Learning Locally-Adaptive Decision Functions for Person Verification

Zhen Li *
UIUC
zhenli3@uiuc.edu

Shiyu Chang *
UIUC
chang87@uiuc.edu

Feng Liang
UIUC
liangf@uiuc.edu

Thomas S. Huang *
UIUC
huang@ifp.uiuc.edu

Liangliang Cao
IBM Research
liangliang.cao@us.ibm.com

John R. Smith
IBM Research
jsmith@us.ibm.com

Abstract

This paper considers the person verification problem in modern surveillance and video retrieval systems. The problem is to identify whether a pair of face or human body images is about the same person, even if the person is not seen before. Traditional methods usually look for a distance (or similarity) measure between images (e.g., by metric learning algorithms), and make decisions based on a fixed threshold. We show that this is nevertheless insufficient and sub-optimal for the verification problem. This paper proposes to learn a decision function for verification that can be viewed as a joint model of a distance metric and a locally adaptive thresholding rule. We further formulate the inference on our decision function as a second-order large-margin regularization problem, and provide an efficient algorithm in its dual from. We evaluate our algorithm on both human body verification and face verification problems. Our method outperforms not only the classical metric learning algorithm including LMNN and ITML, but also the state-of-the-art in the computer vision community.

1. Introduction

Person verification, "Are you the person you claim to be," is an important problem with many applications. Modern image retrieval systems often want to verify whether photos contain the same person or the same object. Person verification also gets more and more important for social network websites, where it is highly preferred to correctly assign personal photos to users. More importantly, the huge amount of surveillance cameras - there are more than 30 million surveillance cameras in U. S. recording about 4 billion hours of videos per week, calls for reliable systems which are able to identify the same person across different videos, a critical task that cannot merely rely on human labors. So developing an automatic verification system is of great interest in practice.

Figure 1. Example images of George W. Bush showing huge intra-person variations.

There are two main visual clues for person verification: face images and human body figures. Although our human vision system has the amazing ability of performing verification - we can judge whether two faces are about the same person without even seeing that person before, it is difficult to build a computer-based automatic system for this purpose. For a given query image, the person in the image may not appear in the database or has only one or few images in the database. Furthermore, the query image and the other images in the database are rarely collected in exactly the same environment, which leads to huge intra-person variations including viewpoint, lighting condition, image quality, resolution, etc. Figure 1 provides some examples illustrating the difficulties with the person verification problem.

We can formally describe the verification problem as follows: for a pair of sample images represented by $x, y \in \mathbb{R}^d$, respectively, each of which corresponds to category label $c(x)$ and $c(y)$, we aim to decide whether they are from the same category, i.e., $c(x) = c(y)$, nor not. Given a set of training samples, our goal is to learn a **decision function**

*This research was supported in part by a research grant from Chongqing Institute of Green and Inteligent Technology, Chinese Academy of Sciences.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.463

$f(x, y)$ where

$$f(x,y)\begin{cases} > 0, & \text{if } c(x) = c(y) \\ < 0, & \text{otherwise.} \end{cases} \quad (1)$$

Note that to infer f we need not to know the respective value of $c(x)$ or $c(y)$, which means it has the generalization ability to verify samples from unseen categories.

Learning the decision function f for verification is fundamentally different from learning a classifier for traditional machine learning problems. Traditional machine learning algorithms consider individual samples instead of a pair of samples. This paired setup for verification naturally imposes some symmetry constraint for the decision function, i.e., $f(x, y) = f(y, x)$, a constraint seldom seen in ordinary learning algorithms. Most multi-class classifiers, which model the category-specific probability distributions (for generative models) or learn the decision boundaries (for discriminant models), are not appropriate for verification. For verification, of interest is to determine whether a pair of samples is from the same category or not, but not to answer which category/categories they belong to. The ability of dealing with unseen categories is the key for person verification, since most testing samples are from unseen persons which are not in the training pool.

Recently, metric learning (ML) approaches [26] have been applied to person verification [12, 19, 6, 2]. The key idea behind ML is to learn a parametric distance metric between two images x and y, which in most cases take the form of $(x - y)^t M(x - y)$ where M is a semi-positive definite matrix Then one can decide whether x and y are from the same class based on some thresholding rule, i.e., $(x - y)^t M(x - y) \leq d$.

Although ML is very important for many supervised learning applications (e.g. classification) that often deal with complex and high-dimensional features, it has a few limitations particularly in the verification setting. The objective of many ML algorithms is to ensure that samples from the same class be closer to each other than those from different classes. In other words, it enforces a *relative* ranking constraint between intra-class and inter-class pairs (in terms of pairwise distances), and this is why ML is often tied with the nearest neighbor classifier for a classification task. However, for verification where many test samples might come from unseen classes, nearest neighbor classifiers are not applicable. Then ML only leads to an *absolute* decision rule with a constant threshold d:

$$f_{ML}(x, y) = d - (x - y)^t M(x - y). \quad (2)$$

This intrinsic mismatch (classification vs. verification, relative ranking vs. absolute discrimination) leaves ML approaches not optimal for verification problems.

In Section 2, after showing the sub-optimality of (2), we propose to adjust the decision rule locally, i.e., consider $f(x, y) = d(x, y) - (x - y)^t M(x - y)$ where $d(x, y)$ is a function of x, y rather than a constant. As a starting point, we assume $d(x, y)$ takes a simple quadratic form, which leads to our general second-order decision function. In Section 3, we formulate the inference on our second-order decision function as a large-margin regularization problem, and provide an efficient algorithm based on its dual. Further, we can interpret our approach as learning a linear SVM in a new feature space which is symmetric with respect to (x, y). With this new interpretation, our second-order decision function can be easily generalized to decision functions of high-orders by the kernel trick. In Section 4, we evaluated our proposed algorithm on three person verification tasks. We show that in all cases, our method achieves state-of-the-art results in comparison with existing works. Finally, we give the conclusion remarks in Section 5.

2. Bridging Distance Metric and Local Decision Rules

Metric learning (related to feature selection, dimension reduction, or subspace projection, etc) plays a fundamental role in machine learning. It is particularly important for computer vision applications, where the feature representation of images or videos is usually of complex high-dimensional form [28, 22]. In these cases, the Euclidean norm associated with the original feature space usually does not provide much useful information for the subsequent learning tasks. In most applications we consider here, the sample data are sparse in the high-dimensional feature space. So we focus on metric learning with respect to a global metric, i.e., the matrix M in (2), although learning a local metric has attracted an increasing interest in machine learning research.

However, metric learning itself is insufficient for verification problem, as discussed in Section 1. The problem is that after metric learning, we still need to make a decision. As to be shown below, a simple constant threshold in (2) is sub-optimal, even if the associated metric is correct. A decision rule that can adapt to the local structures of data [9], is the key to achieve good verification performance. To this end, we consider a joint model that bridges a global distance metric and a local decision rule, and we further show the optimality of our method over ML in the verification setting.

Consider $f(x, y) = d(x, y) - (x - y)^t M(x - y)$ where $d(x, y)$ acts as a **local decision rule** for a learned metric M. Since the metric itself is quadratic, as a starting point, we also assume $d(x, y)$ takes a simple quadratic form. We will see later in Section 3 that, this formulation leads to a kernelized large-margin learning problem, and thus can be easily generalized to decision functions of high-orders by the kernel trick [1].

For now, let us focus on the second-order decision rule,

i.e., $d(x,y) = \frac{1}{2}z^tQz + w^tz + b$, where $z^t = [x^t\ y^t] \in \mathbb{R}^{2d}$, $Q = \begin{bmatrix} Q_{xx} & Q_{xy} \\ Q_{yx} & Q_{yy} \end{bmatrix} \in \mathbb{R}^{2d\times 2d}$, $w^t = [w_x^t\ w_y^t] \in \mathbb{R}^{2d}$, and $b \in \mathbb{R}$. Due to the symmetry property with respect to x and y, we can rewrite $d(x,y)$ as follows:

$$\begin{aligned} d(x,y) &= \frac{1}{2}x^t\tilde{A}x + \frac{1}{2}y^t\tilde{A}y + x^t\tilde{B}y + c^t(x+y) + b \\ &= \frac{1}{4}(x-y)^t(\tilde{A}-\tilde{B})(x-y) \\ &\quad + \frac{1}{4}(x+y)^t(\tilde{A}+\tilde{B})(x+y) \\ &\quad + c^t(x+y) + b, \end{aligned} \tag{3}$$

where $\tilde{A} = Q_{xx} = Q_{yy}$ and $\tilde{B} = Q_{xy} = Q_{yx}$ are both $d\times d$ real symmetric matrices (not necessarily positive semidefinite), $c = w_x = w_y$ is a d-dimensional vector, and b is the bias term.

Now we obtain the **second-order decision function** for verification:

$$\begin{aligned} f(x,y) &= d(x,y) - (x-y)^tM(x-y) \\ &= \frac{1}{2}x^tAx + \frac{1}{2}y^tAy + x^tBy \\ &\quad + c^t(x+y) + b, \end{aligned} \tag{4}$$

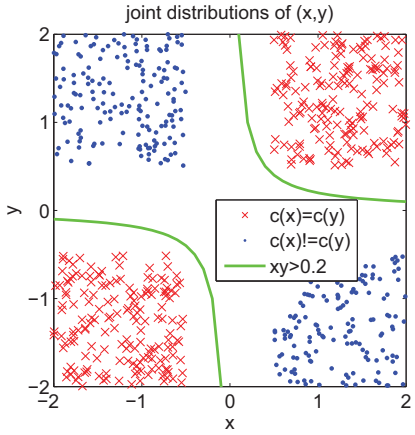
by letting $A - B = \tilde{A} - \tilde{B} - 4M$ and $A + B = \tilde{A} + \tilde{B}$. Again, A and B are real symmetric and need not to be PSD.

The above decision function has the following desirable properties:

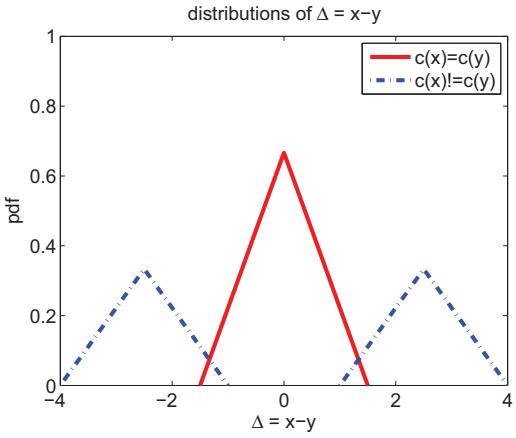
- **Learning globally, acting locally.** We bridge a global metric M and a local decision rule using a joint model (4). Interestingly, the number of parameters is at the same order ($O(d^2)$) as that of ML.
- **Fully informed decision making.** The local decision rule in (3) depends not only on $x - y$, the difference vector usually considered by ML, but also on $x + y$, which contains orthogonal information of (x, y) that would otherwise be neglected by $x - y$ alone.
- **Kernelizable to higher order.** As we will see in Section 3, the decision function in (4) leads to a kernelized large-margin learning problem, and thus can be easily generalized to decision functions of higher-orders by the kernel trick [1].

We now show the optimality of our decision function over ML, by considering a simple case where two categories of samples in $\mathbb{R}^d$ are linearly separable. We show that in the verification setting, the performance of any given metric is inferior to that of our model, in this simple case.

Observation. *Given two linearly separable classes, the verification error rate by our second-order decision function (4) is always lower than that by a learned metric with a fixed threshold (2). More specifically, in this particular setting, our model can always achieve zero verification error while ML does not.*



(a) Joint distribution of (x,y), with zero-error decision function by our model: $xy - 0.2 > 0$.



(b) Distribution of $\Delta = x - y$ in case of metric learning, with finite verification error.

Figure 2. Distributions of same-class pairs (red) vs. different-class pairs (blue).

Proof. Suppose the two classes in $\mathbb{R}^d$ satisfies: $w^tx + b > 0$ for class 1, and $w^tx + b < 0$ for class 2. In verification, we aim to identify if two samples x and y are from the same class or different ones.

1. We first show that our decision function in (4) always achieves zero verification error.

x and y are from the same class if and only if $(w^tx + b)$ and $(w^ty + b)$ are of the same sign. In other words, we can perfectly identify pairs from the same class vs. those from different classes, by checking the sign of $(w^tx + b)(w^ty + b) = x^t(ww^t)y + bw^t(x + y) + b^2$. This decision function is clearly a special case of (4).

2. We then show that the ML approach in (2) does not always achieve zero verification error.

Any Mahalanobis distance between x and y can be regarded

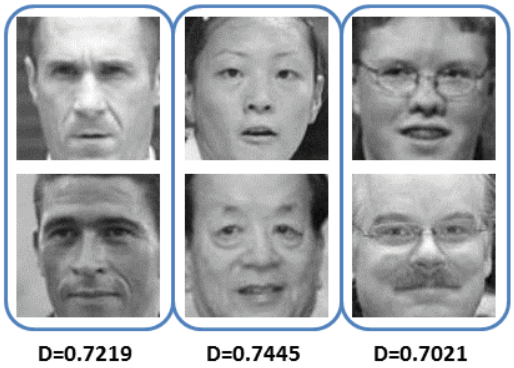
as the Euclidean distance on the space transformed by L, namely, $d(x,y) = (x-y)^t M(x-y) = (x-y)L^t L(x-y) = \|x'-y'\|_2^2$, where $M = L^t L$, $x' = Lx$ and $y' = Ly$. In this new space, the two classes are still linearly separable, since $w^t x + b = w'^t x' + b$ and $w' = wL^{-1}$ (assuming M is full rank). Therefore, in order for ML method in (2), or simply $|x-y| < d$, to achieve zero verification error, the following condition needs to be satisfied:

$$\max_{c(x)=c(y)} \|x'-y'\|_2 < \min_{c(x)\neq c(y)} \|x'-y'\|_2. \qquad (5)$$

Unfortunately, the above condition does not always hold. Consider a counter example in 1-D: class 1 is uniformly distributed in $[-2,\ -0.5]$ and class 2 in $[0.5,\ 2]$. The two classes are indeed separable, but condition (5) is not satisfied since $\max_{c(x)=c(y)} \|x'-y'\|_2 = 1.5$ and $\min_{c(x)\neq c(y)} \|x'-y'\|_2 = 1$. In fact, from Figure 2(b) we see that ML method ($|x-y| < d$) inevitably results in finite verification error, while our model is able to perfectly separate the two types of pairs on the (x,y) space, shown in Figure 2.

A more realistic example that also violates (5) is: face images of the same person but from different poses are usually more dissimilar than those from different person but of the same pose. Figure 3 shows such an example with selected image pairs of the LFW dataset [16]. □

(a) Intra-person distances (different poses).



(b) Inter-person distances (same pose).

Figure 3. Comparison of intra-person and inter-person distances under a learned metric.

3. A Large-Margin Solution with an Efficient Algorithm

3.1. A large margin formulation

Recall that the objective of a verification problem is to learn a symmetric decision function: $f(x,y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that takes a pair of samples $x, y \in \mathbb{R}^d$ as inputs, with decision rule:

$$f(x,y) \begin{cases} > 0, & \text{if } c(x) = c(y) \\ < 0, & \text{otherwise.} \end{cases}$$

Our goal is to find the optimal second-order decision function $f(x,y)$ in (4) that is parametrized by $\{A, B, c, b\}$. This naturally leads to a choice of an SVM-like [4] objective function, as the resulting large-margin model generalizes well to unseen examples.

Specifically, assume we are given a dataset of examples, and pairwise labels are assigned. A sample pair $p_i = (x_i, y_i)$ is labeled as either "positive" ($l_i = +1$), if x_i and y_i are from the same class; or "negative" ($l_i = -1$), otherwise. We further denote by $\mathcal{P}$ the set of all labeled sample pairs. An SVM-like objective function can be formulated as:

$$\begin{aligned} \min \quad & \frac{1}{2}\left(\|A\|_F^2 + \|B\|_F^2 + \|c\|_2^2\right) + \lambda \sum_{i\in\mathcal{P}} \xi_i \qquad (6) \\ \text{s.t.} \quad & l_i f(x_i, y_i) \geq 1 - \xi_i \quad \forall i \in \mathcal{P} \\ & \xi_i \geq 0 \quad \forall i \in \mathcal{P}. \end{aligned}$$

Here $\|A\|_F = \sqrt{\mathbf{tr}(A^t A)}$ is the Frobenius matrix norm, and $\mathbf{tr}(A)$ denotes the trace of matrix A.

Noticing the inner product defined on the matrix space, $\langle A, B\rangle = \mathbf{tr}(A^t B)$, we reformulate the decision function (4) into:

$$\begin{aligned} f(x,y) &= \frac{1}{2}\mathbf{tr}\left(A\left(xx^t + yy^t\right)\right) + \frac{1}{2}\mathbf{tr}\left(B\left(xy^t + yx^t\right)\right) \\ &\quad + c^t(x+y) + b \\ &= \frac{1}{2}\left\langle A,\ xx^t + yy^t\right\rangle + \frac{1}{2}\left\langle B,\ xy^t + yx^t\right\rangle \\ &\quad + \langle c,\ x+y\rangle + b \\ &= \langle \zeta, \psi(x,y)\rangle + b, \qquad (7) \end{aligned}$$

where $\zeta \in \mathbb{R}^{2d^2+2}$ is a vectorized representation of the hyper-parameters (excluding b), and $\psi(x,y)$ defines a map-

ping $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{2d^2+d}$:

$$\zeta = \begin{bmatrix} \mathbf{vec}(A) \\ \mathbf{vec}(B) \\ c \end{bmatrix} \tag{8}$$

$$\psi(x,y) = \begin{bmatrix} \frac{1}{2}\mathbf{vec}(xx^t + yy^t) \\ \frac{1}{2}\mathbf{vec}(xy^t + yx^t) \\ x + y \end{bmatrix}, \tag{9}$$

where $\mathbf{vec}(\cdot)$ denotes the vectorization of a matrix. Note that $\psi(x,y)$ can be viewed as a symmetrization of the original feature space (x,y), that is, any function of $\psi(x,y)$ is now a symmetric function of x and y.

Similarly, the objective function can be rewritten as:

$$\begin{aligned} \min \quad & \frac{1}{2}\langle \zeta, \zeta \rangle + \lambda \sum_{i \in \mathcal{P}} \xi_i \\ \text{s.t.} \quad & l_i(\langle \zeta, \psi_i \rangle + b) \geq 1 - \xi_i \quad \forall i \in \mathcal{P} \\ & \xi_i \geq 0 \quad \forall i \in \mathcal{P}, \end{aligned} \tag{10}$$

where ψ_i is an abbreviation of $\psi(x_i, y_i)$. This looks identical to the standard SVM problem [4]. Thus existing SVM solvers could be employed to solve this problem, such as stochastic gradient decent [23] that works on the primal problem directly, or sequential minimal optimization (SMO) [21] that solves the dual problem instead.

3.2. An efficient dual solver

Though looking straightforward, solving (10) directly is infeasible due to the high dimensionality of $2d^2 + d$. For instance, a moderate image feature of 1000 dimensions will lead to more than 1 million parameters to estimate. What's more, direct application of existing SVM solvers may require forming $\psi(x_i, y_i)$'s explicitly, which is highly inefficient and prohibitive in memory usage. In this section, we will show that the original problem can actually be converted into a kernelized SVM problem that could be solved much more efficiently.

We start with the Lagrange dual of (10):

$$\begin{aligned} \max \quad & \frac{1}{2}\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j l_i l_j \langle \psi_i, \psi_j \rangle \\ \text{s.t.} \quad & \sum_i \alpha_i l_i = 0, \quad 0 \leq \alpha_i \leq \lambda, \end{aligned} \tag{11}$$

where α_i is the Lagrange multiplier corresponding to the i-th constraint. If we could have solved the above problem with optimal α_i^*'s, the solution for the primal is then given by:

$$\begin{aligned} \zeta^* &= \sum_i \alpha_i^* l_i \psi_i \\ b^* &= -l_i - \langle \zeta^*, \psi_i \rangle, \quad \forall i : \alpha_i^* > 0. \end{aligned}$$

And the optimal decision function is therefore:

$$\begin{aligned} f(x,y) &= \langle \zeta^*, \psi(x,y) \rangle + b^* \\ &= \sum_i \alpha_i^* l_i \langle \psi_i, \psi \rangle + b^*. \end{aligned} \tag{12}$$

We notice that either solving the dual problem (11) or applying the optimal function (12) involves only the so-called kernel function $K(\psi_i, \psi_j) = \langle \psi_i, \psi_j \rangle$. By substituting (9) and the equality $\mathbf{vec}(A)^t \mathbf{vec}(B) = \langle A, B \rangle = \mathbf{tr}(A^t B)$, we arrive at:

$$\begin{aligned} K(\psi_i, \psi_j) &= \frac{1}{4}\mathbf{tr}\left((x_i x_i^t + y_i y_i^t)(x_j x_j^t + y_j y_j^t)\right) \\ &\quad + \frac{1}{4}\mathbf{tr}\left((x_i y_i^t + y_i x_i^t)(x_j y_j^t + y_j x_j^t)\right) \\ &\quad + (x_i + y_i)^t (x_j + y_j) \\ &= \frac{1}{4}(x_i^t x_j + y_i^t y_j)^2 + \frac{1}{4}(x_i^t y_j + y_i^t x_j)^2 \\ &\quad + (x_i + y_i)^t (x_j + y_j). \end{aligned} \tag{13}$$

Note that the kernel function here is defined on a new space of $\psi(x,y)$ that is symmetric with respect to x and y. More specifically, different from a traditional kernel function that is between two individual samples, $K(\psi_i, \psi_j)$ is defined between two pairs of samples.

We now see that, to evaluate each kernel function $K(\psi_i, \psi_j)$, one only needs to calculate 4 inner products on $\mathbb{R}^d$: $x_i^t x_j$, $x_i^t y_j$, $y_i^t x_j$, and $y_i^t y_j$, rather than working on the $(2d^2 + d)$-dimensional space instead. In this way we reduce the complexity of each kernel evaluation from $O(d^2)$ to $O(d)$, which is usually the most costly operation in solving large-scale dual SVM problems [7]. In addition, the memory cost is alleviated accordingly, as explicitly constructing $\psi(x,y)$'s by (9) is no longer necessary. Based on (13), existing dual SVM solvers such as SMO algorithm [21, 7] can be applied to solve (11) efficiently.

Moreover, the fact that only inner products are involved in $K(\psi_i, \psi_j)$ implies the extension to implicit kernel embedding of original features, namely,

$$\begin{aligned} K(\psi_i, \psi_j) &= \frac{1}{4}\left(G(x_i, x_j) + G(y_i, y_j)\right)^2 \\ &\quad + \frac{1}{4}\left(G(x_i, y_j) + G(y_i, x_j)\right)^2 \\ &\quad + G(x_i, x_j) + G(x_i, y_j) \\ &\quad + G(y_i, x_j) + G(y_i, y_j), \end{aligned} \tag{14}$$

where $G(\cdot, \cdot)$ is a kernel function of the original feature space. Based on this kernel embedding, we can thus extend our decision function (4) to higher orders by the kernel trick [1]. However, in practice, cubic polynomials or higher order functions often work less robustly, so in experiments, we will mainly use the second-order decision functions.

3.3. Regularizations

In practice, especially when there is only limited amount of training data, we might consider further regularizing the parameters (A and B in particular). For instance, Huang et al. [13] impose various constraints on the learned Mahanalobis matrix (for metric leanring), including positive semi-definity, low rank, sparsity, etc. While all these regularizations can be applied in addition to the Frobenius norm used in (6), we find in practice that positive/negative semi-definity to be particularly useful. Note that here we have two matrices A and B that need to be constrained. Both metric learning and the toy example in Section 2 indicate that we could force A to be positive semi-definite while requiring B be negative semi-definite. So we are adding two constraints to the objective function in (6): $A \in$ PSD and $B \in$ NSD. Gradient projection algorithms can be employed to solve the optimization problem, i.e., after each gradient descent step, we project the updated A onto the PSD space, and B onto the NSD space. Alternatively, we could let $A = MM^t$ and $B = -NN^t$ and optimize on M and N instead. Though the problem on M and N is no longer convex, it does not seem to suffer from seveve local minimum issues [24].

4. Experiments

We conduct experiments on three different datasets: "Viewpoint Invariant Pedestrian Recognition" (VIPeR) [11], "Context Aware Vision using Image-based Active Recognition for Re-Identification" (CAVIAR4REID) [3], and "Labeled Faces in the Wild" (LFW) [16]. The first two datasets focus on person verification from human body images, while the latter one on face verification. In each experiment, we present results by comparing with classic metric learning (ML) algorithms as well as other state-of-the-art approaches. We demonstrate that our proposed approach significantly outperforms existing works and achieves state-of-the-art results on all datasets. The image features and the code for the learning algorithm used in our experiments are available at http://pikachu.ifp.uiuc.edu/~zhenli3/learndecfunc.

4.1. VIPeR

The VIPeR dataset consists of images from 632 pedestrians with resolution 48×128. For each person, a pair of images are taken from cameras with widely differing views. Viewpoint change of 90 degrees or more as well as huge lighting variations make this dataset one of the most challenging datasets available for human body verification. Example images are shown in Figure 4.1.

We follow [28] to extract high level image features based on simple patch color descriptors. To accelerate the learning process, we further reduce the dimensionality of the final feature representation to 600 using PCA (learned on the training set). We also follow exactly the same setup as in [10, 11, 3]: each time half of the 632 people are selected randomly to form the training set, and the remaining people are left for testing (so that no people will appear in both the training and testing). The cumulative matching characteristic (CMC) curve, an estimate of the expectation of finding the correct match in the top n matches, is calculated on the testing set to measure the verification performance (see [10] for details on computing the CMC curve). The final results are averaged over ten random runs.

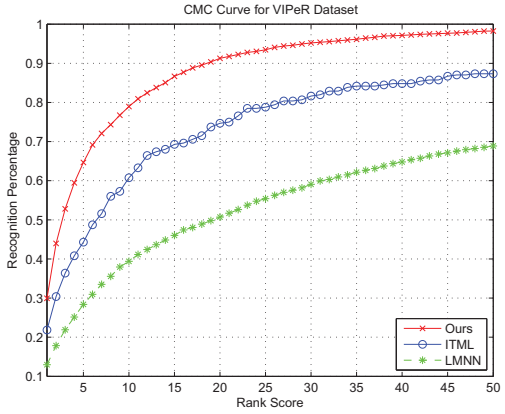
Figure 4. Example images of VIPer dataset. Each column shows two images of the same pedestrian captured under two different cameras.

Figure 5(a) compares our proposed method with classic ML algorithms: LMNN [24] and ITML [5], using the same feature. It is apparent that, in the verification problem, the optimal second-order decision function (4) does significantly improve over traditional ML approaches with a fixed threshold (2). Note that here LMNN performs the worst. One of possible reason is that each class contains only two examples with huge intra-class variations. We are also interested in comparing with other state-of-the-art methods on this dataset, though different features and/or learning algorithms have been used. Figure 5(b) shows the comparison with PS [3], SDALF [8], ELF [11], and PRSVM [20]. Clearly our method outperforms all previous works and achieves state-of-art performance.

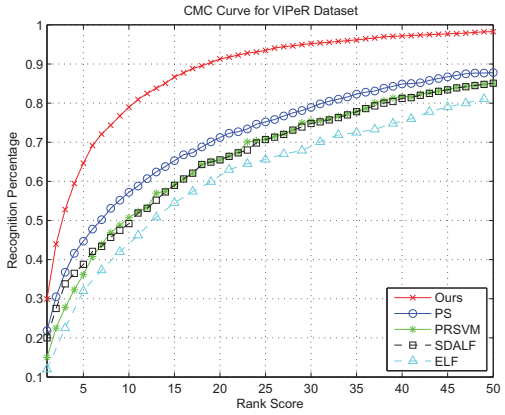
4.2. CAVIAR4REID

CAVIAR4REID [3], extracted from the CAVIAR dataset, is another famous dataset widely used for person verification tasks. This dataset not only covers a wide range of poses and real surveillance footage, but also includes multiple images per pedestrian with different view angles and resolutions. There are in total 72 pedestrians, and each person has images recorded from two different cameras in an indoor shopping mall in Lisbon. All the human body images have been cropped with respect to the ground truth, and the resolution various from 17×39 to 72×144. Here we extract the same image feature as in Section 4.1, and we also use the same training/testing protocol.

Again, we compare with popular ML algorithms as well

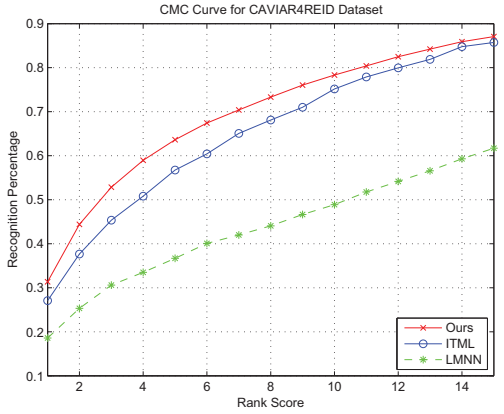


(a) Comparison with metric learning algorithms.

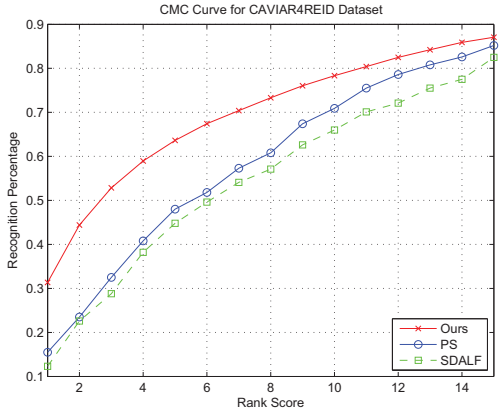


(b) Comparison with other state-of-art algorithms.

Figure 5. Experimental results on VIPeR dataset.



(a) Comparison with metric learning algorithms.



(b) Comparison with other state-of-art algorithms.

Figure 6. Experimental results on CAVIAR4REID dataset.[2]

as other state-of-the-art approaches, as shown in Figure 6(a) and Figure 6(b), respectively. Similarly as in Section 4.1, we observe a substantial improvement over traditional ML algorithms, and our method also outperforms state-of-the-art works including PS [3] and SDALF [8]. It should be noted that, the curves by both PS and SDALF shown in 6(b) have been extrapolated for the sake of fair comparison. The reason is that we have to separate a subset of 36 people for learning the parameters of our decision function (4) or distance metric. With only half of the people left in testing, we rescale the horizontal axis of PS and SDALF by 50% for a fair comparison.

4.3. LFW

The "Labeled Faces in the Wild" (LFW) [16] is a database of face images designed for studying the problem of unconstrained face recognition. The face images were downloaded from Yahoo! News in 2002–2003, and demonstrate a large variety of pose, expression, lighting, etc. The dataset contains more than 13,000 face images from 5,749 people, among which 1,680 people have two or more distinct photos. We extract the same high level image feature as in Section 4.1 and 4.2, except that SIFT [18] descriptors are computed for local patches instead of color, as suggested by [12]. The features are reduced to 500 dimensions using PCA.

We test our algorithm under the standard "image restricted" setting that is particularly designed for verification. In this setting, the dataset is divided into 10 fully independent folds, and it is ensured that not the same person appears across different folds. The identities of the people are hidden from use; instead, 300 positive and 300 negative image pairs are provided within each fold. Figure 3 shows some examples of positive and negative image pairs. Each time we learn both the PCA projection and the parameters of our decision function on 9 training folds, and evaluate on the remaining fold. Pairwise classification accuracy averaged over 10 runs is reported, as suggested by [16].

As shown in Table 1, our approach significantly outperforms state-of-the-art works on the LFW dataset. It should be noted that our verification accuracy of 89.6% outperfoms

[2]We have rescaled the curves by PS [3] and SDALF [8] for a fair comparison. See text.

Table 1. Comparison with state-of-the-art algorithms on LFW dataset. The best performance is highlighted in bold.

Methods	Accuracy (%)
MERL+Nowak [14]	76.2
LDML [12]	79.3
LBP + CSML [19]	85.6
CSML + SVM [19]	88.0
Combined b/g samples [25]	86.8
DML-eig combined [27]	85.7
Deep Learning combined [15]	87.8
Our method	**89.6**

the best reported results[3] in LFW under the category of "no outside data is used beyond alignment/feature extraction". This result also significantly improves our previous work in [17].

5. Conclusion

In this paper, we propose to learn a decision function for the verification problem. Our second-order formulation generalizes from traditional metric learning (ML) approaches by offering a locally adaptive decision rule. Compared with existing approaches including ML, our approach demonstrates state-of-the-art performance on several person verification benchmark datasets such as VIPeR, CAVIAR4REID, and LFW.

References

[1] C. Burges. *Advances in kernel methods: support vector learning*. The MIT press, 1999. 2, 3, 5

[2] X. Chen, Z. Tong, H. Liu, and D. Cai. Metric learning with two-dimensional smoothness for visual analysis. In *CVPR*, pages 2533–2538, 2012. 2

[3] D. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, 2011. 6, 7

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 4, 5

[5] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *ICML*, 2007. 6

[6] M. Dikmen, E. Akbas, T. Huang, and N. Ahuja. Pedestrian recognition with a learned metric. In *ACCV*, 2010. 2

[7] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005. 5

[8] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010. 6, 7

[9] N. Gilardi and S. Bengio. Local machine learning models for spatial data analysis. *Journal of Geographic Information and Decision Analysis*, 4(1):11–28, 2000. 2

[10] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, 2007. 6

[11] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008. 6

[12] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, 2009. 2, 7, 8

[13] C. Huang, S. Zhu, and K. Yu. Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. Technical report, NEC, 2012. 6

[14] G. Huang, M. Jones, E. Learned-Miller, et al. Lfw results using a combined nowak plus merl recognizer. In *Workshop on Faces in Real-Life Images at ECCV*, 2008. 8

[15] G. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR*, pages 2518–2525, 2012. 8

[16] 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. 4, 6, 7

[17] Z. Li, L. Cao, S. Chang, J. R. Smith, and T. S. Huang. Beyond mahalanobis distance: Learning second-order discriminant function for people verification. In *CVPR Workshops*, pages 45–50, 2012. 8

[18] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. 7

[19] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, 2010. 2, 8

[20] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004. 6

[21] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998. 5

[22] J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *CVPR*, 2011. 2

[23] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, 2007. 5

[24] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 6

[25] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, 2009. 8

[26] L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, 2006. 2

[27] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, 13:1–26, 2012. 8

[28] X. Zhou, N. Cui, Z. Li, F. Liang, and T. S. Huang. Hierarchical gaussianization for image classification. In *ICCV*, 2009. 2, 6

[3] http://vis-www.cs.umass.edu/lfw/results.html