput Brain-Inspired Features, aligned [18]	$\mathbf{0.8813 \pm 0.0058}$
LDML + Combined, funneled [7]	0.7927 ± 0.0060
DML-eig + Combined [28]	0.8565 ± 0.0056
DML_p + Combined (this work)	0.8607 ± 0.0058

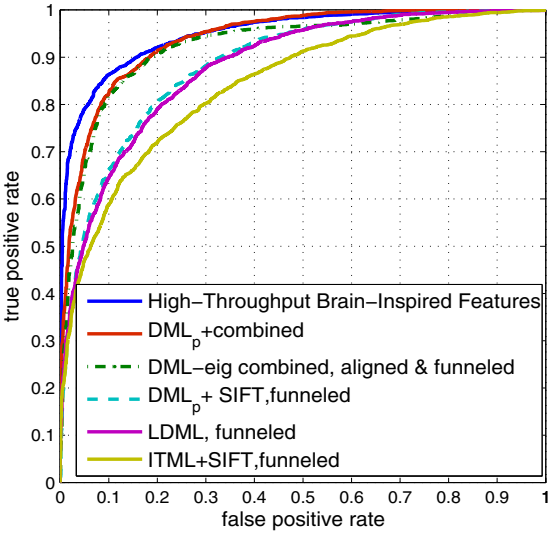

Fig. 5. ROC curves of DML_p and other state-of-the-art methods on LFW dataset

published results in Figure 5. We observe from Table 4 that our method DML_p outperforms LDML [7] and slightly improves the result of DML-eig [28]. The best performance on the restricted setting to date is 88.13% [18]. Note that the results compared here are system to system where metric learning is only one part of the system. We should also point out the result in [18] was not achieved by metric learning method. Instead, it performs sophisticated large scale feature search which used multiple complimentary representations derived through training set augmentation, alternative face comparison functions, and feature set searches with a varying number of model layers. We believe that the performance of DML_p may be further improved by exploring different types of descriptors such as those used in [18].

6 Conclusion

In this paper we extended and developed the metric learning models proposed in [25,28]. In particular, we proposed a general and unified framework which recovers the models in [25,28] as special cases. This novel framework was shown to be equivalent to a semi-definite program over the spectrahedron. This equivalence is important since it enables us to directly apply the Frank-Wolfe algorithm (e.g. [5,8]) to obtain the optimal solution. Experiments on UCI datasets validate the effectiveness of our proposed method and algorithm. In addition, the proposed method performs well on the Labeled Faces in the Wild (LFW) dataset in the task of face verification.

We now discuss some possible future work. It would be interesting to investigate the kernelised version of DML_p using similar ideas from [11,15]. Metric learning can be also regarded as a dimension reduction method. However, in its application to face verification, a common approach is to use PCA to reduce the dimensionality of the original descriptor. This triggers a natural question for future work on how to design effective metric learning methods to directly deal with the original descriptors of the images.

Acknowledgements. This work is supported by the EPSRC under grant EP/J001384/1. The corresponding author is Yiming Ying.

References

1. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a mahalanobis metric from equivalence constraints. Journal of Machine Learning Research 6, 937–965 (2005)
2. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively with application to face verification. In: CVPR (2005)
3. Cox, T., Cox, M.: Multidimensional scaling. Chapman and Hall, London (1994)
4. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)
5. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quaterly 3, 149–154 (1956)
6. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood component analysis. In: NIPS (2004)
7. Guillaumin, M., Verbeek, J., Schmid, C.: Is that you? Metric learning approaches for face identification. In: ICCV (2009)
8. Hazan, E.: Sparse Approximate Solutions to Semidefinite Programs. In: Laber, E.S., Bornstein, C., Nogueira, L.T., Faria, L. (eds.) LATIN 2008. LNCS, vol. 4957, pp. 306–316. Springer, Heidelberg (2008)
9. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press (1991)
10. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49 (2007)

11. Jain, P., Kulis, B., Dhillon, I.S.: Inductive regularized learning of kernel functions. In: NIPS (2010)
12. Jin, R., Wang, S., Zhou, Y.: Regularized distance metric learning: theory and algorithm. In: NIPS (2009)
13. Ojala, T., Pietikainen, M., Maenpaa, T.: 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)
14. Taigman, Y., Wolf, L., Hassner, T.: Multiple one-shots for utilizing class label information. In: The British Machine Vision Conference (2009)
15. Tsang, I.W., Kwok, J.T.: Distance Metric Learning with Kernels. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714. Springer, Heidelberg (2003)
16. Tenenbaum, J., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
17. Roweis, S.T., Lawrance, K.S.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
18. Pinto, N., Cox, D.: Beyond simple features: a large-scale feature search approach to unconstrained face recognition. In: International Conference on Automatic Face and Gesture Recognition (2011)
19. Shen, C., Kim, J., Wang, L., Hengel, A.: Positive semidefinite metric learning with boosting. In: NIPS (2009)
20. Torresani, L., Lee, K.: Large margin component analysis. In: NIPS (2007)
21. Vandenbergheand, L., Boyd, S.: Semidefinite programming. SIAM Review 38(1), 49–95 (1996)
22. Weinberger, K.Q., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbour classification. In: NIPS (2006)
23. Weinberger, K.Q., Saul, L.K.: Fast solvers and efficient implementations for distance metric learning. In: ICML (2008)
24. Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: Workshop on Faces Real-Life Images at ECCV (2008)
25. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side information. In: NIPS (2002)
26. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University (2007)
27. Ying, Y., Huang, K., Campbell, C.: Sparse metric learning via smooth optimization. In: NIPS (2009)
28. Ying, Y., Li, P.: Distance metric learning with eigenvalue optimization. Journal of Machine Learning Research 13, 1–26 (2012)