A Scalable Dual Approach to Semidefinite Metric Learning

Chunhua Shen[1,2] Junae Kim[2,3] Lei Wang[3,4]
[1]University of Adelaide [2]NICTA [3]Australian National University [4]University of Wollongong

Abstract

Distance metric learning plays an important role in many vision problems. Previous work of quadratic Mahalanobis metric learning usually needs to solve a semidefinite programming (SDP) problem. A standard interior-point SDP solver has a complexity of $O(D^{6.5})$ (with D the dimension of input data), and can only solve problems up to a few thousand variables. Since the number of variables is $D(D+1)/2$, this corresponds to a limit around $D < 100$. This high complexity hampers the application of metric learning to high-dimensional problems. In this work, we propose a very efficient approach to this metric learning problem. We formulate a Lagrange dual approach which is much simpler to optimize, and we can solve much larger Mahalanobis metric learning problems. Roughly, the proposed approach has a time complexity of $O(t \cdot D^3)$ with $t \approx 20 \sim 30$ for most problems in our experiments. The proposed algorithm is scalable and easy to implement. Experiments on various datasets show its similar accuracy compared with state-of-the-art. We also demonstrate that this idea may also be able to be applied to other SDP problems such as maximum variance unfolding.

1. Introduction

Distance metric learning is an important problem in computer vision and machine learning. To find an appropriate data-dependent distance metric is a compelling goal in image classification and recognition. Many classic algorithms such as k-nearest neighbor (kNN) or k-means clustering heavily rely on the employed distance metric. Large-margin metric learning focuses on finding a metric such that the same class's data points are close to each other and those in different classes are separated by a large margin. Weinberger *et al*.'s large margin nearest neighbor (LMNN) is a seminal work that belongs to this category [20]. Given the input data $\boldsymbol{a} \in \mathbb{R}^D$, we want to learn a linear transformation such that the projected data $\mathbf{L}\boldsymbol{a} \in \mathbb{R}^d$ can be measured by the traditional Euclidean distance. In order to obtain a convex problem, instead of learning the projection matrix ($\mathbf{L} \in \mathbb{R}^{D\times d}$), one usually optimizes the quadratic product of the projection matrix ($\mathbf{X} = \mathbf{L}\mathbf{L}^\top$) [24, 20]. This linearization *convexifies* the original non-convex problem. The projection matrix may then be recovered by an eigen-decomposition of $\mathbf{X}$. The price is that one needs to solve a semidefinite programming (SDP) problem since $\mathbf{X}$ must be positive semidefinite (p.s.d.). Conventional interior-point SDP solvers have a high computation complexity of $O(D^{6.5})$, with D the dimension of input data. This high complexity hampers the application of metric learning to high-dimensional problems. Here we follow this line to propose a new formulation of quadratic Mahalanobis metric learning using proximity comparison information. The main contribution is that, with the proposed formulation, we can very efficiently solve the SDP problem in the dual. Because of strong duality holds, we can then recover the primal variable $\mathbf{X}$ from the dual solution. The computation complexity of the optimization is dominated by eigen-decomposition $O(D^3)$ and thus the overall complexity is $O(t \cdot D^3)$. Here t is the number of iterations for convergence and typically $t \approx 20$. Therefore, the main contribution of this work is as follows.

1. We propose a novel formulation of metric learning, which is much more scalable than previous approaches by solving its Lagrange dual problem. Now we are able to apply metric learning to high-dimensional data. We refer our metric learning algorithm as to FrobMetric—the core that makes it efficient and scalable is the Frobenius norm regularization.
2. We also generalize the method to any Frobenius norm regularized SDP problems. As an example, we use the proposed algorithm to approximately solve the Frobenius norm perturbed maximum variance unfolding (MVU) problem [21]. We demonstrate that the proposed method is much more efficient than the original MVU implementation on a few data sets and plausible embedding can be obtained with our approach.

   The proposed method can also be applied to other SDP problems of interest in vision applications.

Before we present our algorithm, we briefly review relevant work. For large-margin metric learning, Xing *et al*. [24] proposed a global distance metric learning approach

using a convex optimization method. Although their experiments show improved performance on clustering problems, it does not work so well on classification problems. Goldberger *et al*. [6] showed that neighborhood component analysis (NCA) may outperform traditional dimensionality reduction methods. NCA learns the projection matrix directly and its objective function is non-convex. For high-dimensional problems, NCA is prone to become trapped in a local optimum. Davis *et al*. [5] proposed an information theoretic metric learning (ITML) approach to learn a Mahalanobis metric. Relevant component analysis (RCA) [17] is an unsupervised metric learning method. RCA does not maximize the distance of different classes, but minimizes the distance between data in Chunklets. Chunklets consist of data that come from the same (although unknown) class. The closest work to ours may be LMNN [20] and BoostMetric [16]. The key idea of LMNN is to maximize the margin between different classes, at the same time minimize the intra-class distance. The optimization is an SDP problem. In order to improve the scalability of the algorithm, instead of using standard SDP solvers, Weinberger *et al*. [20] have proposed an alternate projection method. At each iteration, the learned metric $\mathbf{X}$ is projected back to the semidefinite cone using eigen-decomposition, in order to preserve the semi-definiteness of $\mathbf{X}$. In this sense, at each iteration, the computation complexity of their algorithm is similar to ours. However, the alternate method needs an extremely large number of iterations to converge (defaulted to $10,000$ in the author's implementation). In contrast, our algorithm solves the Lagrange dual problem and needs only $20 \sim 30$ iterations in most cases. Besides, our proposed algorithm is much easier to implement.

Recently, Shen *et al*. [16] introduced BoostMetric by adapting the boosting technique to distance metric learning. Based on an important theorem that a positive semidefinite (p.s.d.) matrix with trace of one can always be represented as a convex combination of multiple rank-one matrices, they have generalized AdaBoost in the sense that the weak learner of BoostMetric is a matrix, rather than a classifier. Our approach, FrobMetric, is inspired by BoostMetric in the sense that both algorithms use proximity comparison information among triplets for training. However, their objective functions are different, which leads to completely different optimization strategies. BoostMetric computes the Mahalanobis distance metric using rank-one update at each iteration. Although at each iteration, BoostMetric only needs to compute the leading eigenvector, it needs a large number of iterations for convergence. Moreover, BoostMetric requires a differentiable loss function. Instead our algorithm optimizes a non-differentiable hinge loss.

In summary, we propose a simple, efficient and scalable optimization method for quadratic Mahalanobis metric learning. The formulated optimization problem is convex, therefore the global optimum can be attained in polynomial time. Moreover, by working with the Lagrange dual problem, we are able to use off-the-shelf eigen-decomposition and gradient descent methods like L-BFGS-B to solve the problem.

Let us define some notation first. A column vector is denoted by a bold lower-case letter ($\boldsymbol{x}$) and a matrix is denoted by a bold upper-case letter ($\mathbf{X}$). A positive semidefinite (p.s.d.) matrix is denoted as $\mathbf{A} \succcurlyeq 0$. With two vectors, $\boldsymbol{a} \succcurlyeq \boldsymbol{b}$ denotes the component-wise inequality. For matrices, we consider the vector space $\mathbb{R}^{m\times n}$ of real matrices of size $m \times n$. Let us denote the space of real matrices as $\mathbb{S}$. Similarly, The space of symmetric matrices of size $n \times n$ is $\mathbb{S}^n$, and the space of symmetric positive semidefinite matrices of size $n \times n$ is denoted as $\mathbb{S}^n_+$. The inner product defined on these spaces are $\langle \mathbf{A}, \mathbf{B}\rangle = \mathbf{Tr}(\mathbf{A}^\top \mathbf{B})$. Here $\mathbf{Tr}(\cdot)$ calculates the trace of a matrix. $\mathbf{diag}(\cdot)$ extracts the diagonal elements of a square matrix. Given a symmetric matrix $\mathbf{X}$ and its eigen-decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$ ($\mathbf{U}$ is an orthonormal matrix, and $\mathbf{\Sigma}$ is real and diagonal), we define the positive part of $\mathbf{X}$ as

$$(\mathbf{X})_+ = \mathbf{U}[\max(\mathbf{diag}(\mathbf{\Sigma}), 0)]\mathbf{U}^\top,$$

and the negative part of $\mathbf{X}$ as

$$(\mathbf{X})_- = \mathbf{U}[\min(\mathbf{diag}(\mathbf{\Sigma}), 0)]\mathbf{U}^\top.$$

Clearly $\mathbf{X} = (\mathbf{X})_+ + (\mathbf{X})_-$ holds.

2. Large-margin Distance Metric Learning

We briefly review the idea of learning a quadratic Mahalanobis distance metric. Suppose that we have a set of triplets $\mathbb{S} = \{(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k)\}$, which encodes proximity comparison information. $\mathbf{dist}_{ij}$ computes the Mahalanobis distance between $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ under a proper Mahalanobis matrix. That is, $\mathbf{dist}_{ij} = \|\boldsymbol{a}_i - \boldsymbol{a}_j\|^2_{\mathbf{X}} = (\boldsymbol{a}_i - \boldsymbol{a}_j)^\top \mathbf{X}(\boldsymbol{a}_i - \boldsymbol{a}_j)$. Here $\mathbf{X} \in \mathbb{R}^{D\times D}$, is positive semidefinite. It is equivalent to learn a projection matrix $\mathbf{L}$ such that $\mathbf{X} = \mathbf{L}\mathbf{L}^\top$.

Let us define the margin associated with a training triplets as $\rho_r = (\boldsymbol{a}_i - \boldsymbol{a}_k)^\top \mathbf{X}(\boldsymbol{a}_i - \boldsymbol{a}_k) - (\boldsymbol{a}_i - \boldsymbol{a}_j)^\top \mathbf{X}(\boldsymbol{a}_i - \boldsymbol{a}_j) = \langle \mathbf{A}_r, \mathbf{X}\rangle$, with $\mathbf{A}_r = (\boldsymbol{a}_i - \boldsymbol{a}_k)(\boldsymbol{a}_i - \boldsymbol{a}_k)^\top - (\boldsymbol{a}_i - \boldsymbol{a}_j)(\boldsymbol{a}_i - \boldsymbol{a}_j)^\top$. r indexes the set of training triplets in $\mathbb{S}$. As will be shown in the experiments, this type of proximity comparison information among triplets may be easier to be obtained for some applications like image retrieval. Here the metric learning procedure solely relies on the matrices $\mathbf{A}_r$ ($r = 1, \ldots, m$).

Putting it into the large-margin learning framework, the optimization problem is to maximize the margin with a regularization term that avoids over-fitting (or makes the prob-

lem well-posed):

$$\begin{aligned}\max_{\mathbf{X},\rho,\boldsymbol{\xi}}\ & \rho - \tfrac{C_1}{m}\textstyle\sum_{r=1}^{m}\xi_r \\ \text{s.t. } & \langle \mathbf{A}_r, \mathbf{X}\rangle \geq \rho - \xi_r, r = 1, \cdots, m, \\ & \boldsymbol{\xi} \succcurlyeq 0, \mathbf{Tr}(\mathbf{X}) = 1, \mathbf{X} \succcurlyeq 0.\end{aligned} \tag{P1}$$

Here $\mathbf{Tr}(\mathbf{X}) = 1$ removes the scale ambiguity of $\mathbf{X}$. This is the formulation mentioned in BoostMetric [16]. We can write the above problem into an equivalent form:

$$\begin{aligned}\min_{\mathbf{X},\boldsymbol{\xi}}\ & \mathbf{Tr}(\mathbf{X}) + \tfrac{C_2}{m}\textstyle\sum_{r=1}^{m}\xi_r \\ \text{s.t. } & \langle \mathbf{A}_r, \mathbf{X}\rangle \geq 1 - \xi_r, r = 1, \cdots, m, \\ & \boldsymbol{\xi} \succcurlyeq 0, \mathbf{X} \succcurlyeq 0.\end{aligned} \tag{P2}$$

These formulations are exactly equivalent given the appropriate choice of the trade-off parameters C_1 and C_2. The proof is similar to the one in [15]. Due to lack of space, here we only state the theorem.

Theorem 1. *A solution of* (P1)*,* $\mathbf{X}^\star$*, is also a solution of* (P2) *and vice versa up to a scale.*

More precisely, if (P1) *with parameter* C_1 *has a solution* $(\mathbf{X}^\star, \boldsymbol{\xi}^\star, \rho^\star > 0)$*, then* $(\frac{\mathbf{X}^\star}{\rho^\star}, \frac{\boldsymbol{\xi}^\star}{\rho^\star})$ *is the solution of* (P2) *with parameter* $C_2 = C_1/\mathrm{Opt}$(P1)*. Here* Opt(P1) *is the optimal objective value of* (P1)*.*

Both problems can be written into the format of standard SDP since the objective function is linear and a p.s.d. constraint is involved. Here we are interested in a Frobenius norm regularization instead of trace norm regularization. The key observation is that *the Frobenius norm regularization term can lead to scalable and simple optimization*. So we replace the trace norm in (P2) with the Frobenius norm, and we have:

$$\begin{aligned}\min_{\mathbf{X},\xi}\ & \tfrac{1}{2}\left\|\mathbf{X}\right\|_{\mathrm{F}}^2 + \tfrac{C_3}{m}\textstyle\sum_{r=1}^{m}\xi_r \\ \text{s.t. } & \langle \mathbf{A}_r, \mathbf{X}\rangle \geq 1 - \xi_r, r = 1, \cdots, m, \\ & \boldsymbol{\xi} \succcurlyeq 0, \mathbf{X} \succcurlyeq 0.\end{aligned} \tag{P3}$$

Again, it is straightforward to prove that (P3) and (P2) are equivalent given the appropriate choice of C_2 and C_3.

One may convert (P3) into a standard SDP and then use any off-the-shelf SDP solvers to solve the primal problem directly. However, as mentioned early, the computation complexity is very high and only small-scale problems can be solved in a reasonable CPU time. Next, we show that, the Lagrange dual problem of (P3) has some desirable properties.

We introduce the Lagrangian dual multipliers $\boldsymbol{u}$ to associate with the constraints in the primal and the symmetric matrix $\mathbf{Z}$ to associate with the p.s.d. constraint $\mathbf{X} \succcurlyeq 0$. The Lagrangian of (P3) then writes

$$\begin{aligned}\ell(\underbrace{\mathbf{X}, \boldsymbol{\xi}}_{\text{primal}}, \underbrace{\mathbf{Z}, \boldsymbol{u}}_{\text{dual}}) = {} & \tfrac{1}{2}\|\mathbf{X}\|_{\mathrm{F}}^2 + \tfrac{C_3}{m}\textstyle\sum_{r=1}^{m}\xi_r - \sum_r u_r \langle \mathbf{A}_r, \mathbf{X}\rangle \\ & + \textstyle\sum_r u_r - \sum_r u_r\xi_r - \boldsymbol{p}^\top\boldsymbol{\xi} - \langle \mathbf{X}, \mathbf{Z}\rangle\end{aligned}$$

with $\boldsymbol{u} \succcurlyeq 0$ and $\mathbf{Z} \succcurlyeq 0$. We need to minimize the Lagrangian over $\mathbf{X}$ and $\boldsymbol{\xi}$, which can be done by setting the first derivative to zero and we have

$$\mathbf{X}^\star = \mathbf{Z}^\star + \textstyle\sum_r u_r^\star \mathbf{A}_r, \tag{1}$$

and $\frac{C_3}{m} \succcurlyeq \boldsymbol{u} \succcurlyeq 0$. Substituting the expression for $\mathbf{X}$ back into the Lagrangian and we obtain the dual formulation:

$$\begin{aligned}\max_{\mathbf{Z},\boldsymbol{u}}\ & \textstyle\sum_{r=1}^{m} u_r - \frac{1}{2}\left\|\mathbf{Z} + \sum_{r=1}^{m} u_r\mathbf{A}_r\right\|_{\mathrm{F}}^2 \\ \text{s.t. } & \tfrac{C_3}{m} \succcurlyeq \boldsymbol{u} \succcurlyeq 0, \mathbf{Z} \succcurlyeq 0.\end{aligned} \tag{D3}$$

At the first glance, this dual problem still has a p.s.d. constraint and it is not clear how we can solve it more efficiently than to use standard interior-point methods. Because both the primal and dual problems are convex and under mild conditions, the Slater's condition holds, strong duality between (P3) and (D3) holds [3]. It means that the objective values of these two problem coincide at optimality and we may be able to indirectly solve the primal by solving the dual or the other way around. The KKT condition (1) enables us to recover $\mathbf{X}^\star$, which is the primal variable of interest, from the dual solution.

Given a fixed $\boldsymbol{u}$, the dual problem (D3) is simplified into

$$\min_{\mathbf{Z}} \left\|\mathbf{Z} + \textstyle\sum_{r=1}^{m} u_r\mathbf{A}_r\right\|_{\mathrm{F}}^2, \ \text{s.t. } \mathbf{Z} \succcurlyeq 0. \tag{2}$$

To simplify the notation, we define a symbol

$$\hat{\mathbf{A}} = -\textstyle\sum_{r=1}^{m} u_r\mathbf{A}_r.$$

So $\hat{\mathbf{A}}$ is a function of $\boldsymbol{u}$. (2) is to find a p.s.d. matrix $\mathbf{Z}$ such that $\|\mathbf{Z} - \hat{\mathbf{A}}\|_{\mathrm{F}}^2$ is minimized. Problem 2 has a closed-form solution, which is the positive part of $\hat{\mathbf{A}}$:

$$\mathbf{Z}^\star = (\hat{\mathbf{A}})_+. \tag{3}$$

Now the original dual problem can be simplified into

$$\max_{\boldsymbol{u}} \textstyle\sum_{r=1}^{m} u_r - \frac{1}{2}\left\|(\hat{\mathbf{A}})_-\right\|_{\mathrm{F}}^2, \ \text{s.t. } \tfrac{C_3}{m} \succcurlyeq \boldsymbol{u} \succcurlyeq 0. \tag{4}$$

The KKT condition is simplified into

$$\mathbf{X}^\star = (\hat{\mathbf{A}})_+ - \hat{\mathbf{A}} = -(\hat{\mathbf{A}})_-. \tag{5}$$

From the definition of the operator $(\cdot)_-$, $\mathbf{X}^\star$ computed by (5) must be p.s.d. It is nice that the simplified dual problem has no matrix variables. It only has simple box constraints on $\boldsymbol{u}$. The following theorem allows us to optimize for $\boldsymbol{u}$ in (4) using gradient descent methods.

Theorem 2. *The objective function of* (4) *is differentiable (but not necessarily twice differentiable).*

The proof can be easily obtained by using the results in Sect. 5.2 of [2]. So we can use sophisticated off-the-shelf first-order Newton algorithm such as L-BFGS-B [11] to optimize (4). In summary, the optimization procedure is as follows.

1. Input triplets and calculate $\mathbf{A}_r, r = 1 \dots m$.
2. Calculate the gradient of the objective function in (4), and use L-BFGS-B to optimize (4).
3. Calculate $\hat{\mathbf{A}}$ using the output of L-BFGS-B (namely, $\boldsymbol{u}^\star$) and compute $\mathbf{X}^\star$ from (5) using eigen-decomposition.

To implement this approach, one only needs to implement the callback function of L-BFGS-B, which computes the gradient of the objective function of (4). Note that other gradient methods like conjugate gradients may be preferred when the number of constraints (i.e., the size of training triplet set,m) is large. The gradient of dual problem (4) can be calculated as

$$g(u_r) = 1 + \left\langle (\hat{\mathbf{A}})_-, \mathbf{A}_r \right\rangle, r = 1, \dots, m.$$

So at each iteration, the computation of $(\hat{\mathbf{A}})_-$, which needs full eigen-decomposition, only needs to be calculated once to evaluate all the gradients, as well as the function value. When the number of constraints is not far more than the dimensionality of the data, eigen-decomposition dominates the computation complexity at each iteration. In this case, the overall complexity is $O(t \cdot D^3)$ with t being around $20 \sim 30$.

3. General Frobenius Norm SDP

In this section, we generalize the proposed idea to a broader setting. The general formulation of an SDP problem writes:

$$\min_{\mathbf{X}} \ \langle \mathbf{C}, \mathbf{X} \rangle \,, \ \text{s.t.} \ \mathbf{X} \succcurlyeq 0, \langle \mathbf{A}_i, \mathbf{X} \rangle \leq b_i, i = 1 \dots m.$$

We consider its Frobenius norm regularized version:

$$\min_{\mathbf{X}} \ \langle \mathbf{C}, \mathbf{X} \rangle + \tfrac{1}{2\sigma} \left\| \mathbf{X} \right\|_{\mathrm{F}}^2, \ \text{s.t.} \ \mathbf{X} \succcurlyeq 0, \langle \mathbf{A}_i, \mathbf{X} \rangle \leq b_i, \forall i.$$

Here σ is a regularized constant. We start by deriving the Lagrange dual of this Frobenius norm regularized SDP. The dual problem is,

$$\min_{\mathbf{Z}, \boldsymbol{u}} \ \tfrac{1}{2}\sigma \left\| \mathbf{Z} - \mathbf{C} - \hat{\mathbf{A}} \right\|_{\mathrm{F}}^2 + \boldsymbol{b}^\top \boldsymbol{u}, \ \text{s.t.} \ \mathbf{Z} \succcurlyeq 0, \boldsymbol{u} \succcurlyeq 0. \quad (6)$$

The KKT condition is

$$\mathbf{X}^\star = \sigma(\mathbf{Z}^\star - \hat{\mathbf{A}} - \mathbf{C}), \quad (7)$$

where we have introduced the notation $\hat{\mathbf{A}} = \sum_{i=1}^m u_i \mathbf{A}_i$. Keep it in mind that $\hat{\mathbf{A}}$ is a function of the dual variable $\boldsymbol{u}$. As in the case of metric learning, the important observation is that $\mathbf{Z}$ has an analytical solution when $\boldsymbol{u}$ is fixed:

$$\mathbf{Z} = (\mathbf{C} + \hat{\mathbf{A}})_+. \quad (8)$$

Therefore we can simplify (6) into

$$\min_{\boldsymbol{u}} \ \tfrac{1}{2}\sigma \left\| (\mathbf{C} + \hat{\mathbf{A}})_- \right\|_{\mathrm{F}}^2 + \boldsymbol{b}^\top \boldsymbol{u}, \ \text{s.t.} \ \boldsymbol{u} \succcurlyeq 0. \quad (9)$$

So now we can efficiently solve the dual problem using gradient descent methods. The gradient of the dual function is

$$g(u_i) = \sigma \left\langle (\mathbf{C} + \hat{\mathbf{A}})_-, \mathbf{A}_i \right\rangle + b_i, \forall i = 1 \dots m.$$

At optimality, we have $\mathbf{X}^\star = -\sigma(\mathbf{C} + \hat{\mathbf{A}}^\star)_-$.

The core idea of the proposed method here may be applied to an SDP which has a term in the format of Frobenius norm, either in the objective function or in the constraints.

4. Experiments

We first run metric learning experiments on UCI benchmark data and face recognition data. We then approximately solve the MVU problem [21] using the proposed general Frobenius norm SDP approach.

4.1. Distance metric learning

4.1.1 UCI benchmark test

We perform a comparison between the proposed FrobMetric and a few distance metric learning methods that represent the state-of-the-art such as RCA [17], NCA [6], LMNN [20], BoostMetric [16] and ITML [5] on data sets from the UCI Repository. We have included principal component analysis (PCA) and linear discriminant analysis (LDA) as the baseline approaches. Same as in [20], on some data sets (MNIST, Yale faces and USPS), we have applied PCA to reduce the original dimensionality and also to remove noises.

For all experiments, the task is to classify unseen instances in a testing subset. To accumulate statistics, the data are randomly split into 10 pairs of training/validating/testing subsets except MNIST and Letter, which are already divided into subsets. We tuned the regularization parameter in the compared methods using cross-validation. In this experiment, about 15% of data are used for cross-validation and 15% for testing.

For our FrobMetric and BoostMetric in [16], we use 3-nearest neighbors to generate triplets and check the performance using 3NN. For each training sample $\boldsymbol{a}_i$, its 3 nearest neighbors sharing the same class label as $\boldsymbol{a}_i$ and the 3

[1] When LMNN solves the global optimum on "Wine" set, the error rate is 20.77% (14.18%).

Table 1: Test errors of different metric learning methods on some UCI data sets with 3-NN. NCA [6] does not output a result on those larger data sets due to memory problems. Standard deviation is also reported for data sets having multiple runs.

	MNIST	USPS	letters	Yale faces	Bal	Wine	Iris
# samples	70,000	11,000	20,000	2,414	625	178	150
# triplets	450,000	69,300	94,500	15,210	3,942	1,125	945
dimension	784	256	16	1,024	4	13	4
dimension after PCA	164	60		300			
# training	50,000	7,700	10,500	1,690	438	125	105
# validation	10,000	1,650	4,500	362	94	27	23
# test	10,000	1,650	5,000	362	93	26	22
# classes	10	10	26	38	3	3	3
# runs	1	10	1	10	10	10	10
Error Rates %							
Euclidean	3.19	4.78 (0.40)	5.42	28.07 (2.07)	18.60 (3.96)	28.08 (7.49)	3.64 (4.18)
PCA	3.10	3.49 (0.62)	-	28.65 (2.18)	-	-	-
LDA	8.76	6.96 (0.68)	4.44	**5.08 (1.15)**	12.58 (2.38)	0.77 (1.62)	**3.18 (3.07)**
RCA [17]	7.85	5.35 (0.52)	4.64	7.65 (1.08)	17.42 (3.58)	**0.38 (1.22)**	**3.18 (3.07)**
NCA [6]	-	-	-	-	18.28 (3.58)	28.08 (7.49)	**3.18 (3.74)**
LMNN [20]	**2.30**	3.49 (0.62)	3.82	14.75 (12.11)	12.04 (5.59)	3.46 (3.82)[1]	3.64 (2.87)
ITML [5]	2.80	3.85 (1.13)	7.20	19.39 (2.11)	10.11 (4.06)	28.46 (8.35)	3.64 (3.59)
BoostMetric [16]	2.76	2.53 (0.47)	3.06	6.91 (1.90)	10.11 (3.45)	3.08 (3.53)	**3.18 (3.74)**
FrobMetric (this work)	2.56	**2.32 (0.31)**	**2.72**	9.20 (1.06)	**9.68 (3.21)**	3.85 (4.44)	3.64 (3.59)
Computational Time							
LMNN	11h	20s	1249s	896s	5s	2s	2s
ITML	1479s	72s	55s	5970s	8s	4s	4s
BoostMetric	9.5h	338s	**3s**	572s	**less than 1s**	2s	**less than 1s**
FrobMetric	**280s**	**9s**	13s	**335s**	**less than 1s**	**less than 1s**	**less than 1s**

nearest neighbors that have a different label are used. With 3 nearest neighbors information, the number of triplets of each data set for FrobMetric and BoostMetric are shown in Table 1. FrobMetric and BoostMetric have used exactly the same training information. Note that other methods do not use triplets as training data. The error rates based on 3NN and computational time for each learning metric are shown as well.

Experiment settings for LMNN and ITML follow the original work [20] and [5], respectively. The identity matrix is used for ITML's initial metric matrix. For NCA, RCA, LMNN, ITML and BoostMetric, we used the codes provided by the authors. We implement our FrobMetric in Matlab and L-BFGS-B is in Fortran and a Matlab interface is made. All the computation time is reported on a workstation with 4 Intel Xeon E5520 (2.27GHz) CPUs (only single core is used) and 32 GB RAM.

Table 1 illustrates that our proposed FrobMetric shows comparable error rates against state-of-the-art methods like LMNN, ITML, and BoostMetric. In terms of computation time, FrobMetric is much faster than all convex optimization based learning methods (LMNN, ITML, BoostMetric) on most data sets. On high-dimensional data sets with many data points, as the theory predicts, FrobMetric can be significantly faster than LMNN. For example, on MNIST, FrobMetric is almost 140 times faster. FrobMetric is also faster than BoostMetric, although at each iteration, the computation complexity for BoostMetric is lower. We observe that BoostMetric needs far more number of iterations for convergence.

Next we use FrobMetric to learn a metric for face recognition on the "Labeled Faces in the Wild" data set [9].

4.1.2 Unconstrained face recognition

In this experiment, we have compared the proposed FrobMetric to the state-of-the-art methods for the task of face pair-matching problem on the "Labeled Faces in the Wild" (LFW) [9] data set. This is a data set of unconstrained face images, which has a large range of the variation seen in real life, including $13,233$ images of $5,749$ people collected from news articles on internet. The face recognition task here is pair matching—given two face images, to determine if these two images are of the same individual. So we classify unseen pairs whether each image in the pair indicates same individual or not, by applying MkNN of [7] instead of kNN.

Features of face images are extracted by computing 3-scale, 128-dimensional SIFT descriptors [12], which center on 9 points of facial features extracted by a facial feature descriptor, same as described in [7]. PCA is then performed on the SIFT vectors to reduce the dimension to between 100 and 400.

Since our proposed FrobMetric adopts the triplet con-

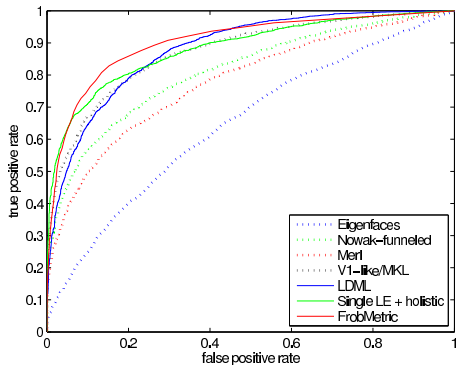
Figure 1: Generated triplets based on pairwise information provided by LFW data set. The first two belong to the same individual and the third is a different individual.

cept, we need to use individual's identity information for generating the third example in a triplet, given a pair. For *match* pairs, we find the third example that belongs to a *different* individual with k nearest neighbors (k is between 5 to 30). For *mismatch* pairs, we find the k nearest neighbors (k is between 5 to 30) that have the same identity with one of the individuals in the given pair. Some of the generated triplets are shown in Figure 1. We select the regularization parameter using cross validation on View 1 and train and test the metric using the 10 provided splits in View 2 as suggested by [9].

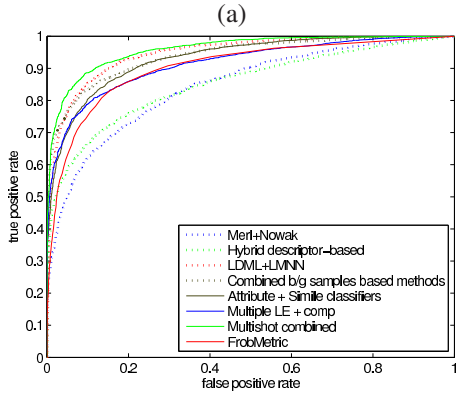
Table 2: Comparison of the face recognition performance accuracy (%) and CPU time of our proposed FrobMetric on LFW datasets varying PCA dimensionality and the number of triplets in each fold for training.

# triplets	100D	200D	300D	400D
Accuracy				
3,000	82.10 (1.21)	83.29 (1.59)	83.81 (1.04)	84.08 (1.18)
6,000	82.26 (1.27)	83.55 (1.28)	84.06 (1.06)	83.91 (1.48)
9,000	82.40 (1.30)	83.62 (1.18)	84.08 (0.92)	84.34 (1.23)
12,000	82.50 (1.22)	83.86 (1.18)	84.13 (0.84)	84.19 (1.31)
15,000	82.55 (1.30)	83.70 (1.22)	84.29 (0.77)	84.27 (0.90)
18,000	82.72 (1.24)	83.69 (1.23)	84.20 (0.84)	84.32 (1.45)
CPU time				
3,000	51s	215s	373s	937s
6,000	100s	222s	661s	1,312s
9,000	142s	534s	1,349s	3,499s
12,000	186s	647s	1,295s	6,418s
15,000	235s	704s	1,706s	3,616s
18,000	237s	830s	2,342s	7,621s

Simple recognition systems with a single descriptor Table 2 shows our FrobMetric's performance by varying PCA dimensionality and the number of triplets. Increasing the number of training triplets gives slight improvement of



(a)



(b)

Figure 2: (a) ROC Curves that use a single descriptor and a single classifier. (b) ROC curves that use hybrid descriptors or single classifiers and FrobMetric's curve. Each point on a curve is the average over the 10 runs.

recognition accuracy. The dimension after PCA has more impact on the final accuracy for this task. We also report the CPU time of the FrobMetric.

In Figure 2, we have drawn ROC curves of other algorithms for face recognition. To draw our ROC curve, MkNN has moved the threshold value across the distributions of match and mismatch similarity scores. Figure 2 (a) shows methods that use a single descriptor and a single classifier only. As can be seen, our system using FrobMetric outperforms all the others in the literature.

Complex recognition systems with one or more descriptors Figure 2 (b) plots the performance of more complicated recognition systems that use hybrid descriptors or combination of classifiers. See Table 3 for details.

As stated above, the leading algorithms have used either 1) additional appearance information, 2) multiple scores of multiple descriptors, or 3) complex recognition systems with hybrid of two or more methods. In contrast, our system using FrobMetric employs neither a combination of other methods nor the use of multiple descriptors. That is, our system exploits a very simple pipeline of recognition. As a result, this system could reduce computational cost for extracting descriptors, generating prior information, training methods and computing the recognition scores.

With such a simple metric learning, our method is only

Table 3: Test accuracy (%) on LFW datasets. ROC curve labels in Figure 2 are described here with details.

	SIFT or single descriptor + single classifier	multiple descriptors or classifiers
Matthew *et al.* [19]	60.02 (0.79) 'Eigenfaces'	-
Nowak *et al.* [13]	73.93 (0.49) 'Nowak-funneled'	-
Huang *et al.* [8]	70.52 (0.60) 'Merl'	76.18 (0.58) 'Merl+Nowak'
Wolf *et al.* in 2008[22]	-	78.47 (0.51) 'Hybrid descriptor-based'
Wolf *et al.* in 2009[23]	72.02 -	86.83 (0.34) 'Combined b/g samples based methods'
Pinto *et al.* [14]	79.35 (0.55) 'V1-like/MKL'	-
Taigman *et al.* [18]	83.20 (0.77) -	**89.50 (0.40)** 'Multishot combined'
Kumar *et al.* [10]	-	85.29 (1.23) 'attribute + simile classifiers'
Cao *et al.* [4]	81.22 (0.53) 'single LE + holistic'	84.45 (0.46) 'multiple LE + comp'
Guillaumin *et al.* [7]	83.2 (0.4) 'LDML'	87.5 (0.4) 'LMNN + LDML'
FrobMetric (this work)	**84.34 (1.23)** 'FrobMetric' on SIFT	-

slightly worse than state-of-the-art hybrid systems. We expect improved accuracy of FrobMetric when more features such LBP are used. In particular, compared to other algorithms adopting a simple recognition system using a single descriptor, our method achieves the best performance.

4.2. Maximum variance unfolding

We can approximately solve MVU's SDP problem using the proposed general Frobenius norm SDP approach. The MVU's optimization problem writes

$$\max_{\mathbf{X}} \mathbf{Tr}(\mathbf{X}) \text{ s.t. } \langle \mathbf{A}_i, \mathbf{X} \rangle \leq b_i, \forall i; \mathbf{1}^\top \mathbf{X} \mathbf{1} = 0; \mathbf{X} \succcurlyeq 0.$$

Here $\{\mathbf{A}_i, b_i\}$, $i = 1 \cdots$, encodes the local distance constraints. This problem can be solved using off-the-shelf SDP solvers, which is not scalable. Instead, we change the objective function to $\max_{\mathbf{X}} \mathbf{Tr}(\mathbf{X}) - \frac{1}{2\sigma}||\mathbf{X}||_{\mathrm{F}}^2$. When σ is sufficiently large, the solution to this Frobenius norm perturbed version is a reasonable approximation to the original problem. So we use the proposed approach to solve MVU approximately.

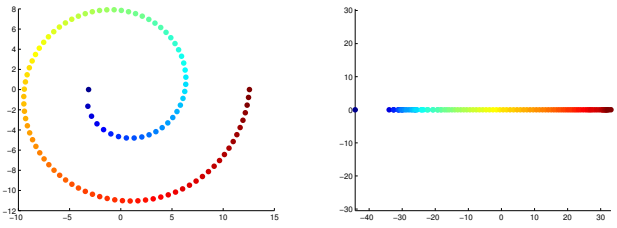


Figure 3: 2D curve data and the embedded data.

Figure 3 shows an example on toy data with 100 points. We find $k = 5$ nearest neighbors to construct the local distance constraints and $\sigma = 10^4$ for our algorithm. MVU (code from the author) needs 3 seconds and ours is less than 1 second.

The teapot images were obtained by rotating a teapot which contains 400 images of different views. Each image size in the set is 101×76 pixels. Figure 4 shows two dimensional embedding of the teapot data sets by our method. It takes about 4 seconds using $k = 5$ and $\sigma = 10^{10}$ for our method and MVU needs about 85 seconds. We have also used the first 200 images. For these data images, we set $k = 30$ and $\sigma = 10^8$ and we obtained our results in 6 seconds while MVU needs 723 seconds. As can be seen, our method preserves the order of teapot images corresponding to the angles images taken and produces plausible embeddings.

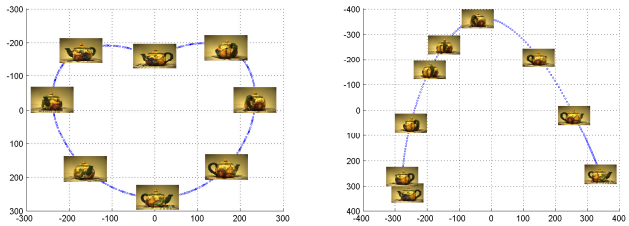


Figure 4: The embedding results of 400 teapot images and 200 teapot images.

Figure 5 shows two-dimensional embedding of face data sets. The data contains 1965 images (28×20) of different views and expressions of the same individual face. Our

computational time to solve this metric was 131 seconds using $k = 5$ nearest neighbors whereas the original MVU needed 4732 seconds.

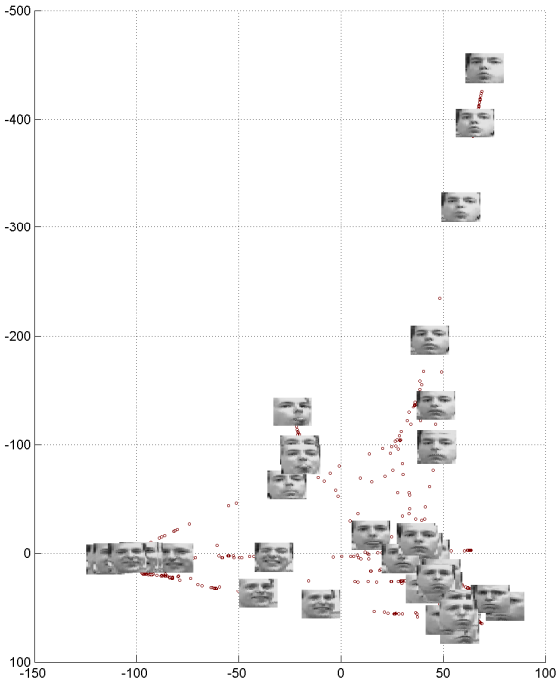


Figure 5: 2D embedding of face data by our approach.

To show the efficiency of our approach, in Figure 6 we have compared the computational time between the original MVU implementation and the proposed method, by varying the number of data samples, which determines the number of variables in MVU. Note that the original MVU implementation uses CSDP [1], which is an interior-point based Newton algorithm. We use the "Swiss roll" data here, which has dimensionality $d = 3$.

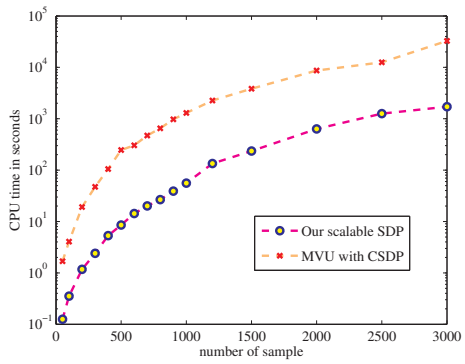


Figure 6: Comparison of computational time between MVU and our fast approach. Our algorithm uses $\sigma = 10^2$. Our algorithm is about 15 times faster.

5. Conclusion

We have presented an efficient and scalable semidefinite metric learning algorithm. Our algorithm is simple to implement and much more scalable than most SDP solvers. The key observation is that, instead of solving the original primal problem, we solve the Lagrange dual problem by exploiting its special structure. Experiments on UCI benchmark data sets as well as the unconstrained face recognition task show its efficiency and efficacy.

We have also extended it to solve more general Frobenius norm regularized SDPs. Experiments on the MVU problem demonstrates its scalability.

References

[1] B. Borchers. CSDP, a C library for semidefinite programming. *Optim. Methods and Softw.*, 11(1):613–623, 1999.

[2] J. Borwein and A. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer-Verlag, New York, 2000.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, 2010.

[5] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proc. Int. Conf. Mach. Learn.*, pages 209–216, Corvalis, Oregon, 2007. ACM Press.

[6] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Proc. Adv. Neural Inf. Process. Syst.* MIT Press, 2004.

[7] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2009.

[8] G. B. Huang, M. J. Jones, and E. Learned-Miller. LFW results using a combined nowak plus merl recognizer. In *Faces in Real-Life Images Workshop, in Euro. Conf. Comp. Vision*, 2008.

[9] 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. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[10] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2009.

[11] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.: Series A and B*, 45(3):503–528, 1989.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.*, 60(2):91–110, 2004.

[13] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, 2007.

[14] N. Pinto, J. DiCarlo, and D. Cox. How far can you get with a modern face recognition test set using only simple features? In *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, 2009.

[15] G. Raetsch, B. Schoelkopf, A. Smola, K. R. Muller, T. Onoda, and S. Mika. ν-Arc: Ensemble learning in the presence of outliers. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 561–567. MIT Press, 2000.

[16] C. Shen, J. Kim, L. Wang, and A. van den Hengel. Positive semidefinite metric learning with boosting. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1651–1659, 2009.

[17] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *Proc. Euro. Conf. Comp. Vis.*, volume 4, pages 776–792, London, UK, 2002. Springer-Verlag.

[18] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *Proc. British Mach. Vis. Conf.*, 2009.

[19] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. Comp. Vis. Pattern Recogn.*, 1991.

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

[21] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comp. Vis.*, 70(1):77–90, 2005.

[22] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop, in Euro. Conf. Comp. Vision*, 2008.

[23] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Proc. Asian Conf. Comp. Vis.*, 2009.

[24] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Proc. Adv. Neural Inf. Process. Syst.* MIT Press, 2002.