Efficient Dual Approach to Distance Metric Learning

Chunhua Shen, Junae Kim, Fayao Liu, Lei Wang, *Member, IEEE*, and Anton van den Hengel

***Abstract*— Distance metric learning is of fundamental interest in machine learning because the employed distance metric can significantly affect the performance of many learning methods. Quadratic Mahalanobis metric learning is a popular approach to the problem, but typically requires solving a semidefinite programming (SDP) problem, which is computationally expensive. The worst case complexity of solving an SDP problem involving a matrix variable of size $D \times D$ with $O(D)$ linear constraints is about $O(D^{6.5})$ using interior-point methods, where D is the dimension of the input data. Thus, the interior-point methods only practically solve problems exhibiting less than a few thousand variables. Because the number of variables is $D(D+1)/2$, this implies a limit upon the size of problem that can practically be solved around a few hundred dimensions. The complexity of the popular quadratic Mahalanobis metric learning approach thus limits the size of problem to which metric learning can be applied. Here, we propose a significantly more efficient and scalable approach to the metric learning problem based on the Lagrange dual formulation of the problem. The proposed formulation is much simpler to implement, and therefore allows much larger Mahalanobis metric learning problems to be solved. The time complexity of the proposed method is roughly $O(D^3)$, which is significantly lower than that of the SDP approach. Experiments on a variety of data sets demonstrate that the proposed method achieves an accuracy comparable with the state of the art, but is applicable to significantly larger problems. We also show that the proposed method can be applied to solve more general Frobenius norm regularized SDP problems approximately.**

***Index Terms*— Convex optimization, Lagrange duality, Mahalanobis distance, metric learning, semidefinite programming (SDP).**

I. Introduction

DISTANCE metric learning has attracted much research interests recently in the machine learning and pattern recognition community because of its wide applications in various areas [1]–[4]. Methods relying upon the identification of an appropriate data-dependent distance metric have been applied to a range of problems from image classification and object recognition to the analysis of genomes. The performance of many classic algorithms such as k-nearest neighbor (k-NN) and k-means clustering depend critically upon the distance metric employed.

Large-margin metric learning is an approach that focuses on identifying a metric by which the data points within the same class lie close to each other and those in different classes are separated by a large margin. Weinberger *et al.* large-margin nearest neighbor (LMNN) [1] is a seminal work illustrating the approach whereby the metric takes the form of a Mahanalobis distance. Given input data $\alpha \in \mathbb{R}^D$, this approach to the metric learning problem can be framed as that of learning the linear transformation $\mathbf{L}$, which optimizes a criterion expressed in terms of Euclidean distances among the projected data $\mathbf{L}\alpha \in \mathbb{R}^d$.

To obtain a convex problem, instead of learning the projection matrix ($\mathbf{L} \in \mathbb{R}^{D\times d}$), one usually optimizes over the quadratic product of the projection matrix ($\mathbf{X} = \mathbf{L}\mathbf{L}^\top$) [1], [3]. This linearization convexifies the original nonconvex problem. The projection matrix may then be recovered by an eigen-decomposition or Cholesky decomposition of $\mathbf{X}$.

Typical methods that learn the projection matrix $\mathbf{L}$ are most of the spectral dimensionality reduction methods such as principle component analysis (PCA), Fisher linear discriminant analysis (LDA), neighborhood component analysis (NCA) [5], and relevant component analysis (RCA) [6]. Goldberger *et al.* [5] showed that NCA may outperform traditional dimensionality reduction methods. NCA learns the projection matrix directly through optimization of a nonconvex objective function. NCA is therefore prone to become trapped in local optima, particularly when applied to high-dimensional problems. RCA [6] is an unsupervised metric learning method. RCA does not maximize the distance between different classes, but minimizes the distance between data in Chunklets. Chunklets consist of data that come from the same (although unknown) class.

More methods on the topic of large-margin metric learning actually learn $\mathbf{X}$ directly because Xing *et al.* [3] proposed a global distance metric learning approach using a convex optimization method. Although the experiments in [3] show improved performance on clustering problems, this is not the case when the method is applied to most of the classification problems. Davis *et al.* [7] proposed an information theoretic metric learning (ITML) approach to the problem. The closest

Manuscript received May 29, 2012; accepted July 20, 2013. Date of publication September 4, 2013; date of current version January 10, 2014. The work of C. Shen was supported in part by ARC under Grant LP120200485, and in part by ARC Future Fellowship FT120100969. The work of A. van den Hengel was supported by ARC under Grant LP120200485.

C. Shen, F. Liu, and A. van den Hengel are with the Australian Center for Visual Technologies, and School of Computer Science, The University of Adelaide, Adelaide 5005, Australia (e-mail: chunhua.shen@adelaide.edu.au; fayao.liu@adelaide.edu.au; anton.vandenhengelg@adelaide.edu.au).

J. Kim was with NICTA, Canberra Research Laboratory, Adelaide 2600, Australia. She is now with Defence Science and Technology Organisation (DSTO), Edinburgh 5111, Australia (e-mail: junae.kim@gmail.com).

L. Wang is with the School of Computer Science and Software Engineering, University of Wollongong, Wollongong 2522, Australia (e-mail: leiw@uow.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2275170

work to ours may be LMNN [1] and BoostMetric [8]. LMNN is a Mahanalobis metric form of k-NN whereby the Mahanalobis metric is optimized such that the k-NN are encouraged in belonging to the same class while data points from different classes are separated by a large margin. The optimization takes the form of an SDP problem. To improve the scalability of the algorithm, instead of using standard SDP solvers, Weinberger *et al.* [1] proposed an alternating estimation and projection method. At each iteration, the updated estimate $\mathbf{X}$ is projected back to the semidefinite cone using eigen-decomposition, to preserve the semidefiniteness of $\mathbf{X}$. In this sense, at each iteration, the computational complexity of their algorithm is similar to that of ours. The alternating method, however, needs an extremely large number of iterations to converge (the default value being 10 000 in the authors' implementation). In contrast, our algorithm solves the corresponding Lagrange dual problem and needs only 20–30 iterations in most of the cases. In addition, the algorithm proposed here is significantly easier to implement.

As pointed in these earlier works, the disadvantage of solving for $\mathbf{X}$ is that one needs to solve a semidefinite programming (SDP) problem because $\mathbf{X}$ must be positive semidefinite (p.s.d.). Conventional interior-point SDP solvers have a computation complexity of $O(D^{6.5})$, where D is the dimension of input data. This high complexity hampers the application of metric learning to high-dimensional problems.

To tackle this problem, here, we propose a new formulation of quadratic Mahalanobis metric learning using proximity comparison information and Frobenius norm regularization. The main contribution is that, with the proposed formulation, we can very efficiently solve the SDP problem in the dual space. Because strong duality holds, we can then recover the primal variable $\mathbf{X}$ from the dual solution. The computational complexity of the optimization is dominated by eigen-decomposition, which is $O(D^3)$, and hence the overall complexity is $O(t \cdot D^3)$, where t is the number of iterations required for convergence. Note that t does not depend on the size of the data, and typically $t \approx 20$–30.

A number of methods exist in the literature for large-scale p.s.d.metric learning. Shen *et al.* [8], [9] introduced BoostMetric by adapting the boosting technique, typically applied to classification, to distance metric learning. This paper exploits an important theorem, which shows that a p.s.d.matrix with trace of one can always be represented as a convex combination of multiple rank-one matrixes. Demiriz *et al.* [10] generalized LPBoost and AdaBoost by showing that it is possible to use matrixes as weak learners within these algorithms, in addition to the more traditional use of classifiers or regressors as weak learners. The approach we propose here, FrobMetric, is inspired by BoostMetric in the sense that both algorithms use proximity comparisons between triplets as the source of the training information. The critical distinction between FrobMetric and BoostMetric, however, is that reformulating the problem to use the Frobenius regularization—rather than the trace norm regularization—allows the development of a dual form of the resulting optimization problem, which may be solved far more efficiently. The BoostMetric approach iteratively computes the squared Mahalanobis distance metric using a rank-one update at each iteration. This has the advantage that only the leading eigenvector needs to be calculated, but leads to slower convergence. Indeed, for BoostMetric, the convergence rate remains unclear. The proposed FrobMetric method, in contrast, requires more calculations per iteration, but converges in significantly fewer iterations. Actually in our implementation, the convergence rate of FrobMetric is guaranteed by the employed quasi–Newton method.

The main contributions of this paper are as follows.

1) We propose a novel formulation of the metric learning problem, based on the application of Frobenius norm regularization.
2) We develop a method for solving this formulation of the problem, which is based on optimizing its Lagrange dual. This method may be practically applied to much more complex data sets than the competing SDP approach, as it scales better to large databases and to high-dimensional data.
3) We generalize the method such that it may be used to solve any Frobenius norm regularized SDP problem. Such problems have many applications in machine learning and computer vision, and by way of example, we show that it may be used to approximately solve the Frobenius norm perturbed maximum variance unfolding (MVU) problem [11]. We demonstrate that the proposed method is considerably more efficient than the original MVU implementation on a variety of data sets and that a plausible embedding is obtained.

The proposed scalable semidefinite optimization method can be viewed as an extension of [12]. The subject in [12] was similarly a semidefinite least squares problem: finding the covariance matrix that is closest to a given matrix under the Frobenius norm metric. Here, we study the large-margin Mahalanobis metric learning problem, where, in contrast, the objective function is not a least squares fitting problem. We also discuss, in Section III, the application of the proposed approach to general SDP problems, which have Frobenius norm regularization terms. Note also that a precursor to the approach described here also appeared in [13]. Here, we have provided more theoretical analysis as well as experimental results.

In summary, we propose a simple, efficient, and scalable optimization method for quadratic Mahalanobis metric learning. The formulated optimization problem is convex, thus guaranteeing that the global optimum can be attained in polynomial time [14]. Moreover, by working with the Lagrange dual problem, we are able to use off-the-shelf eigen-decomposition and gradient descent methods such as L-BFGS-B to solve the problem.

A. Notation

A column vector is denoted by a bold lower case letter ($\boldsymbol{x}$) and a matrix is by a bold upper case letter ($\mathbf{X}$). The fact that a matrix $\mathbf{A}$ is p.s.d.is denoted as $\mathbf{A} \succcurlyeq 0$. The inequality $\mathbf{A} \succcurlyeq \mathbf{B}$ is intended to show that $\mathbf{A} - \mathbf{B} \succcurlyeq 0$. In the case of vectors, $\boldsymbol{a} \geq \boldsymbol{b}$ denotes the elementwise version of the inequality, and when applied relative to a scalar (e.g., $\boldsymbol{a} \geq 0$),

the inequality is intended to apply for every element of the vector. For matrixes, we denote the vector space of real matrixes of size $m \times n$ by $\mathbb{R}^{m\times n}$, and the space of real symmetric matrixes as $\mathbb{S}$. Similarly, the space of symmetric matrixes of size $n \times n$ is $\mathbb{S}^n$, and the space of symmetric p.s.d.matrixes of size $n \times n$ is denoted as $\mathbb{S}^n_+$. The inner product defined on these spaces is $\langle \mathbf{A}, \mathbf{B}\rangle = \mathbf{Tr}(\mathbf{A}^\top \mathbf{B})$. Here, $\mathbf{Tr}(\cdot)$ calculates the trace of a matrix. The Frobenius norm of a matrix is defined as $\|\mathbf{X}\|_{\mathrm{F}}^2 = \mathbf{Tr}(\mathbf{X}\mathbf{X}^\top) = \mathbf{Tr}(\mathbf{X}^\top\mathbf{X})$, which is the sum of all the squared elements of $\mathbf{X}$; $\mathbf{diag}(\cdot)$ extracts the diagonal elements of a square matrix. Given a symmetric matrix $\mathbf{X}$ and its eigen-decomposition $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$ ($\mathbf{U}$ being an orthonormal matrix, and $\mathbf{\Sigma}$ being real and diagonal), we define the positive part of $\mathbf{X}$ as

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

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

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

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

B. Euclidean Projection Onto the p.s.d.Cone

Our proposed method relies on the following standard results, which can be found in textbooks such as Chapter 8 of [14]. The p.s.d.part $(\mathbf{X})_+$ of $\mathbf{X}$ is the projection of $\mathbf{X}$ onto the p.s.d.cone

$$(\mathbf{X})_+ = \left\{ \min_{\mathbf{Y}} \|\mathbf{Y} - \mathbf{X}\|_{\mathrm{F}}^2, \text{ s.t. } \mathbf{Y} \succcurlyeq 0 \right\}. \tag{1}$$

It is not difficult to check that, for any $\mathbf{Y} \succcurlyeq 0$,

$$\|\mathbf{X} - (\mathbf{X})_+\|_{\mathrm{F}}^2 = \|(\mathbf{X})_-\|_{\mathrm{F}}^2 \le \|\mathbf{X} - \mathbf{Y}\|_{\mathrm{F}}^2.$$

In other words, although the optimization problem in (1) appears as an SDP problem, it can be simply solved using eigen-decomposition, which is efficient. Therefore, if a problem is to seek a p.s.d.matrix that minimizes the Frobenius norm, then it can be easily solved by conducting eigen-decomposition. It is this key observation that serves as the backbone of the proposed fast method.

The rest of the paper is organized as follows. In Section II, we present the main algorithm for learning a Mahalanobis metric using an efficient optimization. In Section III, we extend our algorithm to more general Frobenius norm regularized semidefinite problems. The experiments on various data sets are in shown Section IV and we conclude the paper in Section V.

II. Large-Margin Distance Metric Learning

We now briefly review quadratic Mahalanobis distance metrics. Suppose that we have a set of triplets $\mathcal{Q} = \{(\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k)\}$, which encodes proximity comparison information. Suppose also that $\mathbf{dist}_{ij}$ computes the Mahalanobis distance between $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ under a proper Mahalanobis matrix. That is, $\mathbf{dist}_{ij} = \|\boldsymbol{a}_i - \boldsymbol{a}_j\|_{\mathbf{X}}^2 = (\boldsymbol{a}_i - \boldsymbol{a}_j)^\top \mathbf{X}(\boldsymbol{a}_i - \boldsymbol{a}_j)$, where $\mathbf{X} \in \mathbb{S}_+^{D\times D}$ is p.s.d.. Such a Mahanalobis metric may equally be parameterized by a projection matrix $\mathbf{L}$, where $\mathbf{X} = \mathbf{L}\mathbf{L}^\top$.

Let us define the margin associated with a training triplet 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 $\mathbf{A}_r = (\boldsymbol{a}_i - \boldsymbol{a}_k)(\boldsymbol{a}_i - \boldsymbol{a}_k)^\top - (\boldsymbol{a}_i - \boldsymbol{a}_j)(\boldsymbol{a}_i - \boldsymbol{a}_j)^\top$. Here, r is the index of the current triplet within the set of m training triplets $\mathcal{Q}$. As will be shown in the experiments below, this type of proximity comparison among triplets may be easier to obtain that explicit distances for some applications like image retrieval. Here, the metric learning procedure solely relies on the matrixes $\mathbf{A}_r$ ($r = 1, \ldots, m$).

A. Primal Problems of Mahanalobis Metric Learning

Putting it into the large-margin learning framework, the optimization problem is to maximize the margin with a regularization term that is intended to avoid overfitting (or, in some cases, makes the problem well posed)

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

Here, $\mathbf{Tr}(\mathbf{X}) = 1$ removes the scale ambiguity in $\mathbf{X}$. This is the formulation proposed in BoostMetric [8]. We can write the above problem equivalently as

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

These formulations are exactly equivalent given the appropriate choice of the tradeoff parameters C_1 and C_2. The theorem is as follows.

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

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

See Appendix for the proof.

Both problems can be written in the form of standard SDP problems because the objective function is linear and a p.s.d.constraint is involved. Recall that we are interested in a Frobenius norm regularization rather than a trace norm regularization. The key observation is that the Frobenius norm regularization term leads to a simple and scalable optimization. Therefore, replacing the trace norm in (P2) with the Frobenius norm we have

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

Although (P3) and (P2) are not exactly the same, the only difference is the regularization term. Different regularizations can lead to different solutions. However, as the ℓ_1 and ℓ_2 norm regularizations in the case of vector variables such as in support vector machines (SVMs), in general, these two regularizations would perform similarly in terms of the final classification accuracy. Here, one does not expect that a

particular form of regularization, either the trace or Frobenius norm regularization, would perform better than the other one. As we have pointed out, the advantage of the Frobenius norm is faster optimization.

One may convert (P3) into a standard SDP problem by introducing an auxiliary variable

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

The last constraint can be formulated as a p.s.d.constraint $\begin{bmatrix} 1 & \mathbf{X} \\ \mathbf{X} & 2\delta \end{bmatrix} \succcurlyeq 0$. Therefore, in theory, we can use an off-the-shelf SDP solver to solve this primal problem directly. As mentioned previously, the computational complexity of this approach is, however, very high, meaning that only small-scale problems can be solved within reasonable CPU time limits.

Next, we show that the Lagrange dual problem of (P3) has some desirable properties.

B. Dual Problems and Desirable Properties

We first introduce the Lagrangian dual multipliers, $\mathbf{Z}$ which we associate with the p.s.d.constraint $\mathbf{X}\succcurlyeq 0$, and $\boldsymbol{u}$ which we associate with the remaining constraints upon $\boldsymbol{\xi}$.

The Lagrangian of (P3) then becomes

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

with $\boldsymbol{u}\geq 0$ 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, from which we see that

$$\mathbf{X}^{\star} = \mathbf{Z}^{\star} + \textstyle\sum_r u_r^{\star}\mathbf{A}_r \tag{2}$$

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

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

This dual problem still has a p.s.d.constraint and it is not clear how it may be solved more efficiently than using standard interior-point methods. Note, however, that as both the primal and dual problems are convex, Slater's condition holds, under mild conditions (see [14] for details). Strong duality therefore holds between (P3) and (D3), which means that the objective values of these two problem coincide at optimality and in many cases, we are able to indirectly solve the primal by solving the dual and vice versa. The Karush–Kuhn–Tucker (KKT) conditions (2) enable 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) may be simplified as follows:

$$\min_{\mathbf{Z}}\ \left\|\mathbf{Z}+\textstyle\sum_{r=1}^{m}u_r\mathbf{A}_r\right\|_{\mathrm{F}}^2,\ \text{s.t.:}\ \mathbf{Z}\succcurlyeq 0. \tag{3}$$

To simplify the Notation, we define $\hat{\mathbf{A}}$ as a function of $\boldsymbol{u}$

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

Problem (3) then becomes that of finding the p.s.d. matrix $\mathbf{Z}$ such that $\|\mathbf{Z}-\hat{\mathbf{A}}\|_{\mathrm{F}}^2$ is minimized. This problem has a closed-form solution, which is the positive part of $\hat{\mathbf{A}}$

$$\mathbf{Z}^{\star} = (\hat{\mathbf{A}})_{+}. \tag{4}$$

Now, the original dual problem may be simplified

$$\max_{\boldsymbol{u}}\ \textstyle\sum_{r=1}^{m}u_r - \tfrac{1}{2}\left\|(\hat{\mathbf{A}})_{-}\right\|_{\mathrm{F}}^2,\ \text{s.t.:}\ \tfrac{C_3}{m}\geq \boldsymbol{u}\geq 0. \tag{5}$$

The KKT condition is simplified into

$$\mathbf{X}^{\star} = (\hat{\mathbf{A}})_{+} - \hat{\mathbf{A}} = -(\hat{\mathbf{A}})_{-}. \tag{6}$$

From the definition of the operator $(\cdot)_{-}$, $\mathbf{X}^{\star}$ computed by (6) must be p.s.d. Note that we have now achieved a simplified dual problem, which has no matrix variables and only simple box constraints on $\boldsymbol{u}$. The fact that, the objective function of (5) is differentiable (but not twice differentiable), allows us to optimize for $\boldsymbol{u}$ in (5) using gradient descent methods (see Sect. 5.2 in [15]). To illustrate why the objective function is differentiable, we can see the following simple example. For $F(\mathbf{X}) = 1/2\left\|(\mathbf{X})_{-}\right\|_{\mathrm{F}}^2$, the gradient can be calculated as

$$\nabla F(\mathbf{X}) = (\mathbf{X})_{-}$$

because of the following fact. Given a symmetric $\delta\mathbf{X}$, we have

$$F(\mathbf{X}+\delta\mathbf{X}) = F(\mathbf{X}) + \mathbf{Tr}(\delta\mathbf{X}(\mathbf{X})_{-}) + o(\delta\mathbf{X}).$$

This can be verified using the perturbation theory of eigenvalues of symmetric matrixes. When we set $\delta\mathbf{X}$ to be very small, the above equality is the definition of gradient.

Hence, we can use a sophisticated off-the-shelf first-order Newton algorithm such as L-BFGS-B [16] to solve (5). In summary, the optimization procedure is as follows.

1) Input the training triplets and calculate $\mathbf{A}_r$, $r = 1,\ldots,m$.
2) Calculate the gradient of the objective function in (5), and use L-BFGS-B to optimize (5).
3) Calculate $\hat{\mathbf{A}}$ using the output of L-BFGS-B (namely, $\boldsymbol{u}^{\star}$) and compute $\mathbf{X}^{\star}$ from (6) 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 (5). Note that other gradient methods such as 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 (5) can be calculated as

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

Therefore, at each iteration, the computation of $(\hat{\mathbf{A}})_{-}$, which requires full eigen-decomposition, only needs to be calculated once to evaluate all of 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 computational complexity at each iteration. In this case, the overall complexity is $O(t\cdot D^3)$ with t being around 20–30.

III. General Frobenius Norm SDP

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

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

We consider its Frobenius norm regularized version

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

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

$$\min_{\mathbf{Z}, \boldsymbol{u}} \ \tfrac{1}{2}\sigma \|\mathbf{Z} - \mathbf{C} - \hat{\mathbf{A}}\|_{\mathrm{F}}^2 + \boldsymbol{b}^{\top}\boldsymbol{u}, \ \text{s.t.} \ \mathbf{Z} \succcurlyeq 0, \boldsymbol{u} \geq 0. \tag{7}$$

The KKT condition is

$$\mathbf{X}^{\star} = \sigma(\mathbf{Z}^{\star} - \hat{\mathbf{A}} - \mathbf{C}) \tag{8}$$

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

$$\mathbf{Z} = (\mathbf{C} + \hat{\mathbf{A}})_{+}. \tag{9}$$

Therefore, we can simplify (7) into

$$\min_{\boldsymbol{u}} \ \tfrac{1}{2}\sigma \|(\mathbf{C} + \hat{\mathbf{A}})_{-}\|_{\mathrm{F}}^2 + \boldsymbol{b}^{\top}\boldsymbol{u}, \ \text{s.t.} \ \boldsymbol{u} \geq 0. \tag{10}$$

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

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

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.

To demonstrate the performance of the proposed general Frobenius norm SDP approach, we will show how it may be applied to the problem of MVU. The MVU 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, $\{\mathbf{A}_i, b_i\}, i = 1 \cdots$, encode the local distance constraints. This problem can be solved using off-the-shelf SDP solvers, which, as is described above, does not scale well. Using the proposed approach, we modify the objective function to $\max_{\mathbf{X}} \mathbf{Tr}(\mathbf{X}) - 1/2\sigma ||\mathbf{X}||_{\mathrm{F}}^2$. When σ is sufficiently large, the solution to this Frobenius norm perturbed version is a reasonable approximation to the original problem. We can use the proposed approach to solve MVU approximately.

IV. Experimental Results

We first run the metric learning experiments on UCI benchmark data, face recognition, and action recognition data sets. We then approximately solve the MVU problem [11] using the proposed general Frobenius norm SDP approach.

A. Distance Metric Learning

1) UCI Benchmark Test: We perform a comparison between the proposed FrobMetric and a selection of the current state-of-the-art distance metric learning methods, including RCA [6], NCA [5], LMNN [1], BoostMetric [8], and ITML [7] on data sets from the UCI repository.

We have included the results of PCA, LDA, and SVMs with Gaussian kernel as baseline approaches. A more recent state-of-the-art method, named extreme learning machines (ELMs) [17], is also compared here using Gaussian kernel. We used the `libsvm` [18] implementation for SVMs results[1] and code downloaded from the author's website[2] for ELMs results. The kernel width and the regularization parameters are cross validated on the validation set from $\{2^{-15}, 2^{-13}, \ldots, 2^{5}\}$ and $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$, respectively. Note that ELMs ran out of memory on MNIST and performed on par with SVM on other data sets.

As in [1], for some data sets (MNIST, Yale faces, and USPS), we have applied PCA to reduce the original dimensionality and noise.

For all the experiments, the task is to classify unseen instances in a testing subset. To accumulate statistics, the data are randomly split into 10 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 FrobMetric and BoostMetric in [8], we use three nearest neighbors to generate triplets and check the performance using 3-NN. For each training sample $\boldsymbol{a}_i$, we find its three nearest neighbors in the same class and the three nearest neighbors in the difference classes. With three nearest neighbors' information, the number of triplets of each data set for FrobMetric and BoostMetric are shown in Table I. 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 3-NN and computational time for each learning metric are shown as well.

Experiment settings for LMNN and ITML follow [1] and [7], 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 are reported on a workstation with four Intel Xeon E5520 (2.27 GHz) CPUs (only single core is used) and 32-GB RAM.

Table I illustrates that the proposed FrobMetric shows error rates comparable with state-of-the-art methods such as LMNN, ITML, and BoostMetric. It also performs on par with a nonlinear SVM on these data sets.

In terms of computation time, ELM runs extremely fast, taking only 2s on letters data set, while less than 1s on all other data sets. SVM ranks the second. For a fair comparison, we only report the computation time of several convex opti-

[1]LMNN can solve for either $\mathbf{X}$ or the projection matrix $\mathbf{L}$. When LMNN solves for $\mathbf{X}$ on Wine set, the error rate is 20.77% $\pm$ 14.18%.

[2]http://www.ntu.edu.sg/home/egbhuang/

TABLE I
TEST ERRORS OF VARIOUS METRIC LEARNING METHODS ON UCI DATA SETS WITH 3-NN. NCA [5] DOES NOT OUTPUT A RESULT ON THOSE LARGER DATA SETS BECAUSE OF MEMORY PROBLEMS. STANDARD DEVIATION IS 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)**
SVM	2.97	**2.15 (0.30)**	2.96	4.94 (2.14)	**5.59 (3.61)**	1.15 (1.86)	3.64 (3.59)
ELM [17]	-	2.42 (0.39)	2.82	**3.09 (0.65)**	9.03 (4.73)	2.69 (3.17)	**3.18 (3.07)**
RCA [6]	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 [5]	-	-	-	-	18.28 (3.58)	28.08 (7.49)	**3.18 (3.74)**
LMNN [1]	**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 [7]	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 [8]	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**

mization based metric learning methods in Table I. As can be observed, FrobMetric is much faster than all compared metric learning methods (LMNN, ITML, and BoostMetric) on most of the data sets. On high-dimensional data sets with many data points, as the theory predicts, FrobMetric is 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 computational complexity of BoostMetric is lower. We observe that BoostMetric requires significantly more iterations to converge.

Next, we use FrobMetric to learn a metric for face recognition on the labeled faces in the wild (LFW) data set [19].

2) Unconstrained Face Recognition: In this experiment, we have compared the proposed FrobMetric to state-of-the-art methods for the task of face pair-matching problem on the LFW [19] data set. This is a data set of unconstrained face images, including 13 233 images of 5749 people collected from the news articles on the Internet. The data set is particularly interesting because it captures much of the variation seen in real images of faces. The face recognition task here is to determine whether a presented pair of images is of the same individual. Therefore, we classify unseen pairs whether each image in the pair shows same individual or not, by applying Mk-NN of [20] instead of k-NN.

Features of face images are extracted by computing three-scale, 128-dimensional SIFT descriptors [21], which center on nine points of facial features extracted by a facial feature descriptor, as described in [20]. PCA is then performed on the SIFT vectors to reduce the dimension between 100 and 400.

Because the proposed FrobMetric method adopts the triplet-training concept, we need to use individual's identity information to generate the third example in a triplet, given a pair. For matched pairs, we find the third example that belongs to a different individual with k-NN (k is between five and 30). For mismatched pairs, we find the k-NN (k is between five and 30) that have the same identity as one of the individuals in the given pair. Some of the generated triplets are shown in Fig. 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 in [19].

a) Simple recognition systems with a single descriptor: Table II shows the performance of FrobMetric's under varying PCA dimensionality and number of triplets. Increasing the number of training triplets gives a slight improvement in recognition accuracy. The dimension after PCA has more impact on the final accuracy for this task. We also report the CPU time required.

In Fig. 2, we show ROC curves for FrobMetric and related face recognition algorithms. These curves were generated by altering the threshold value across the distributions of match and mismatch similarity scores within Mk-NN. Fig. 2(a) shows the methods that use a single descriptor and a single classifier only. As can be seen, our system using FrobMetric outperforms all others.

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

TABLE II
COMPARISON OF THE FACE RECOGNITION PERFORMANCE ACCURACY (%) AND CPU TIME OF OUR PROPOSED FROBMETRIC ON LFW DATA SETS 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

b) Complex recognition systems with one or more descriptors: Fig. 2(b) plots the performance of more complicated recognition systems that use hybrid descriptors or combinations of classifiers. See Table III for details. Cao *et al.* [30] performed a pose-adaptive matching applying pose-specific classifiers on LFW, labeled single LE + holistic and multiple LE + comp in Fig. 2. Kumar *et al.* [29] show attribute + simile classifiers used additional features such

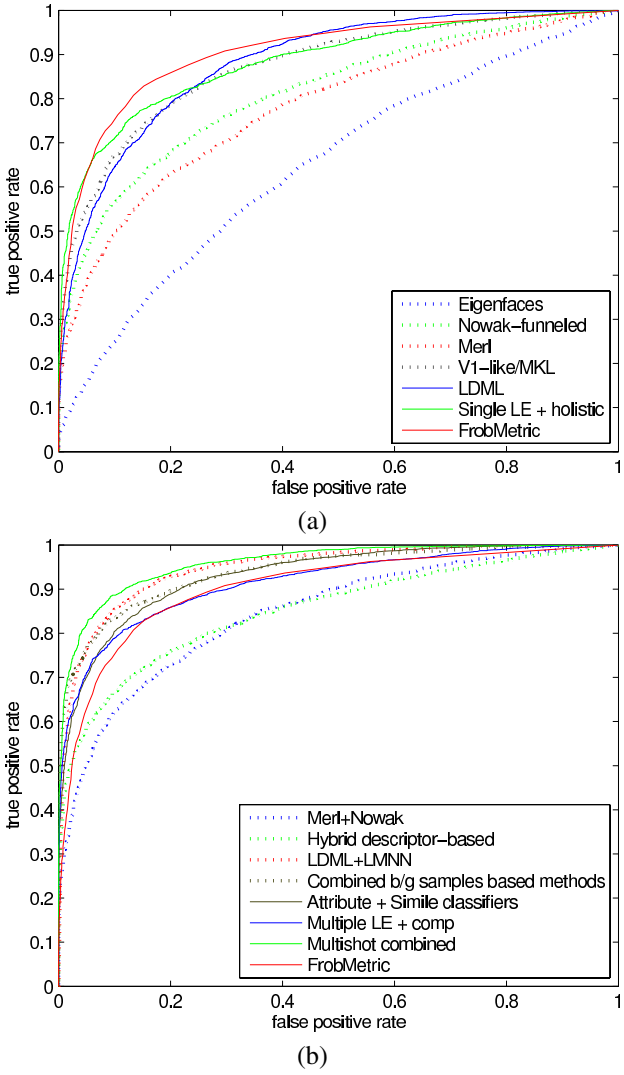

Fig. 2. (a) ROC curves that use a single descriptor and a single classifier. (b) ROC curves that use hybrid descriptors or single classifiers and FrobMetric's curve. Each point on a curve is the average over 10 runs.

as pose and expression. Varying pose and facial expression results in larger image differences. Therefore, it is obvious that knowing such additional information would boost the recognition performance. In addition, Kumar *et al.* collected huge-sized labels outside of the LFW data set.

Taigman *et al.* [28] work separated facial images using pose information to improve performance as well. Moreover, their best performance, "Multishot combined" used eight descriptors (SIFT, LBP, TPLBP and FPLBP, and four with square root of four descriptors) and combined 16 scores using the ITML + Multiple OSS ID method and the pose-based multiple shots. Applying SVMs classifier, the accuracy of "Multishot combined," 89.50% became the best among computer vision approaches on LFW in the literature. However, when their system used Multiple OSS with SIFT only and did not apply pose information, the accuracy was decreased to 83.20% while our FrobMetric's best accuracy was 84.34% as shown in Table III.

TABLE III
TEST ACCURACY (%) ON LFW DATA SETS. ROC CURVE LABELS IN FIG. 2 ARE DESCRIBED HERE WITH DETAILS

	SIFT or single descriptor + single classifier	multiple descriptors or classifiers
Turk *et al.* [22]	60.02 (0.79) 'Eigenfaces'	-
Nowak *et al.* [23]	73.93 (0.49) 'Nowak-funneled'	-
Huang *et al.* [24]	70.52 (0.60) 'Merl'	76.18 (0.58) 'Merl+Nowak'
Wolf *et al.* in 2008 [25]	-	78.47 (0.51) 'Hybrid descriptor-based'
Wolf *et al.* in 2009 [26]	72.02 -	86.83 (0.34) 'Combined b/g samples based methods'
Pinto *et al.* [27]	79.35 (0.55) 'V1-like/MKL'	-
Taigman *et al.* [28]	83.20 (0.77) -	**89.50 (0.40)** 'Multishot combined'
Kumar *et al.* [29]	-	85.29 (1.23) 'attribute + simile classifiers'
Cao *et al.* [30]	81.22 (0.53) 'single LE + holistic'	84.45 (0.46) 'multiple LE + comp'
Guillaumin *et al.* [20]	83.2 (0.4) 'LDML'	87.5 (0.4) 'LMNN + LDML'
FrobMetric (this work)	**84.34 (1.23)** 'FrobMetric' on SIFT	-

Wolf *et al*. [26] used a hybrid descriptor which has 10 distances, 10 One-Shot distances, 10 Two-Shot distances, 10 ranking based distances and 20 additional dimensions using LDA. Applying the hybrid descriptors and adding SVM, the paper achieved 86.83% accuracy which labeled 'Combined b/g samples based methods', whereas the accuracy became 72.02% when single SIFT feature in Funneled image was used at this task.

Guillaumin *et al*. [20], 'LDML + LMNN' in Fig. 2 combined 8 scores of 4 descriptors with the combination of LDML and LMNN. Their best results was 87.5% while LDML using single descriptor showed 83.20% accuracy.

As stated above, the leading algorithms have used either 1) additional appearance information; 2) multiple scores from multiple descriptors; or 3) complex recognition systems with hybrids of two or more methods. In contrast, our system using FrobMetric employs neither a combination of other methods nor multiple descriptors. That is, our system exploits a very simple recognition pipeline. The method hence reduces the computational costs associated with extracting the descriptors, generating the prior information, training, and computing the recognition scores.

With such a simple metric learning approach, and modest computational cost, it is notable that the method is only slightly outperformed by state-of-the-art hybrid systems (test accuracy of 84.34% $\pm$ 1.23% versus 89.50% $\pm$ 0.40% on the LFW data sets). We would expect that the accuracy of the FrobMetric approach would improve similarly if more features, such as local binary pattern [31] for instance, were used.

The FrobMetric approach shows better classification performance at a lower computational cost than comparable single descriptor methods. Despite this level of performance, it is surprisingly simple to implement, in comparison to the state-of-the-art.

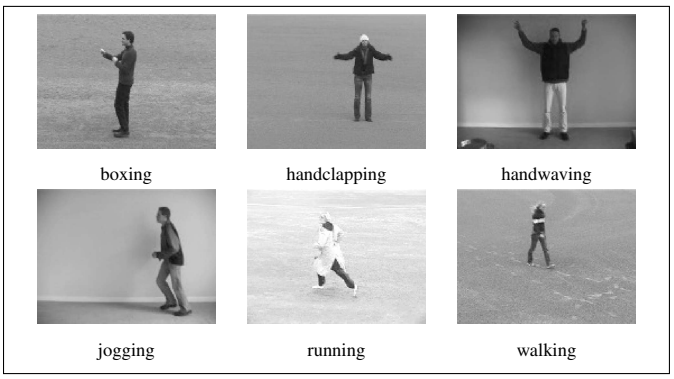

Fig. 3. Examples of the actions from the KTH action data set [32].

3) Metric Learning for Action Recognition: In this experiment, we compare the performance of the proposed method with that of existing approaches on two action recognition benchmark data sets, KTH [32] and Weizmann [33]. Some examples of the actions are shown in Fig. 3. We aim to demonstrate again the advantage of our method in reducing computational overhead while achieving excellent recognition performance.

The KTH data set in this experiment consists 2387 video sequences. They can be categorized into six types of human actions including boxing, hand clapping, jogging, running, walking, and hand waving. These actions are conducted by 25 subjects and each action is performed multiple times by the same subject. The length of each video is about 4 s at 25 fps, and the resolution of each frame is 160 $\times$ 120.

TABLE IV
COMPARISON OF FROBMETRIC AND OTHER METRIC LEARNING METHODS ON ACTION RECOGNITION DATA SETS WITH 3-NN (STANDARD DEVIATION IS REPORTED FOR THE DATA SETS HAVING MULTIPLE RUNS)

		KTH	Weizmann
# samples		2,387	5,594
# triplets		13,761	35,280
dimension		500	286
# training		1,529	3,920
test		858	1,674
# classes		6	10
# runs		10	10
Error Rates %	Euclidean	10.55 (2.46)	1.14 (0.19)
	RCA	21.05 (3.86)	3.21 (0.66)
	LMNN	15.72 (2.57)	0.30 (0.09)
	ITML	27.67 (1.47)	1.06 (0.16)
	BoostMetric	7.05 (1.42)	0.85 (0.31)
	FrobMetric	7.03 (1.46)	0.59 (0.20)
Comp. Time	LMNN	1023.89s	1343.25s
	ITML	1004.94s	368.68s
	BoostMetric	4048.67	1139.02s
	FrobMetric	289.58s	169.30s

We randomly split all the video sequences based on the subjects into 10 pairs, each of which contains all the sequences from 16 subjects for training and those from the remaining nine subjects for test. The space–time interest points [34] were extracted from each video sequence and the the corresponding descriptors were calculated. The descriptors extracted from all the training sequences were clustered into 4000 clusters using k-means, with the cluster centers used to form a visual codebook. Accordingly, each video sequence is characterized by a 4000-dimensional histogram stating the occurrence of each visual word in this sequence. To achieve a compact and discriminative representation, a recently proposed visual word merging algorithm, called AIB [35], is applied to merge the histogram bins to reduce the dimensionality. Subsequently, each video sequence is represented by a 500-dimensional histogram.

The Weizmann data set contains temporal segmentations of video sequences into 10 types of human actions including running, walking, skipping, jumping jack, jumping forward on two legs, jumping in place on two legs, galloping sideways, waving two hands, waving one hand, and bending. The actions are performed by nine actors. The action video sequences are represented by space–time shape features such as space–time saliency, degree of plateness, and degree of stickness, which compute the degree of location and orientation movement in space–time domain by a Poisson equation [36]. This leads to a 286-dimensional feature vector for each action video sequence, which is as in [36]. In this experiment, 70% sequences are used for training and the remaining 30% for testing.

The experimental results are shown in Table IV. The first part of this table shows the experimental setting and the second compares the results of various metric learning methods. On the KTH data set, the proposed method, FrobMetric, performs almost as well as BoostMetric with an error rate of $7.03 \pm 1.46\%$, and outperforms all others. In doing so, FrobMetric requires only 289.58 s to complete the metric learning, which is approximately one quarter of the time required by the fastest competing method (which has more than double the error rate). On the Weizmann data set, the error rate of FrobMetric is $0.59 \pm 0.20\%$, which is the second best among all the compared methods. It is slightly higher than (but still comparable with) the lowest one $0.30 \pm 0.09\%$ obtained by LMNN. However, in terms of computational efficiency, FrobMetric requires approximately one-eighth of the time used by LMNN, and is the fastest of the methods compared. These results demonstrate the computational efficiency and the excellent classification performance of the proposed method in action recognition.

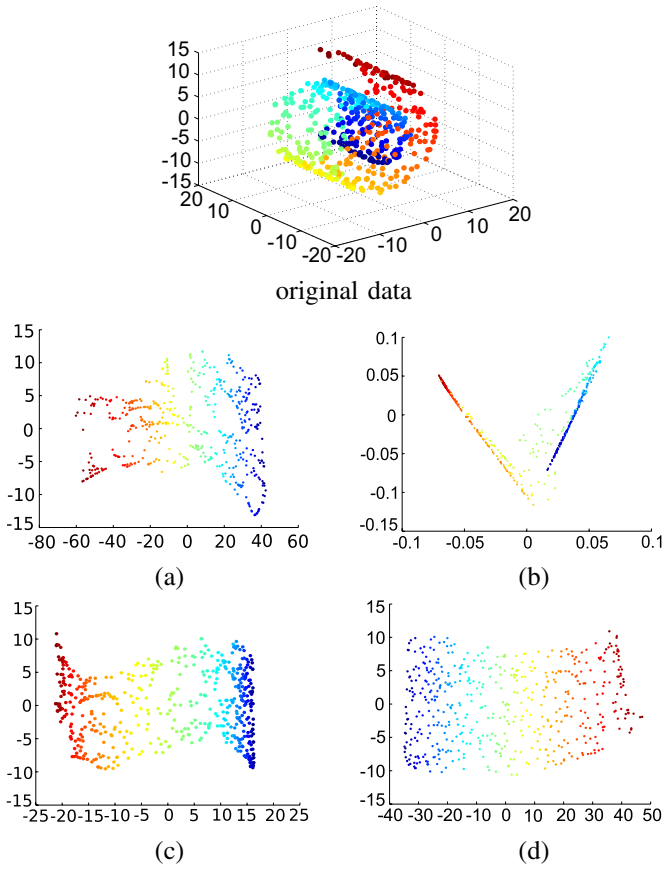

Fig. 4. Embedding results of different methods on the 3-D swiss-roll data set, with the neighborhood size $k = 6$ for all, and $\sigma = 10^5$ for our method. (a) Isomap. (b) LLE. (c) Our method. (d) MVU.

B. Maximum Variance Unfolding

In this section, we run MVU experiments on a few data sets and compare with other embedding methods. Fig. 4 shows the embedding results for several different methods, namely, isometric mapping (Isomap) [37], locally linear embedding (LLE) [38], and MVU [11] on the 3-D swiss-roll with 500 points. We use 6-NN to construct the local distance constraints and set $\sigma = 10^5$.

We have also applied our method to the teapot and face image data sets from [11]. The teapot set contains 200 images obtained by rotating a teapot through 360°. Each image is of 101×76 pixels. Fig. 5 shows the 2-D embedding results of our method and MVU. As can be seen, both methods preserve the order of teapot images corresponding to the angles from which

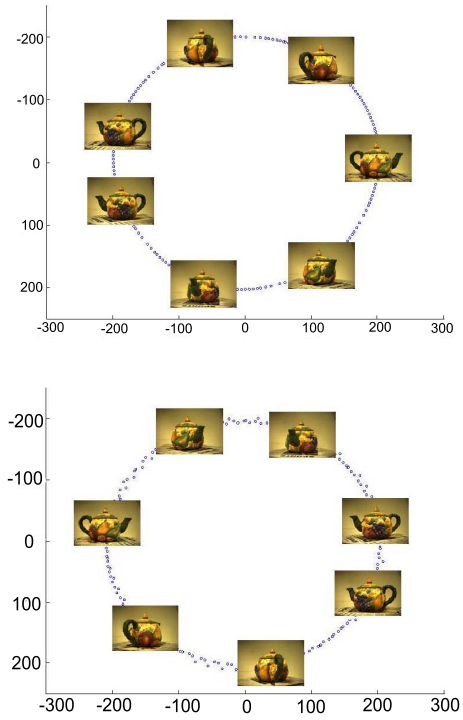

Fig. 5. Embedding results of our method and MVU on the teapot data set. Top: our results with $\sigma = 10^{10}$. Bottom: MVU's results.

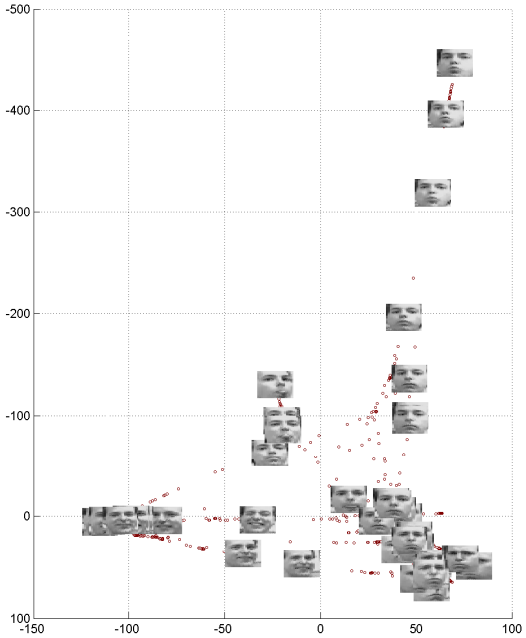

Fig. 6. 2-D embedding of face data by our approach.

the images were taken, and produce plausible embeddings. However, in terms of running time, our algorithm is more than an order of magnitude faster than MVU, requiring only 4 s to run using $k = 6$ and $\sigma = 10^{10}$, whereas MVU required 85 s.

Fig. 6 shows a 2-D embedding of the images from the face data set. The set contains 1965 images (at 28×20 pixels) of the same individual from different views and with differing

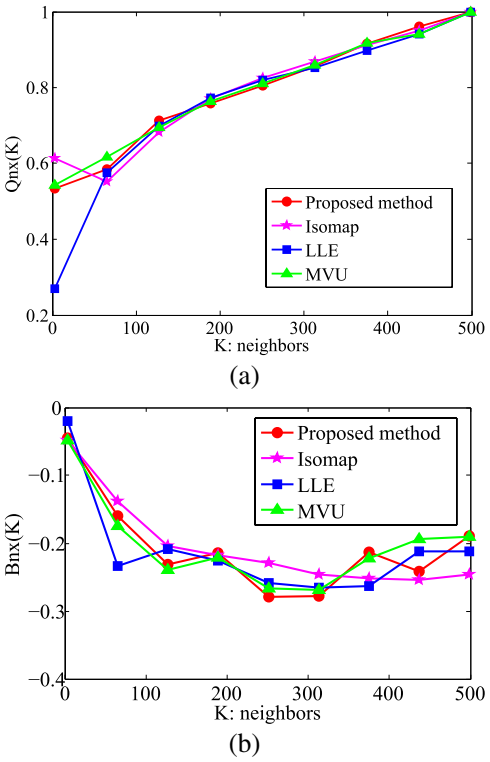

Fig. 7. Quality assessment of neighborhood preservation of different algorithms on 3-D swiss-roll. (a) $Q_{nx}(K)$. (b) $B_{nx}(K)$.

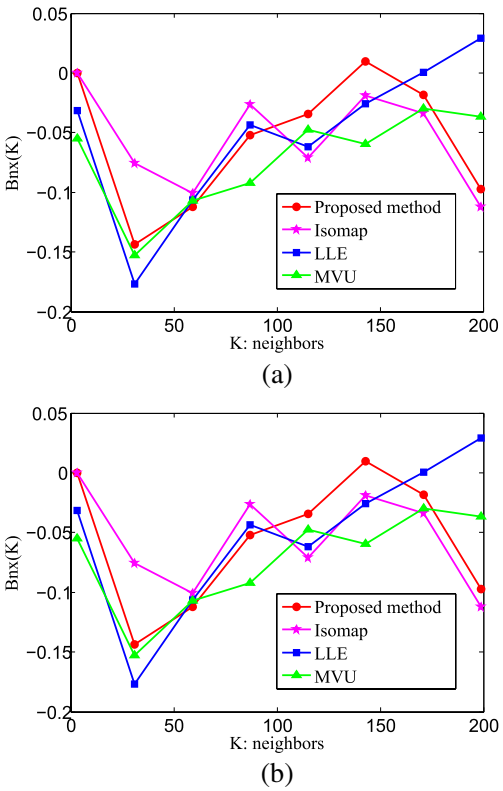

Fig. 8. Quality assessment of neighborhood preservation of different algorithms on the teapot data set. (a) $Q_{nx}(K)$. (b) $B_{nx}(K)$.

expressions. The proposed method required 131 s to solve this metric using 5-NN whereas the original MVU needed 4732 s.

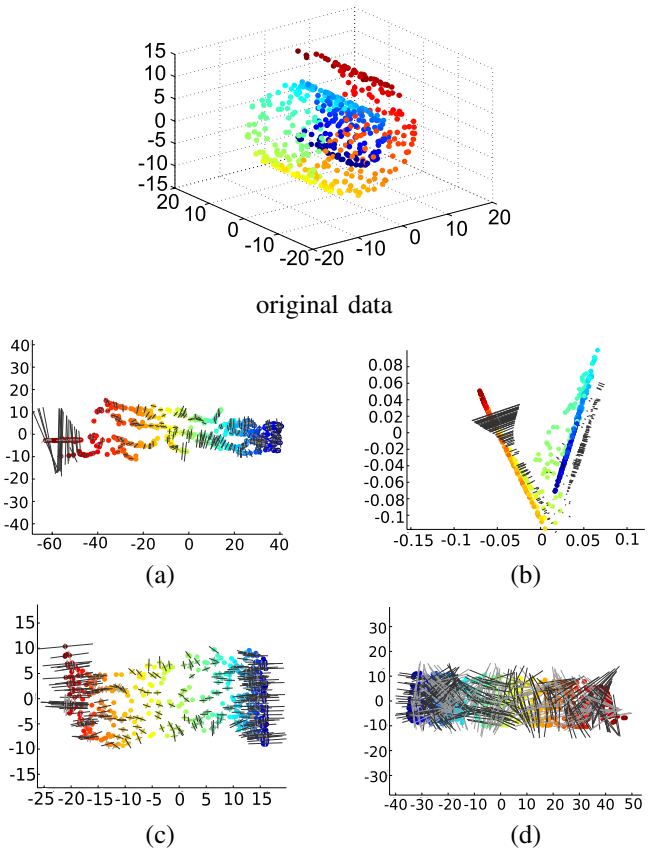

Fig. 9. Visualization results based on PSD on the 3-D swiss-roll, with the neighborhood size $k = 6$ for all, and $\sigma = 10^5$ for our method. (a) Isomap. (b) LLE. (c) Our method. (d) MVU.

TABLE V
ALSTD AND ALED RESULTS ON THE 3-D SWISS-ROLL AND TEAPOT DATA SETS

Algorithm	FrobMetric	MVU	Isomap	LLE
ALSTD				
Swiss roll	0.113	**0.038**	0.245	0.328
Teapot	0.377	0.481	**0.0611**	0.0805
ALED				
Swiss roll	0.311	**0.102**	0.668	0.886
Teapot	1.0	1.28	**0.173**	0.232

TABLE VI
ALCD AND GSCD RESULTS ON THE 3-D SWISS-ROLL AND TEAPOT DATA SETS

Algorithm	FrobMetric	MVU	Isomap	LLE
ALCD				
Swiss roll	0.983	0.964	0.969	**0.994**
Teapot	0.995	0.942	**0.997**	0.995
GSCD				
Swiss roll	0.1163	**0.0404**	0.2572	0.3317
Teapot	0.3792	0.5105	**0.0613**	0.0808

1) Quantitative Assessment: To better illustrate the effectiveness of our method, here we provide a quantitative evaluation of the embeddings generated for the 3-D swiss-roll and teapot data sets. Specifically, we adopt two quality mapping indexes, the unweighted Q_{nx} and B_{nx} [39], to measure

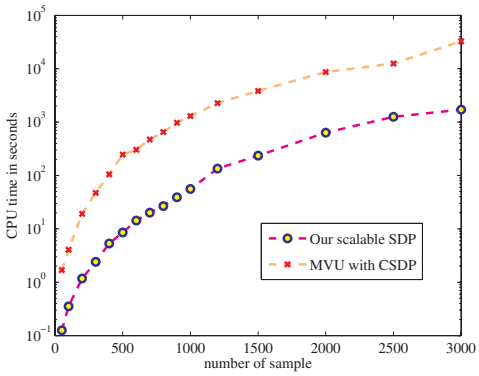

Fig. 10. Comparison of computational time on the 3-D swiss-roll data set between MVU and our fast approach. Our algorithm uses $\sigma = 10^2$. Our algorithm is about 15 times faster. Note that the y-axis is in log scale.

the K-ary neighborhood preservation between the high- and low-dimensional spaces. Q_{nx} represents the proportion of points that remains inside the K-neighborhood after projection, and thus larger Q_{nx} shows better neighborhood preservation. B_{nx} is defined as the difference in the fractions of mild K-extrusions and mild K-intrusions. It shows the behavior of a dimensionality reduction method, namely, whether it tends to produce an intrusive ($B_{nx}(K) > 0$) or extrusive ($B_{nx}(K) < 0$) embedding. Intrusive embedding tends to crush the manifold, which means faraway points can become neighbors after embedding, while extrusive one tends to tear the manifold, meaning some close neighbors can be embedded faraway from each other. In an ideal projection, B_{nx} should be zero. See [39] for more details.

The comparison of LLE, Isomap, MVU, and our proposed method on the teapot (with $\sigma = 10^{10}$) and the swiss-roll ($\sigma = 10^5$) data sets are shown in Figs. 7 and 8. As can be seen from Figs. 7 and 8(a), the proposed FrobMetric method performs on par with MVU, while better than both Isomap and LLE in terms of neighborhood preservation. Note that all the methods tend to tear the manifold as $B_{nx}(K)$ is below zero in all the cases.

We have also made quantitative analysis of the proposed algorithm based on [40]. They proposed several quantitative criteria, specifically, average local standard deviation (ALSTD) and average local extreme deviation (ALED) to measure the global smoothness of a recovered low-dimensional manifold, average local co-directional consistence (ALCD) to estimate the average codirectional consistence of the principle spread direction (PSD) of the data points, and a combined criterion to simultaneously evaluate the global smoothness and codirectional consistence (GSCD).

We give the visual results of swiss-roll data set based on PSD in Fig. 9, in which the longer line at each sample represents the first PSD, and the second line is orthogonal to the first PSD. We also report the ALSTD and ALED, ALCD and GSCD in Tables V and VI. From these tables, we see that MVU performs best on this swiss-roll data set, whereas the proposed FrobMetric method ranks the second best. On the teapot data set, the proposed method performs slightly better than MVU, while worse than both Isomap and LLE. Overall, the proposed method is similar to the original MVU in terms

of these embedding quality criteria. The proposed method is, however, much faster than MVU in all the cases.

To show the efficiency of our approach, in Fig. 10, 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 [41], which is an interior-point-based Newton algorithm. We use the 3-D swiss-roll data here.

V. Conclusion

We have presented an efficient and scalable semidefinite metric learning algorithm. Our algorithm is simple to implement and much more scalable than most of the 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.

Appendix

A. Proof of Theorem 1

Proof: It is well known that the necessary and sufficient conditions for the optimality of SDP problems are primal feasibility, dual feasibility, and equality of the primal and dual objectives. We can easily derive the duals of (P1) and (P2), respectively,

$$\begin{aligned} \min_{\gamma,\boldsymbol{u}} \quad & \gamma \\ \text{s.t.}: \quad & \textstyle\sum_{r=1}^{m} u_r \mathbf{A}_r \preccurlyeq \gamma \mathbf{I}, \\ & \mathbf{1}^\top \boldsymbol{u} = 1; 0 \leq \boldsymbol{u} \leq \tfrac{C_1}{m} \end{aligned} \tag{D1}$$

and

$$\begin{aligned} \max_{\boldsymbol{u}} \quad & \textstyle\sum_{r=1}^{m} u_r \\ \text{s.t.}: \quad & \textstyle\sum_{r=1}^{m} u_r \mathbf{A}_r \preccurlyeq \mathbf{I}, \\ & 0 \leq \boldsymbol{u} \leq \tfrac{C_2}{m}. \end{aligned} \tag{D2}$$

Here, $\mathbf{I}$ is the identity matrix.

Let $(\mathbf{X}^\star, \boldsymbol{\xi}^\star, \rho^\star)$ represent the optimum of the primal problem (P1). Primal feasibility of (P1) implies primal feasibility of (P2), and so that

$$\left\langle \mathbf{A}_r, \tfrac{\mathbf{X}^\star}{\rho^\star} \right\rangle \geq 1 - \tfrac{\xi^\star}{\rho^\star}.$$

Let $(\gamma^\star, \boldsymbol{u}^\star)$ be the optimal solution of the dual problem (D1). Dual feasibility of (D1) implies dual feasibility of (D2), and thus that $\sum_r u_r^\star/\gamma^\star \mathbf{A}_r \preccurlyeq \mathbf{I}$, and $0 \leq \boldsymbol{u}^\star/\gamma^\star \leq C_2/(m\gamma^\star)$. Because the duality gap between (P1) and (D1) is zero, $\text{Opt(D1)} = \gamma^\star = \text{Opt(P1)}$.

Finally, we need to show that the objective function values of (P2) and (D2) are the same. This is easy to verify from the fact that $\text{Opt(P1)} = \text{Opt(D1)}$:

$$\begin{aligned} \rho^\star - \tfrac{C_1}{m}\mathbf{1}^\top \boldsymbol{\xi} &= \gamma^\star \\ \Longrightarrow \rho^\star(\mathbf{1}^\top \boldsymbol{u}^\star) - \tfrac{C_1}{m}\mathbf{1}^\top \boldsymbol{\xi} &= \gamma^\star \mathbf{Tr}(\mathbf{X}^\star) \\ \Longrightarrow \mathbf{Tr}\,(\mathbf{X}^\star)/\rho^\star + \tfrac{C_2}{m}\mathbf{1}^\top \boldsymbol{\xi}^\star/\rho^\star &= (\mathbf{1}^\top \boldsymbol{u}^\star)/\gamma^\star. \end{aligned}$$

This concludes the proof. ∎

References

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

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

[3] 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*. Cambridge, MA, USA: MIT Press, 2002, pp. 505–512.

[4] C. Domeniconi, D. Gunopulos, and J. Peng, "Large margin nearest neighbor classifiers," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 899–909, Jul. 2005.

[5] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood component analysis," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2004, pp. 513–520.

[6] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment learning and relevant component analysis," in *Proc. Eur. Conf. Comput. Vis.*, vol. 4. London, U.K., 2002, pp. 776–792.

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. Int. Conf. Mach. Learn.*, Corvalis, OR, USA, 2007, pp. 209–216.

[8] C. Shen, J. Kim, L. Wang, and A. van den Hengel, "Positive semidefinite metric learning with boosting," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2009, pp. 1651–1659.

[9] C. Shen, J. Kim, L. Wang, and A. van den Hengel, "Positive semidefinite metric learning using boosting-like algorithms," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1007–1036, 2012.

[10] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor, "Linear programming boosting via column generation," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 225–254, Mar. 2002.

[11] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 77–90, 2005.

[12] S. Boyd and L. Xiao, "Least-squares covariance matrix adjustment," *SIAM J. Matrix Anal. Appl.*, vol. 27, no. 2, pp. 532–546, 2005.

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

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

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

[16] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program., Ser. A, B*, vol. 45, no. 3, pp. 503–528, 1989.

[17] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[18] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[19] 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," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.

[20] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 498–505.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[22] M. A. Turk and A. P. Pentland, "Face recognition using Eigenfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1991, pp. 586–591.

[23] E. Nowak and F. Jurie, "Learning visual similarity measures for comparing never seen objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[24] G. B. Huang, M. J. Jones, and E. Learned-Miller, "LFW results using a combined Nowak plus MERL recognizer," in *Proc. Faces Real-Life Images Workshop, ECCV*, 2008.

[25] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Proc. Faces Real-Life Images Workshop ECCV*, 2008.

[26] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on background samples," in *Proc. Asian Conf. Comput. Vis.*, vol. 2. 2009, pp. 88–97.
[27] 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. Comput. Vis. Pattern Recognit.*, Dec. 2009, pp. 2591–2598.
[28] Y. Taigman, L. Wolf, and T. Hassner, "Multiple one-shots for utilizing class label information," in *Proc. Brit. Mach. Vis. Conf.*, 2009.
[29] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2009, pp. 365–372.
[30] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2707–2714.
[31] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
[32] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. Int. Conf. Pattern Recognit.*, vol. 3. 2004, pp. 32–36.
[33] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
[34] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[35] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 179–192.
[36] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 548–561.
[37] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec. 2000.
[38] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec. 2000.
[39] J. Lee and M. Verleysen, "Quality assessment of nonlinear dimensionality reduction based on K-ary neighborhoods," in *Proc. JMLR, Workshop Conf.*, vol. 4. 2008, pp. 21–35.
[40] J. Zhang, Q. Wang, L. He, and Z.-H. Zhou, "Quantitative analysis of nonlinear embedding," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1987–1998, Dec. 2011.
[41] B. Borchers, "CSDP, a C library for semidefinite programming," *Optim. Methods Softw.*, vol. 11, no. 1, pp. 613–623, 1999.

Chunhua Shen received the Ph.D. degree from University of Adelaide, Adelaide, Australia, in 2006.

He has been a Faculty Member with the School of Computer Science, University of Adelaide, since 2011. He was with the Computer Vision Program, National ICT Australia, Canberra Research Laboratory, Canberra, Australia. His current research interests include the intersection of computer vision and statistical machine learning. His recent work has been on real-time object detection, large-scale image retrieval and classification, and scalable nonlinear optimization.

Dr. Shen received the Australian Research Council Future Fellowship in 2012.

Junae Kim received the B.S. degree from Ewha Womans University, Seoul, Korea, in 2000, the M.S. degree from the Pohang University of Science and Technology, Pohang, Korea, in 2002, the M.Phil. and Ph.D. degrees from Australian National University, Acton, Australia, in 2007 and 2011, respectively.

She is currently with DSTO, SA, Australia. Her current research interests include computer vision and machine learning.

Fayao Liu received the B.Eng. and M.Eng. degrees from the School of Computer Science, National University of Defense Technology, Hunan, China, in 2008 and 2010, respectively. She is currently pursuing the Ph.D. degree with the University of Adelaide, Adelaide, Australia.

Her current research interests include machine learning and computer vision.

Lei Wang (SM'10) received the B.Eng. and M.Eng. degrees from Southeast University, Nanjing, China, in 1996 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2004.

He is currently a Senior Lecturer with the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, Australia. His current research interests include machine learning, pattern recognition, and computer vision.

Dr. Wang received the Australian Post-Doctoral Fellowship by the Australian Research Council in 2007 and the Early Career Researcher Award by the Australian Academy of Science in 2009.

Anton van den Hengel received the Bachelor of Mathematical Science, the Bachelor of Laws, the master's degree in computer science, the Ph.D. degree in computer vision from University of Adelaide, Adelaide, Australia, in 1991, 1993, 1994, and 2000, respectively.

He is the Founding Director of The Australian Centre for Visual Technologies, University of Adelaide. He is a Professor with the School of Computer Science, University of Adelaide, Adelaide, Australia.