Information Fusion 30 (2016) 80–90

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

MetricFusion: Generalized metric swarm learning for similarity measure

Lei Zhang[a,b,*], David Zhang[b]

[a] College of Communication Engineering, Chongqing University, Chongqing 400044, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Article history:
Received 11 June 2015
Revised 3 December 2015
Accepted 14 December 2015
Available online 21 December 2015

Keywords:
Metric learning
Face verification
Similarity
Classification

ABSTRACT

Learning distance metrics for measuring the similarity between two data points in unsupervised and supervised pattern recognition has been widely studied in unconstrained face verification tasks. Motivated by the fact that enforcing single distance metric learning for verification via an empirical score threshold is not robust in uncontrolled experimental conditions, we therefore propose to obtain a metric swarm by learning local patches alike sub-metrics simultaneously that naturally formulates a generalized metric swarm learning (GMSL) model with a joint similarity score function solved by an efficient alternative optimization algorithm. Further, each sample pair is represented as a similarity vector via the well-learned metric swarm, such that the face verification task becomes a generalized SVM-alike classification problem. Therefore, the verification can be enforced in the represented metric swarm space that can well improve the robustness of verification under irregular data structure. Experiments are preliminarily conducted using several UCI benchmark datasets for solving general classification problem. Further, the face verification experiments on real-world LFW and PubFig datasets demonstrate that our proposed model outperforms several state-of-the-art metric learning methods.

1. Introduction

Visual categorization refers to deciding the images that describe the same or similar object into one class. Slightly different, visual identification tries to answer whether or not two images depict the same or similar object from some class. More specifically, face verification as a binary classification problem over pair-wise faces (i.e. face pairs), requires that human or machine can answer whether a pair of face images depict the same person or not. In other words, given two images, both containing faces taken under natural conditions (i.e. in the wild), our goal is to answer: are the two images depicting the same person, or not? In recent years, a challenging benchmark dataset for this problem, i.e. Labeled Faces in the Wild (LFW) released by Huang et al. [1] has been widely exploited and explored by world-wide researchers from computer vision community [2–4]. LFW is for unconstrained face verification, which exhibits appearance variations due to the uncontrolled settings, including variations in scale, pose, background, illumination, and also different attributes like hairstyle, expression, clothing, focus, image resolution, color saturation, etc. Some similar/dissimilar sample-pair examples in LFW are shown in Fig. 1.

Throughout the recent works for face verification, metric learning has received a lot of attention [5–8]. Metric learning provides a fundamental prospective for answering the following question in pattern recognition: how to measure the similarity/dissimilarity between two data points? Most metric learning methods try to learn a Mahalanobis distance metric $\mathbf{M}$ by means of a labeled training set (for classification problems) or from sets of positive (similar) and negative (dissimilar) pairs (for verification problems) based on a predefined objective function. This objective function is usually designed to punish the large distance between similar pairs and the small distance between dissimilar pairs [1,9–11]. The distance metric function of a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ can be written as $(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$. In addition to learning distance metric, similarity metric learning approaches such as cosine similarity metric [7] and bilinear function similarity metric [8] have also been explored. Among them, learning Mahalanobis distance metric for k-nearest neighbor classification is widely reported and of research interest, like information theoretic metric learning (ITML) [9], large margin nearest neighbor (LMNN) [11] which extends the doublets to triplets in constraint for desired distance metric, and scalable large margin Mahalanobis metric learning

* Corresponding author at: College of Communication Engineering, Chongqing University, Chongqing 400044, China. Tel.: +86 13629788369.
E-mail addresses: leizhang@cqu.edu.cn (L. Zhang), csdzhang@comp.polyu.edu.hk (D. Zhang).

http://dx.doi.org/10.1016/j.inffus.2015.12.004
1566-2535/

Fig. 1. Some real-world face pairs of LFW: same pairs (first row) and not same pairs (second row).

[12,13]. Recently, a kernel based distance metric learning method (doublet-SVM vs. triplet-SVM) solved by SVM tool is proposed [14]. Extensions of metric learning have been utilized for multi-task learning [15], semi-supervised learning [16,17], nonlinear distance similarity learning [18], etc.

Despite of these metric learning methods, there is a common property that all try to learn a single distance metric for classification or verification by finding a discriminative score threshold. The weakness may lie in two folds: first, learning single metric is in fact not robust due to the variation of data structure, like LFW; second, decide whether two face images are the same or not through a score threshold is not stable. The reason holds the same as the first, though it is at a glance. Multi-view learning [19–21] and multiple instance learning [22,23] have been proposed in machine learning community. Multi-metric learning has also been proposed for face recognition [24] and face/kinship verification [25]. Even with the fact that learning multiple distance metrics for face recognition can improve the performance, but only Mahalanobis distance metrics are considered, such that the diversity of *similar* metrics is missed based on only distance and the function of multiple metrics is prohibited. In [8], a similarity metric learning method was proposed coupled with distance metric in objective function, which exhibits an improvement for face verification. However, the contribution between similarity and distance metric is treated identically, without appropriate optimization. Moreover, in identification, it finds a discriminative score threshold which holds the same way as other metric learning methods. As a result, the discriminative ability and the potential for similarity measure may be limited.

With above considerations, in this paper we present generalized metric swarm learning for classification and face verification, nominated as GMSL. The goal of GMSL is to learn each local diagonal patch of the metric *swarm* **M** by enforcing with different function. Besides, we propose to map the pair-wise samples into a discriminative metric swarm space with vectors by using the local patches of sub-metrics, such that the standard SVM can be used for final verification. We demonstrate the effectiveness of our framework on several UCI benchmark datasets, real-world LFW and PubFig face datasets. Fig. 2 illustrates the framework of our GMSL for unconstrained face verification.

Our GMSL method is related to the previous method in [8], and there is clear novelty by comparing to [8] and the pre-existing methods. Specifically, the main contributions of this paper are three folds:

(1) Multiple metrics with weights optimization for information sharing are integrated into an optimal metric, such that the similarity/dissimilarity between pairwise samples can be correctly measured. However, in [8], two metrics are combined without recognizing their importance.
(2) Rather than learning two different metrics for similarity score computation in [8], the proposed method aims at learning an optimal metric swarm for representing pairwise samples via a generalized framework, such that the matching problem can be solved by a binary classifier (e.g. SVM).
(3) In [8], the metrics are learned based on one feature type, which limits the capability of the learned metrics. Therefore, in this paper, we also consider learn each metric with/without data centralization.

The rest of this paper is organized as follows. In Section 2, we review the most related works of metric learning. The proposed GMSL with problem formulation and dual optimization is presented in Section 3. The experiments on several UCI benchmark datasets for classification problem, and the experiments for unconstrained face verification on LFW and PubFig data are conducted in Section 4. Finally, Section 5 concludes the paper.

2. Related work

In supervised and unsupervised pattern recognition, metric learning has received much attention in computer vision. As state-of-the-art methods, Weinberger et al. introduced a LMNN method that learns a distance metric **M** to improve the *k*-nearest neighbor classifier [11]. The key point is that it encourages target neighbors to be at least one distance unit closer than points from other classes; therefore, it requires labeled triplets (i, j, k), where data point $\mathbf{x}_j$ is a neighbor of data point $\mathbf{x}_i$, but data point $\mathbf{x}_k$ is not.

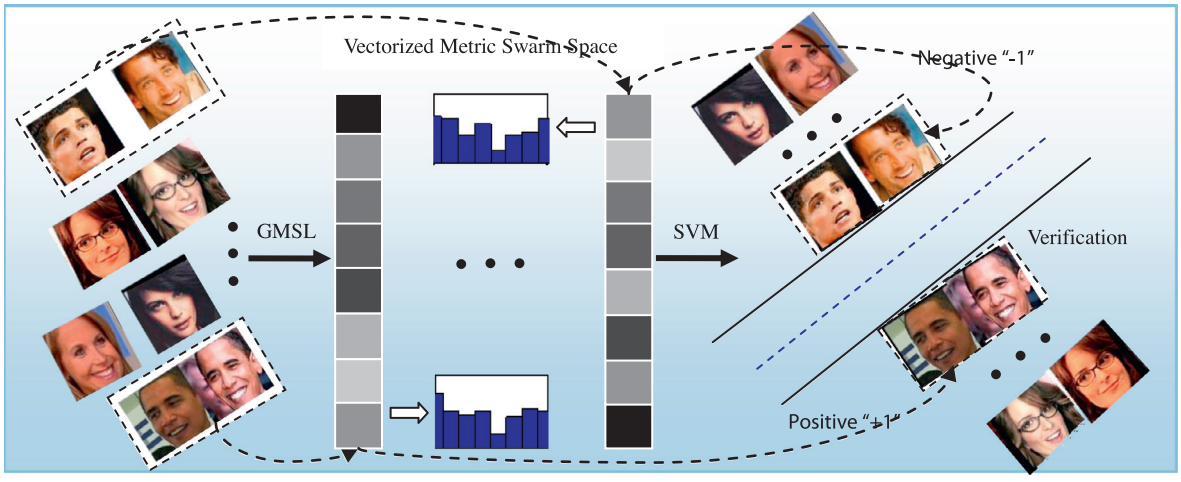

Fig. 2. Illustration of our GMSL for face verification.

LMNN can be written as the following:

$$\min_{\mathbf{M},\xi} \xi_{ijk} + \gamma \sum_{i,j} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$$
$$s.t.\ d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k) - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijk},\ \mathbf{M} \succ 0,\ \xi_{ijk} \geq 0$$

Due to that triplet set is required in LMNN, the restricted setting in LFW is difficult to implement based on doublets, except for the unrestricted setting. In addition, ITML proposed by Davis et al. [9] is another state-of-the-art, which learns a distance metric $\mathbf{M}$ by regularizing it as close as possible to a known prior $\mathbf{M}_0$ on Mahalanobis distance. After LMNN and ITML, Guillaumin et al. [5] proposed a logistic discriminant based metric learning (LDML) which requires that the distance between similar pairs should be smaller than the distance between dissimilar pairs, and try to obtain a probabilistic estimation on whether the two images are similar or not. Taigman et al. [6] proposed a MultiOSS method based on ITML for utilizing class label information in unconstrained face recognition, and outperform large margin based methods.

Up to now, LFW data for unconstrained face verification has attracted rich experience in practice, from metric learning to deep learning. In [26], Kumar et al. presented an attribute and simile classifier based on attribute features which obtained 85.29% accuracy. In [7], Nguyen et al. proposed a cosine similarity metric learning (CSML) and reach an accuracy of 88%. In [24], Cui et al. proposed a SAFR-PMML method which obtained a high accuracy of 89.35% by fusing multiple robust visual descriptors. In [25], Hu et al. proposed a novel large margin multi-metric learning (LM^3L) method for face and kinship verification, which aims at maximizing the correlation of multiple feature representation. Recently, Cao et al. proposed a similarity metric learning (SubSML) [8] and achieved 89.73% accuracy by combing multiple low-level feature descriptors in restricted setting. Also, a deep learning framework for metric learning (DDML) [27] which aims to learn two layered deep features for distance metric using gradient descent algorithm was proposed for face verification, and obtains the best accuracy 90.68% on LFW based on multiple features.

3. Generalized metric swarm learning approach

3.1. Notations

The sets of similar and dissimilar pairs are denoted as $\mathbf{S}$ and $\mathbf{D}$, respectively. Assume that there are G latent metrics defined as $\mathbf{M}_g \in \Re^{d\times d}$ $(g = 1, \cdots\cdots, G)$ which are used to measure the similarity of a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ using a joint metric swarm function $f_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{M}$ represents the metric swarm and $\mathbf{x}_i \in \Re^d$ (d is the dimension of each sample). The metric function of a single $\mathbf{M}_g$ is denoted as $F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j)$. The loss function is defined as $L(\cdot)$ and the Lagrange function is defined as $\Gamma(\cdot)$. In this paper, $\|\cdot\|_F$ and $\|\cdot\|_2$ denote Frobenius norm and ℓ_2-norm.

3.2. Formulation of GMSL

In our proposed GMSL model, we would like to learn the joint metric function implied in the data, that is established based on some predefined metric swarm $\mathbf{M}$ consisting of G different sub-metrics, i.e. $\mathbf{M}_1, \mathbf{M}_2, \cdots\cdots, \mathbf{M}_G$. Suppose a sample set $\mathbf{V}$ consisting of a similar set $\mathbf{S}$ and a dissimilar set $\mathbf{D}$ (i.e. $\mathbf{V} = \mathbf{S} \cup \mathbf{D}$), the proposed joint metric (score) function of a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ is summarized as following definitions:

Definition 1. *$\forall i,j \in \mathbf{V}$, the proposed joint similarity score function $f_{\mathbf{M}}$ of a pair $(\mathbf{x}_i, \mathbf{x}_j)$ under the metric swarm $\mathbf{M} \leftarrow \{\mathbf{M}_1, \mathbf{M}_2, \cdots\cdots, \mathbf{M}_G\}$, can be written as follows:*

$$f_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{h}^{\mathrm{T}} F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

where $F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = [F^1_{\mathbf{M}_1}(\mathbf{x}_i, \mathbf{x}_j), \cdots\cdots, F^G_{\mathbf{M}_G}(\mathbf{x}_i, \mathbf{x}_j)]^{\mathrm{T}} \in \Re^G$ denotes the multi-metric distance function with MetricFusion, $\mathbf{h} = [h_1, \cdots, h_G]^{\mathrm{T}} \in \Re^G$, and $F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j)$ denotes the sub-metric distance function with $\mathbf{M}_g$. The indicator vector $\mathbf{h}$ is a known vector for each sub-metric, which is shown in Definition 2. Note that the arrow $\leftarrow$ denotes that $\mathbf{M}$ can be represented by each sub-metric of $\{\mathbf{M}_1, \mathbf{M}_2, \cdots\cdots, \mathbf{M}_G\}$ via some specific form in application (e.g. Eq(4)).

Definition 2. *$\forall i,\ j \in \mathbf{S},\ i,\ k \in \mathbf{D},\ g \in \{g | g = 1, \cdots\cdots, G\}$, there is*

$$h_g = \begin{cases} -1, & \text{if } F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j) \text{ negatively change indistancefunction} \\ +1, & \text{if } F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j) \text{ positively change indistancefunction} \end{cases} \tag{2}$$

Specifically, Definition 2 can be described as follows: the indicator $h_g = 1$ if the distance function $F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j)$ under sub-metric $\mathbf{M}_g$ increase with the increasing similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$; and $h_g = -1$, if the distance function $F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j)$ decrease with the increasing similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. For example, for Euclidean distance metric $\mathbf{M}$, the indicator $h = -1$, due to that the $F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ of a similar pair is smaller than that of a dissimilar pair. Therefore, how to set h_g value depends on whether the selected sub-metric $\mathbf{M}_g$ has positive (+1) or negative (−1) property on the distance function from a qualitative level. Note that the indicator vector $\mathbf{h}$ does not imply the quantitative weight vector of the proposed MetricFusion. Instead, it implies the property of each sub-metric.

Based on Definitions 1 and 2, the primal problem of the proposed GMSL is formulated as

$$\min_{\mathbf{M}_g, \forall g} L(\mathbf{M}) + \frac{\gamma}{2} \sum_{g=1}^{G} \Theta_g \left\| \mathbf{M}_g - \tilde{\mathbf{M}}_g \right\|_{\mathrm{F}}^2 \tag{3}$$

where $L(\cdot)$ is the loss function, $\tilde{\mathbf{M}}_g$ is a known matrix that is similar to the prior matrix in ITML, $0 < \Theta_g < 1$ is the contribution coefficient for the gth sub-metric, and γ is the regularization parameter.

For clear understanding the metric swarm $\mathbf{M}$, it is shown as a large metric represented by G sub-metrics (diagonal patch) as

$$\mathbf{M} = \begin{bmatrix} h_1\mathbf{M}_1 & & \\ & \ddots & \\ & & h_G\mathbf{M}_G \end{bmatrix} \in \Re^{Gd\times Gd} \tag{4}$$

Note here that the target metric swarm $\mathbf{M}$ denotes a large metric constructed by local patches of sub-metrics. From model (3), it can be observed that we tend to learn the G sub-metrics simultaneously instead of $\mathbf{M}$. After optimization of G sub-metrics, the metric swarm can be easily obtained by using (4).

In general, $L(\mathbf{M})$ in (3) is defined as the following *hinge-loss* function in this paper.

$$L(\mathbf{M}) = \sum\nolimits_{(i,j)\in\mathbf{V}=\mathbf{S}\cup\mathbf{D}} (1 - y_{i,j}\mathbf{h}^{\mathrm{T}} F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j))_+ \tag{5}$$

where $y_{i,j}$ denotes the label +1 and −1 for positive and negative pair, respectively.

Minimizing the loss function term constructed by a metric swarm can promise the discriminative ability of the similarity score function with complex data, and $F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = [F^1_{\mathbf{M}_1}(\mathbf{x}_i, \mathbf{x}_j), \ldots, F^G_{\mathbf{M}_G}(\mathbf{x}_i, \mathbf{x}_j)]^{\mathrm{T}}$. The specific formulation of $F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ can be referred in *Remarks*, and a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ will be transformed in other representation (see Eq. (11)) for similarity measure by $\mathbf{M}$ because it is easy to observe that the dimension of $\mathbf{M}$ is not consistent with $(\mathbf{x}_i, \mathbf{x}_j)$. The regularization term is to prevent the learned sub-metric from being distorted, and hence retains the robustness to the variations of data structure.

To solve model (3) coupled with the hinge-loss function term (5), we introduce the slacking variables $\xi_{i,j}(\mathbf{x}_i, \mathbf{x}_j)$, $i,j \in \mathbf{V}$, the proposed model with constraints is reformulated as

$$\begin{aligned}&\min_{\mathbf{M}_g,\forall g} \sum\nolimits_{(i,j)\in\mathbf{V}} \xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j) + \frac{\gamma}{2}\sum_{g=1}^{G}\Theta_g\|\mathbf{M}_g-\tilde{\mathbf{M}}_g\|_{\mathrm{F}}^2\\ &\text{s.t. } y_{i,j}\mathbf{h}^{\mathrm{T}}F_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) \ge 1-\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j),\\ &\quad \xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\ge 0,\ \forall(i,j)\in\mathbf{V}\end{aligned} \tag{6}$$

Direct optimization of the *primal* problem (6) with inequality constraints is difficult, so its *dual* problem is solved by using Lagrange multiplier method in the following section.

3.3. Optimization of GMSL

To solve the optimal metric swarm $\mathbf{M}$ in (6), we tend to solve its dual problem instead of the primal problem. For detail, the deduction process of its dual problem and the optimization algorithm will be described as follows:

First, we write the Lagrange function $\Gamma(\cdot)$ of (6) as

$$\begin{aligned}\Gamma(\mathbf{M},\xi;\alpha,\beta) = &\sum\nolimits_{(i,j)\in\mathbf{V}}\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j) + \frac{\gamma}{2}\sum_{g=1}^{G}\Theta_g\left\|\mathbf{M}_g-\tilde{\mathbf{M}}_g\right\|_{\mathrm{F}}^2\\ &-\sum\nolimits_{(i,j)\in\mathbf{V}}\alpha_{i,j}\left(y_{i,j}\mathbf{h}^{\mathrm{T}}F_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)-1+\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\right)\\ &-\sum\nolimits_{(i,j)\in\mathbf{V}}\beta_{i,j}\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\end{aligned} \tag{7}$$

where $\alpha \ge 0$ and $\beta \ge 0$ denote the Lagrange multipliers, and it implies that $\alpha_{i,j}+\beta_{i,j}=1$ in subsequent dual analysis.

By calculating the partial derivatives of $\Gamma(\mathbf{M}, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta})$ in (7) with respect to $\mathbf{M}_g$ and $\xi_{i,j}(\mathbf{x}_i, \mathbf{x}_j)$, one can have

$$\begin{cases}\frac{\partial\Gamma(\mathbf{M},\xi;\alpha,\beta)}{\partial\mathbf{M}_g} = \gamma\,\Theta_g\left(\mathbf{M}_g-\tilde{\mathbf{M}}_g\right) - \sum\limits_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}h_g\frac{\partial F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)}{\partial\mathbf{M}_g}\\ \frac{\partial\Gamma(\mathbf{M},\xi;\alpha,\beta)}{\partial\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)} = 1-\alpha_{i,j}-\beta_{i,j}\end{cases} \tag{8}$$

Let $\frac{\partial\Gamma(\mathbf{M},\xi;\alpha,\beta)}{\partial\mathbf{M}_g} = \frac{\partial\Gamma(\mathbf{M},\xi;\alpha,\beta)}{\partial\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)} = 0$, one can easily obtain the expression of sub-metric $\mathbf{M}_g$ as follows:

$$\begin{cases}\mathbf{M}_g = \tilde{\mathbf{M}}_g + \frac{h_g}{\gamma\,\Theta_g}\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\frac{\partial F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)}{\partial\mathbf{M}_g}\\ \beta_{i,j} = 1-\alpha_{i,j}\end{cases} \tag{9}$$

Substituting $\mathbf{M}_g$ and $\beta_{i,j}$ back into $\Gamma(\mathbf{M}, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta})$ in (7), there is the following expression:

$$\begin{aligned}&\Gamma(\mathbf{M},\xi;\alpha,\beta)\\ &= \sum\nolimits_{(i,j)\in\mathbf{V}}\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j) + \frac{\gamma}{2}\sum_{g=1}^{G}\Theta_g\left\|\frac{1}{\gamma\,\Theta_g}\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\frac{\partial F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)}{\partial\mathbf{M}_g}\right\|_{\mathrm{F}}^2\\ &\quad-\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{h}^{\mathrm{T}}F_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) + \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} - \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\\ &\quad-\sum_{(i,j)\in\mathbf{V}}\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j) + \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}\xi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} + \frac{1}{2\gamma}\sum_{g=1}^{G}\frac{1}{\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\frac{\partial F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)}{\partial\mathbf{M}_g}\right\|_{\mathrm{F}}^2\\ &\quad-\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{h}^{\mathrm{T}}F_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)\end{aligned} \tag{10}$$

Note that $h_g^2 = 1$ (see Definition 2 in this paper).

For easily solving the proposed metric swarm $\mathbf{M}$ involved in the optimization problem, we expect $F_{\mathbf{M}_g}^g(\mathbf{x}_i, \mathbf{x}_j)$ to be strictly convex with respect to $\mathbf{M}_g$, such that the variable $\mathbf{M}_g$ can be eliminated in $\frac{\partial F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)}{\partial\mathbf{M}_g}$. Therefore, the following expression of $F_{\mathbf{M}_g}^g(\mathbf{x}_i, \mathbf{x}_j)$ is introduced:

$$F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j) = \mathbf{u}_g^{\mathrm{T}}\mathbf{M}_g\mathbf{v}_g, \forall g = 1,\ldots,G \tag{11}$$

where $\mathbf{u}_g$ and $\mathbf{v}_g$ are vectors formed by a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ (see the following Theorem 1 in this paper).

According to (11), we get that

$$\frac{\partial F_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)}{\partial\mathbf{M}_g} = \mathbf{u}_g\mathbf{v}_g^{\mathrm{T}} \tag{12}$$

By substituting (12) back into (9), we can obtain

$$\mathbf{M}_g = \tilde{\mathbf{M}}_g + \frac{h_g}{\gamma\,\Theta_g}\sum\nolimits_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}} \tag{13}$$

So, we can rewrite (10) step by step as follows:

$$\begin{aligned}&\Gamma(\mathbf{M},\xi;\alpha,\beta)\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} + \frac{1}{2\gamma}\sum_{g=1}^{G}\frac{1}{\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &\quad-\sum\nolimits_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\sum\nolimits_{g=1}^{G}h_gF_{\mathbf{M}_g}^g(\mathbf{x}_i,\mathbf{x}_j)\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} + \frac{1}{2\gamma}\sum_{g=1}^{G}\frac{1}{\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &\quad-\sum\nolimits_{g=1}^{G}h_g\left(\sum\nolimits_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\left(\mathbf{u}_g^{\mathrm{T}}\mathbf{M}_g\mathbf{v}_g\right)\right)\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} + \frac{1}{2\gamma}\sum_{g=1}^{G}\frac{1}{\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &\quad-\sum_{g=1}^{G}h_g\left(\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\left(\mathbf{u}_g^{\mathrm{T}}\left(\tilde{\mathbf{M}}_g+\frac{h_g}{\gamma\,\Theta_g}\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right)\mathbf{v}_g\right)\right)\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} + \frac{1}{2\gamma}\sum_{g=1}^{G}\frac{1}{\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &\quad-\sum_{g=1}^{G}\left(h_g\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g^{\mathrm{T}}\tilde{\mathbf{M}}_g\mathbf{v}_g\right)\\ &\quad-\sum_{g=1}^{G}\frac{1}{\gamma\,\Theta_g}\left(\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g^{\mathrm{T}}\left(\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right)\mathbf{v}_g\right)\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} + \frac{1}{2\gamma}\sum_{g=1}^{G}\frac{1}{\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &\quad-\sum_{g=1}^{G}\left(h_g\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g^{\mathrm{T}}\tilde{\mathbf{M}}_g\mathbf{v}_g\right)-\sum_{g=1}^{G}\frac{1}{\gamma\,\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &= \sum_{(i,j)\in\mathbf{V}}\alpha_{i,j} - \sum_{g=1}^{G}\frac{1}{2\gamma\,\Theta_g}\left\|\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2\\ &\quad-\sum_{g=1}^{G}\left(h_g\sum_{(i,j)\in\mathbf{V}}\alpha_{i,j}y_{i,j}\mathbf{u}_g^{\mathrm{T}}\tilde{\mathbf{M}}_g\mathbf{v}_g\right)\end{aligned} \tag{14}$$

For further simplifying the Lagrange function (14), as claimed before, $\tilde{\mathbf{M}}_g$ is a defined as a known matrix. Therefore, in this paper,

we define a diagonal matrix $\tilde{\mathbf{M}}_g = \delta_g \mathbf{I}$. Note that $0 \le \delta_g \le 1$ and $\mathbf{I}$ is an identity matrix. Then expression (14) can be written as

$$\begin{aligned}
&\Gamma(\mathbf{M}, \xi; \alpha, \beta) \qquad (15)\\
&= \sum_{(i,j)\in\mathbf{V}} \alpha_{i,j} - \sum_{g=1}^{G} \frac{1}{2\gamma\,\Theta_g} \left\| \sum_{(i,j)\in\mathbf{V}} \alpha_{i,j} y_{i,j} \mathbf{u}_g \mathbf{v}_g^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \\
&\quad - \sum_{g=1}^{G} \left(\delta_g h_g \sum_{(i,j)\in\mathbf{V}} \alpha_{i,j} y_{i,j} \mathbf{u}_g^{\mathrm{T}} \mathbf{v}_g \right) \\
&= \sum_{(i,j)\in\mathbf{V}} \alpha_{i,j} - \sum_{g=1}^{G} \left(\frac{1}{2\gamma\,\Theta_g} \left\| \sum_{(i,j)\in\mathbf{V}} \alpha_{i,j} y_{i,j} u_g v_g^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \right. \\
&\quad \left. + \delta_g h_g \sum_{(i,j)\in\mathbf{V}} \alpha_{i,j} y_{i,j} \mathbf{u}_g^{\mathrm{T}} \mathbf{v}_g \right) \qquad (15)
\end{aligned}$$

Specifically, the Lagrange dual problem of GMSL formulation (6) is summarized as the following theorem.

Theorem 1. *With the prerequisites* $F_{\mathbf{M}_g}^{g}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{u}_g^{\mathrm{T}} \mathbf{M}_g \boldsymbol{v}_g$, $\forall g = 1, \ldots, G$ *and* $\tilde{\mathbf{M}}_g = \delta_g \mathbf{I}$, $0 \le \delta_g \le 1$, *the dual formulation of our GMSL model (6) can be written as*

$$\max_{0\le\alpha\le 1} \sum\nolimits_{(i,j)\in\mathbf{V}} \alpha_{i,j} - \sum_{g=1}^{G} \left(\begin{array}{c} \frac{1}{2\gamma\,\Theta_g} \left\| \sum\nolimits_{(i.j)\in V} \alpha_{i,j} y_{i,j} \mathbf{u}_g \mathbf{v}_g^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \\ + \delta_g h_g \sum\nolimits_{(i.j)\in V} \alpha_{i,j} y_{i,j} \mathbf{u}_g^{\mathrm{T}} \mathbf{v}_g \end{array} \right) \qquad (16)$$

where $\mathbf{u}_g$ *and* $\mathbf{v}_g$ *can be, but not limited to the following cases:*

a. $\mathbf{u}_1 = \mathbf{x}_i - \mathbf{x}_j$, $\mathbf{v}_1 = \mathbf{x}_i - \mathbf{x}_j$
b. $\mathbf{u}_2 = (\mathbf{x}_i - \bar{\mathbf{x}}_i \circ \vec{1}) - (\mathbf{x}_j - \bar{\mathbf{x}}_j \circ \vec{1})$, $\mathbf{v}_2 = (\mathbf{x}_i - \bar{\mathbf{x}}_i \circ \vec{1}) - (\mathbf{x}_j - \bar{\mathbf{x}}_j \circ \vec{1})$
c. $\mathbf{u}_3 = \mathbf{x}_i$, $\mathbf{v}_3 = \mathbf{x}_j$
d. $\mathbf{u}_4 = \mathbf{x}_i - \bar{\mathbf{x}}_i \circ \vec{1}$, $\mathbf{v}_4 = \mathbf{x}_j - \bar{\mathbf{x}}_j \circ \vec{1}$

where $\vec{1}$ *is a full one vector,* $\bar{\mathbf{x}}$ *denote the mean of vector* $\mathbf{x}$ *and* $\circ$ *denotes element-wise multiplication. It is clear that common Mahalanobis distance metric is considered in case a, and it is implied in case b with data centralization; similarity metric is used in case c and d via bilinear function.*

Note also that we require $F_{\mathbf{M}_g}^{g}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{u}_g^{\mathrm{T}} \mathbf{M}_g \mathbf{v}_g$ because $\frac{\partial F_{\mathbf{M}_g}^{g}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{M}_g} = \mathbf{u}_g \mathbf{v}_g^{\mathrm{T}}$ is constant which makes the optimization easier. Also, we set the known matrix $\tilde{\mathbf{M}}_g = \delta_g \mathbf{I}$ as a diagonal matrix for avoiding the distortion of the learned metric swarm, and retain the robustness. It is obvious that formulation (16) in Theorem 1 is a standard quadratic programming (SQP) problem and it can be easily solved by QP solvers. However, with increasing number of pairs in training set, the QP solvers with interior point methods may lose efficiency. Therefore, we follow the solving algorithm based on the accelerated first order algorithm implemented in [8,28] in this paper.

After obtaining the optimal solution $\boldsymbol{\alpha}*$, based on Karush-Kuhn-Tucker (KKT) condition, the learned sub-metric $\mathbf{M}_g$ $(g = 1, \ldots, G)$ can be obtained as

$$\mathbf{M}_g^* = \delta_g \mathbf{I} + \frac{h_g}{\gamma\,\Theta_g} \sum\nolimits_{(i,j)\in\mathbf{V}} \alpha_{i,j}^* y_{i,j} \mathbf{u}_g \mathbf{v}_g^{\mathrm{T}} \qquad (17)$$

3.4. Solving Θ with model modification

We notice from (16) that there is coefficient Θ in iterations. For finding the coefficient, we propose to solve the following model. Note that for differentiating the Θ in (16), we use ϑ and $\Theta_g \leftarrow \vartheta_g^q$. We impose the constraints $\sum_{g=1}^{G} \vartheta_g = 1$ and $0 < \vartheta_g < 1$. The detail of solving coefficient ϑ is shown as follows. First, model (6) is slightly modified as

$$\begin{aligned}
&\min_{\mathbf{M}_g, \Theta_g, \forall g} \sum\nolimits_{(i,j)\in\mathbf{V}} \xi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) + \frac{\gamma}{2} \sum_{g=1}^{G} \vartheta_g^q \left\| \mathbf{M}_g - \tilde{\mathbf{M}}_g \right\|_{\mathrm{F}}^2 \\
&\text{s.t. } y_{i,j} \mathbf{h}^{\mathrm{T}} F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \ge 1 - \xi_{i,j}(\mathbf{x}_i, \mathbf{x}_j), \\
&\xi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \ge 0, \forall (i,j) \in \mathbf{V}, \sum_{g=1}^{G} \vartheta_g = 1,\ q > 1 \qquad (18)
\end{aligned}$$

We see that q-square ϑ_g^q is used with $q > 1$, this is to avoid the trivial solution like $\Theta_g = 0$ or $\Theta_g = 1$ which would pursue the best metric. However, we would like to utilize the information of all sub-metrics in learning process, which will improve the robustness of model. By setting the partial derivative $\frac{\partial \Gamma(\mathbf{M}_g, \vartheta_g)}{\partial \mathbf{M}_g}$ of the Lagrange Eq. (18) to be 0, we have

$$\begin{cases} \frac{\gamma}{2} q\, \vartheta_g^{q-1} \left\| \mathbf{M}_g - \tilde{\mathbf{M}}_g \right\|_{\mathrm{F}}^2 = \mu \\ \sum_{g=1}^{G} \vartheta_g = 1 \end{cases} \qquad (19)$$

where μ is the Lagrange multiplier. From (19), ϑ_g can be easily solved as

$$\vartheta_g = \left(\frac{1}{\left\| \mathbf{M}_g - \tilde{\mathbf{M}}_g \right\|_{\mathrm{F}}^2} \right)^{1/(q-1)} \Big/ \sum\nolimits_{g=1}^{G} \left(\frac{1}{\left\| \mathbf{M}_g - \tilde{\mathbf{M}}_g \right\|_{\mathrm{F}}^2} \right)^{1/(q-1)} \qquad (20)$$

In experiments, q is set as 2. The initial ϑ_g is set as $1/G$.

We propose to solve the proposed GMSL via an alternative optimization algorithm, shown in Algorithm 1.

3.5. Representation of pairwise samples in metric swarm space

To match the sample pair in terms of SVM, we propose to represent each sample pair as a vector in metric swarm space, such that binary classification-alike can be implemented by SVM. In other words, we want to represent each face pair as a similarity vector with labels +1 and −1 for positive and negative pairs, respectively, such that the verification tasks can be realized with dis-

Algorithm 1 GMSL.

Input:
Similar pairwise sample set **S**;
Dissimilar pairwise sample set **D**;
Indicator vector $\mathbf{h} = [h_1, \ldots, h_G]^{\mathrm{T}}$;
Initialize $\mathbf{M}_g^{(0)}, g = 1, \ldots, G$
Initialize $\vartheta_g^{(0)} \leftarrow 1/G, g = 1, \ldots, G$ and $t \leftarrow 0$;
Procedure:
Repeat
1. Update $\boldsymbol{\alpha}^{(t)}$ by solving the dual optimization problem (16) using FISTA algorithm;
2. Update the sub-metric $\mathbf{M}_g^{(t)} \in \Re^{d\times d}, g = 1, \ldots, G$ in metric swarm by using (17) and $\boldsymbol{\alpha}^{(t)}$;
3. Update $\vartheta_g^{(t)}, g = 1, \ldots, G$;
4. $t \leftarrow t + 1$;
until convergence.
Output: $\mathbf{M}^{(t)} = diag([h_1 \mathbf{M}_1^{(t)}, \ldots, h_G \mathbf{M}_G^{(t)}])$ with (4).

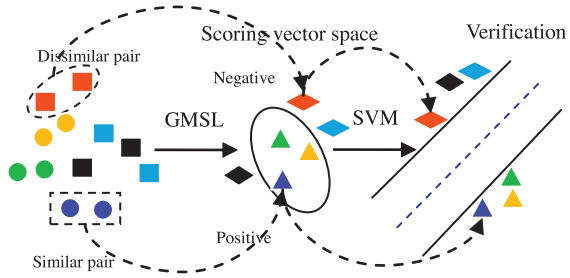

Fig. 3. A diagram of the proposed GMSL for classification. Left: the circles with different colors denote similar pairs, the squares with different colors denote dissimilar pairs; medium: the triangles denote the represented vector space of similar pairs, and the diamonds denote the represented vector space of dissimilar pairs; right: the binary classification in the learned metric space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

criminative classifiers. The advantage is that it can inherit the merits from both metric learning and general classifiers, while avoiding using one simple score threshold of the metric function for verification. Additionally, by learning a discriminative SVM classifier in the represented metric swarm space, the verification tasks will be more robust to the noise, corruption etc. encountered in image acquisition.

Therefore, the similarity vector (*MS*-space) for each sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ based on the learned metrics can be represented as

$$\mathbf{MS}(\mathbf{x}_i, \mathbf{x}_j) = \left[F^1_{\mathbf{M}_1}(\mathbf{x}_i, \mathbf{x}_j), \ldots, F^G_{\mathbf{M}_G}(\mathbf{x}_i, \mathbf{x}_j)\right]^{\mathrm{T}} \in \Re^G \tag{21}$$

3.6. Sample-pair verification

With the learned metric swarm in (17), all face pairs are represented in a similarity vector space according to (21). The labels for positive and negative pairs are set as +1 and −1, respectively. The verification is finally transformed into a standard binary classification problem, which is easily solved by a standard SVM using a general toolbox.

Fig. 2 illustrates our basic framework for face verification. Generally, the proposed method is also adaptive in classification. The diagram of the proposed method in general classification is illustrated in Fig. 3.

In Theorem 1, we have learned four sub-metrics in the metric swarm based on distance and similarity metrics, considering their strict convex property with respect to $\mathbf{M}_g$. For measuring the contribution of each sub-metric, in learning process, we first pretrain the coefficients Θ_g by (20). Then, we fix the coefficients, and use the fast iterative shrinkage thresholding algorithm (FISTA) [28] to solve the dual problem (16). After optimization of the metric swarm, the discriminative space is reconstructed through the metric swarm space representation (21). The sample pairs are then transformed into vectors, respectively, and the verification of "similar" or "not similar" can be done by a standard SVM in this paper. Specifically, the verification process by using GMSL is summarized as Algorithm 2.

3.7. Remarks

3.7.1. Joint metric swarm score function

As shown in Theorem 1, by using the predefined size of the metric swarm $\mathbf{M} \equiv \{\mathbf{M}_1, \ldots, \mathbf{M}_4\}$ with respect to the four cases of *a*, *b*, *c* and *d*, the final joint metric score function $f_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ in (1) with $F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = [F^1_{\mathbf{M}_1}(\mathbf{x}_i, \mathbf{x}_j), \ldots, F^G_{\mathbf{M}_G}(\mathbf{x}_i, \mathbf{x}_j)]^{\mathrm{T}} \in \Re^G$, and $\mathbf{h} = [h_1, \ldots, h_G]^{\mathrm{T}} \in \Re^G$ can be shown as

$$\begin{aligned} f_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{h}^{\mathrm{T}} F_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \\ &= -F_{\mathbf{M}_1}(\mathbf{x}_i, \mathbf{x}_j) - F_{\mathbf{M}_2}(\mathbf{x}_i, \mathbf{x}_j) + F_{\mathbf{M}_3}(\mathbf{x}_i, \mathbf{x}_j) + F_{\mathbf{M}_4}(\mathbf{x}_i, \mathbf{x}_j) \\ &= -\mathbf{u}_1^{\mathrm{T}}\mathbf{M}_1\mathbf{v}_1 - \mathbf{u}_2^{\mathrm{T}}\mathbf{M}_2\mathbf{v}_2 + \mathbf{u}_3^{\mathrm{T}}\mathbf{M}_3\mathbf{v}_3 + \mathbf{u}_4^{\mathrm{T}}\mathbf{M}_4\mathbf{v}_4 \\ &= \left[\mathbf{u}_1^{\mathrm{T}}, \mathbf{u}_2^{\mathrm{T}}, \mathbf{u}_3^{\mathrm{T}}, \mathbf{u}_4^{\mathrm{T}}\right] \underbrace{\begin{bmatrix} -\mathbf{M}_1 & & & \\ & -\mathbf{M}_2 & & \\ & & \mathbf{M}_3 & \\ & & & \mathbf{M}_4 \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \end{bmatrix} \end{aligned} \tag{22}$$

where $\mathbf{h} = [-1, -1, 1, 1]^{\mathrm{T}}$ according to Definition 2, $\mathbf{u}_1, \ldots, \mathbf{u}_4$ and $\mathbf{v}_1, \ldots, \mathbf{v}_4$ are represented in Theorem 1. Here, we know that the value $h_g = 1$ if the score $F^g_{\mathbf{M}_g}(\mathbf{x}_i, \mathbf{x}_j)$ has a positive (ascent) change with the increasing similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, and −1 otherwise. It implies that the GMSL requires the joint score function (22) of a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ to be high if they are similar. From (22), it is clear that the optimization of proposed metric swarm $\mathbf{M}$ is divided into four sub-metrics in diagonal, such that the learned metric with local patches is more robust when the data structure is complex. The proposed GMSL also brings some new prospective to metric learning that the metric $\mathbf{M}$ can be optimized in patches, and each local patch of $\mathbf{M}$ has different impact on the similarity computation of sample pairs.

Algorithm 2 GMSL for verification.

Input:
Similar pairwise sample set **S**;
Dissimilar pairwise sample set **D**;
Parameters γ, δ and Θ.
Procedure:
Step 1. GMSL optimization.
Obtain the metric swarm **M** by using Algorithm 1.
Step 2. Metric swarm space representation.
Represent each pair $(\mathbf{x}_i, \mathbf{x}_j)$ as a similarity vector by using (18) with **M**;
Step 3. Verification.
Train and test a standard SVM in the represented metric swarm space by using 10-fold cross validation.
Output: Verification result.

3.7.2. Optimality condition

We investigate the optimality condition of GMSL by checking the duality gap (*DualityGap*), which is formulated as the difference between the primal objective function (6) and the dual objective function (16) (with the *q*-power on Θ_g) at the *t*-th iteration, i.e.

$$\begin{aligned} DualityGap^{(t)} = & \sum_{(i,j)\in\mathbf{V}} \xi^{(t)}_{i,j}(\mathbf{x}_i, \mathbf{x}_j) + \frac{\gamma}{2}\sum_{g=1}^{G}\left(\Theta_g^{(t)}\right)^q \left\|\mathbf{M}_g^{(t)} - \tilde{\mathbf{M}}_g\right\|_{\mathrm{F}}^2 \\ & - \sum\nolimits_{(i,j)\in\mathbf{V}} \alpha^{(t)}_{i,j} \\ & + \sum_{g=1}^{G}\left(\begin{array}{c} \frac{1}{2\gamma\left(\Theta_g^{(t)}\right)^q}\left\|\sum_{(i,j)\in V}\alpha^{(t)}_{i,j}y_{i,j}\mathbf{u}_g\mathbf{v}_g^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ +\delta_g\mathbf{h}_g\sum_{(i,j)\in V}\alpha^{(t)}_{i,j}y_{i,j}\mathbf{u}_g^{\mathrm{T}}\mathbf{v}_g \end{array}\right) \end{aligned} \tag{23}$$

where the error term is computed as

$$\xi^{(t)}_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = \left[1 - y_{i,j}\mathbf{h}^{\mathrm{T}}F_{\mathbf{M}^{(t)}}(\mathbf{x}_i, \mathbf{x}_j)\right]_+ \tag{24}$$

where y_{ij}=1 for similar pairs, and −1, otherwise. By substituting (22) and (24) into (23), the duality gap can be calculated. For example, the duality gap curve by using LFW data within 10 iterations is shown in Fig. 4, from which, we can observe that GMSL can converge to a global optimum within 10 iterations. The efficiency of GMSL is clearly demonstrated.

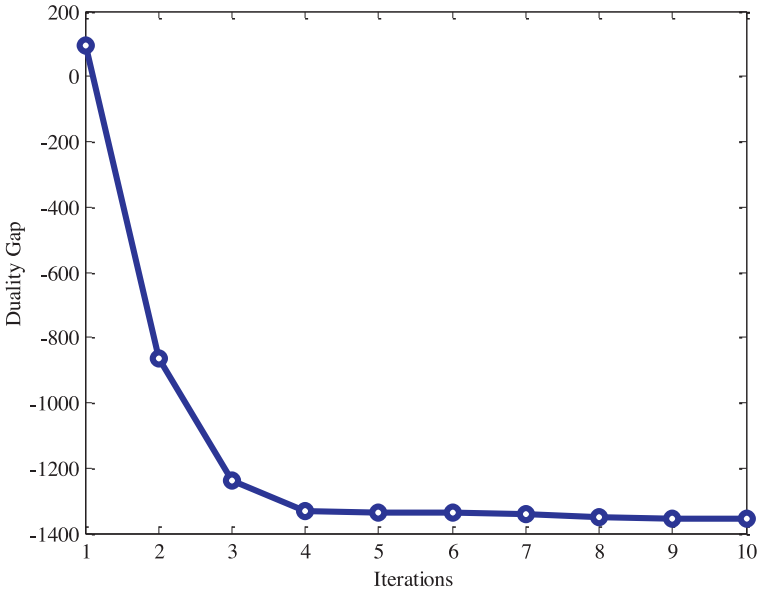

Fig. 4. Duality gap vs. the iterations on LFW data.

4. Experiments

In the experiments, we evaluate the proposed GMSL for classification by using UCI benchmark datasets and unconstrained face verification using LFW and PubFig datasets. We have compared with most popular metric learning methods, including DML-eig [10], LMNN [11], ITML [9], Sparse-ML [29], Sub-SML [8], LDML [5], CSML [7], and deep metric learning method DDML [27].

4.1. Parameter setting

Throughout the paper, the two parameters in GMSL, γ is set as 100 and δ is set as 0.5, respectively, in experiments. The initial parameter θ_g is set as $1/G$.

4.2. Test results on UCI datasets

We first test our proposed GMSL on 8 UCI benchmark datasets from UCI machine learning repository [30] and observe the preliminary effectiveness for general classification problem. The information of the selected 8 benchmark datasets in experiments is described in Table 1.

For each data we use 10-folds cross validation to evaluate the represented metric learning methods, and present the comparisons of the average classification accuracy. Note that for metric learning, the algorithm input should be similar/dissimilar sample pairs.

We have compared the classification error rate with several most popular metric learning methods including DML-eig [10], LMNN [11], ITML [9], Sparse-ML [29], and Sub-SML [8] in Fig. 5. From the bar plots of error rates, we can see that GMSL outperforms other metric learning methods in five datasets. For datasets 5, 7 and 8, GMSL is slightly inferior to ITML. Therefore, for total comparison on all datasets, the average rank of error rate is calculated and illustrated in Fig. 6. We can observe that GMSL has the highest rank with the lowest error rate among the seven metric methods. The results demonstrate the effectiveness of our GMSL in general classification problem.

Table 1
Description of 8 UCI datasets.

Dataset	Feature dimensions	# of classes	# of training samples	# of test samples
Wine	13	3	125	53
Iris	4	3	105	45
SPECTF-Heart	44	2	80	187
Statlog-Heart	13	2	189	81
UserKnowledge	5	4	101	44
ILPD	10	2	525	58
Sonar	60	2	188	20
Seeds	7	3	147	63

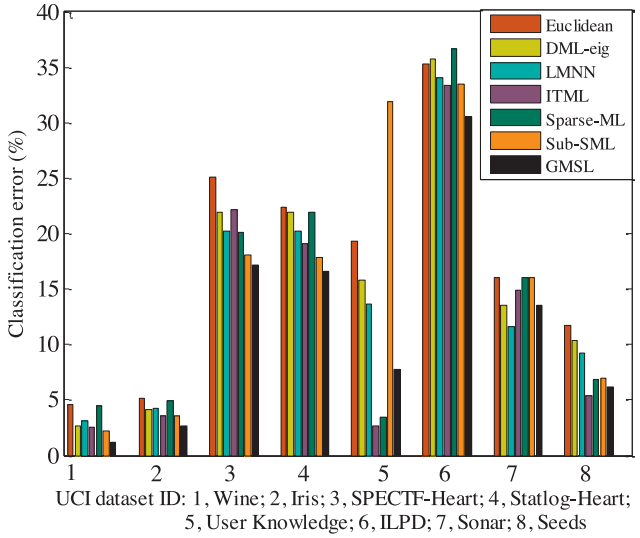

Fig. 5. Classification error (%) of Euclidean, DML-eig, LMNN, ITML, Sparese ML, Sub-SML and the proposed GMSL metric learning methods.

4.3. Test results on LFW faces

LFW (Labeled Faces in the Wild) is commonly regarded as a challenging dataset for unconstrained face verification. It contains 13,233 face images from 5749 persons [1]. Restricted and unrestricted protocols are included in LFW. The only available information in restricted protocol is whether each pair depicts the same subject or not. For image unrestricted protocol, the identity

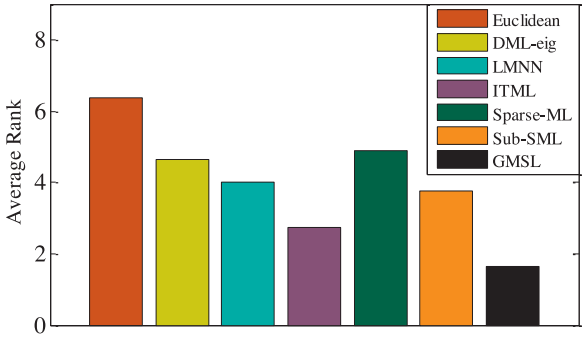

Fig. 6. Average rank of the classification error rate.

information of each face image is known, and extra face images with known identities can also be used. For comparing with most state-of-the-art methods in the same experimental setting, image restricted protocol is adopted in this paper.

We use the LBP and SIFT low-level image features from [8] with 300 feature dimensions after PCA dimension reduction for analysis. Additionally, the attribute feature with 73 attributes like hair style, age, gender, race, etc. from [26] are also used. The performance of a face verification algorithm is evaluated by 10-fold cross validation, with each fold containing 300 positive and 300 negative image pairs. Specifically, for each fold cross-validation, 2700 similar pairs and 2700 dissimilar pairs from 9 folds are used for training, and 300 similar pairs and 300 dissimilar pairs from the remaining fold are used for testing.

To better show the effectiveness of the proposed GMSL method, for each single feature, the verification accuracy (%) by learning each metric $\mathbf{M}_i$ ($i = 1,\ldots,4$) separately have been reported in Table 2, in which *GMSL-Comb* denotes the straightforward combination of the separately learned 4 metrics instead of joint learning. The results show that the proposed GMSL joint learning method outperforms the combined method. Additionally, the comparisons by learning each metric separately based on multiple features have also been conducted and reported in Table 3, from which we can clearly observe the significant improvement by using the proposed GMSL joint learning framework. From Tables 2 and 3, it is clear that learning multiple metrics outperforms that of learning single metric, and the combination of multiple metrics straightforward cannot better exploit the correlation and association among different metrics.

The average accuracy (%) and standard deviation of 10 folds are provided in Table 4, which presents the results of popular metric learning methods on LFW for LBP, SIFT, and Attribute features, respectively. Note that some results are absent because the results were not reported in their previous work. Besides, the LMNN needs the triplet information of image pairs, and it is not compared in this work for face verification. From the results, we can see that the proposed method outperforms other popular methods for LBP, SIFT and Attribute, respectively. We have also experimented by using the joint metric score in Eq. (22) for verification based on a learned threshold. For SIFT feature, the accuracy is 85.6 ± 0.45 in percent; for LBP feature, the accuracy is 87.0 ± 0.52 in percent; for attribute feature, the accuracy is 85.2 ± 0.68 in percent. For further validating the performance with different q-values and make deep insight of the advantage of the proposed MetricFusion, we have studied how q-value performs on the metric weights and face verification in Fig. 7. Fig. 7(a) shows the θ values (weights of metric fusion) for different q-values. Fig. 7(b) shows the performance curve of different q-values for three features considered in this paper. From Fig. 7 (b) the performance curve, it clearly shows the best result of MetricFusion when $q = 2$. The weight of each metric for $q = 2$ is shown in Fig. 7 (a). The results of three features have almost consistent performance variation with increasing q-value. We have also experimented by setting the $\theta = 0.25$ for 4 metrics in average, and the accuracy is 83.9%, 84.9% and 85.2% for SIFT, LBP and Attribute features, respectively, which demonstrates that simply average multiple metrics may not capture the best performance.

Specifically, for comparing with state-of-the-art results on constrained LFW via metric learning methods, we conduct a systematical comparison and provide the best results in Table 5, in which NoD denotes the number of feature descriptors. From Table 5, we observe that with 3 descriptors, the proposed method achieves the highest face verification accuracy of 90.67%, and higher than state-of-the-art 89.73% by Sub-SML [8] with 6 descriptors. By implementing Sub-SML on the same 3 descriptors, 89.33% accuracy is obtained with 1.34% lower than ours. For other methods, SFRD + PMML and ITML + Multiple OSS obtain inferior perfor-

Table 2
Comparisons with baselines of single metric and combined metrics with single feature on LFW dataset.

Single feature	GMSL-$\mathbf{M}_1$	GMSL-$\mathbf{M}_2$	GMSL-$\mathbf{M}_3$	GMSL-$\mathbf{M}_4$	GMSL-Comb	GMSL
LBP	86.10 ± 0.49	86.17 ± 0.46	85.17 ± 0.49	85.08 ± 0.51	86.68 ± 0.46	**87.12 ± 0.41**
SIFT	84.40 ± 0.43	84.38 ± 0.41	82.93 ± 0.28	82.75 ± 0.32	84.85 ± 0.45	**85.98 ± 0.40**
Attribute	84.10 ± 0.59	84.05 ± 0.52	81.78 ± 0.44	79.78 ± 0.79	85.45 ± 0.64	**85.62 ± 0.69**

Table 3
Comparisons with baselines of single metric and combined metrics with multiple features on LFW dataset.

Multiple features	GMSL-$\mathbf{M}_1$	GMSL-$\mathbf{M}_2$	GMSL-$\mathbf{M}_3$	GMSL-$\mathbf{M}_4$	GMSL-Comb	GMSL
LBP + SIFT	87.82 ± 0.53	87.82 ± 0.45	86.73 ± 0.38	86.70 ± 0.38	87.98 ± 0.61	**89.35 ± 0.44**
LBP + Attribute	89.10 ± 0.44	88.98 ± 0.45	87.85 ± 0.38	87.60 ± 0.54	89.40 ± 0.49	**89.45 ± 0.41**
SIFT + Attribute	87.30 ± 0.36	87.23 ± 0.36	85.92 ± 0.52	85.98 ± 0.60	87.95 ± 0.38	**88.20 ± 0.51**
LBP + SIFT + Attribute	89.67 ± 0.35	89.73 ± 0.39	88.22 ± 0.57	88.17 ± 0.60	90.06 ± 0.41	**90.67 ± 0.46**

Table 4
Comparisons with state of the art metric learning methods on LFW dataset.

Method	SILD [32]	ITML [9]	DML-eig [10]	LDML [5]	KissME [37]	CSML [7]	Att.&Sim. classifier [26]	SubSML [8]	**GMSL**
LBP	80.07 ± 1.35	79.98 ± 0.39	82.28 ± 0.41	80.65 ± 0.47	83.37 ± 0.54	85.57 ± 0.52	–	86.73 ± 0.53	**87.12 ± 0.41**
SIFT	80.85 ± 0.61	78.12 ± 0.45	81.27 ± 2.30	77.50 ± 0.50	83.08 ± 0.56	–	–	85.55 ± 0.61	**85.98 ± 0.40**
Attribute	–	84.00	–	83.40	84.60	–	85.29	84.77	**85.62**

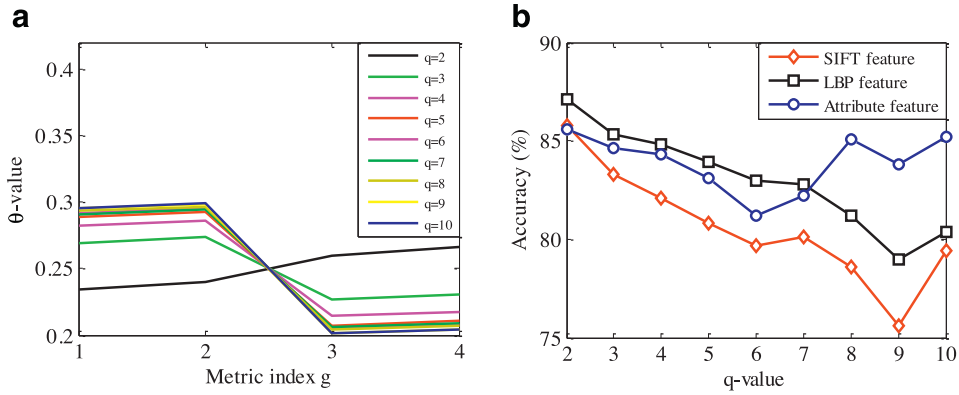

Fig. 7. The weights of MetricFusion for different q-values (a) and performance curve of three features with different q-values ($q = 2,\ldots,10$).

Table 5
Accuracy (%) comparison with state of the art results on LFW dataset under image restricted protocol.

Method	Features	NoD	Accuracy
Combine b/g samples [4]	SIFT, LBP, etc.	8	86.83 ± 0.34
LDML + SVM [31]	SIFT, LBP, etc.	8	79.27 ± 0.60
DML-eig + SVM [11]	SIFT, LBP, etc.	8	85.65 ± 0.56
SILD +SVM [21]	Intensity, LBP, etc.	8	85.78 ± 2.05
CSML + SVM [34]	Intensity + LBP, etc.	6	88.00 ± 0.37
HTBI [33]	Inspired features	16	88.13 ± 0.58
Att.&Sim. classifiers [19]	Attributes	1	85.29
Sub-SML + SVM [8]	LBP + SIFT	2	88.87 ± 0.60
Sub-SML + SVM [8]	LBP, SIFT, Attributes	3	89.33 ± 0.54
SFRD + PMML [18]	Spatial-temporal	8	89.35 ± 0.50
Sub-SML + SVM [8]	LBP, SIFT, etc.	6	89.73 ± 0.38
SEAML [23]	SIFT, Attributes	2	87.50 ± 1.30
ITML + Multiple OSS [18]	SIFT, LBP, etc.	16	89.50 ± 1.58
GMSL	LBP, SIFT	2	89.35 ± 0.44
GMSL	LBP, SIFT, Attributes	3	**90.67** ± 0.46

mance of 89.35% and 89.50% with 8 and 16 descriptors, respectively. Furthermore, the ROC curves and AUCs (Area Under Curve) for state-of-the-art methods are illustrated in Fig. 8, from which we can clearly observe that our proposed GMSL method outperforms other related metric learning methods in restricted setting.

Additionally, we also employ a comparison with existing deep learning methods on LFW in restricted protocol. We compare our GMSL with four recently proposed deep learning based face verification methods: convolutional deep belief network (CDBN) [35], deep nonlinear metric learning with independent subspace analysis (DNLML-ISA) [36], discriminative shadow metric learning (DSML) [27] and discriminative deep metric learning (DDML) [27]. The comparisons with existing deep learning based face verification in restricted setting are shown in Table 6. We see that the DDML achieves the best accuracy of 90.68% with 6 descriptors. While our GMSL achieves 90.67% with only 3 descriptors, and 0.01% lower than DDML. The ROC curves of GMSL and DDML are shown in Fig. 9. The observed results demonstrate GMSL achieves comparative performance with DDML.

Table 6
Accuracy (%) comparisons with existing deep metric learning on LFW data in restricted protocol.

Method	Features	NoD	Accuracy
CDBN [37]	Image descriptors, etc.	6	86.88 ± 0.62
CDBN + Hand-crafted	Hand-crafted, etc.	12	87.77 ± 0.62
DNLML-ISA [36]	LBP, SIFT, etc.	8	88.50 ± 0.40
DSML [27]	LBP, SIFT, etc.	6	87.45 ± 1.45
DDML [27]	Sparse SIFT (SSIFT)	1	87.83 ± 0.93
DDML [27]	LBP, SIFT, etc.	6	**90.68** ± 1.41
GMSL	LBP, SIFT	2	89.35 ± 0.44
GMSL	LBP, SIFT, Attribute	3	**90.67 ± 0.46**

In this paper, four metrics are learned simultaneously in GMSL. For better visualization of the effectiveness of GMSL, we have conducted the experiment using single metric and the direct combi-

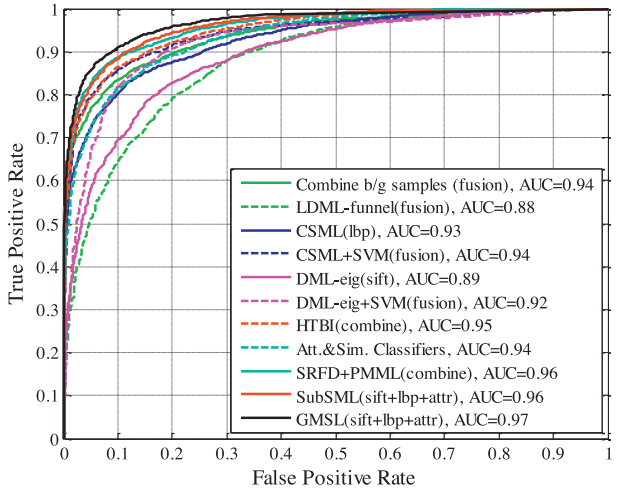

Fig. 8. Comparisons of ROC curves and AUCs between our GMSL and the state-of-the-art methods on LFW.

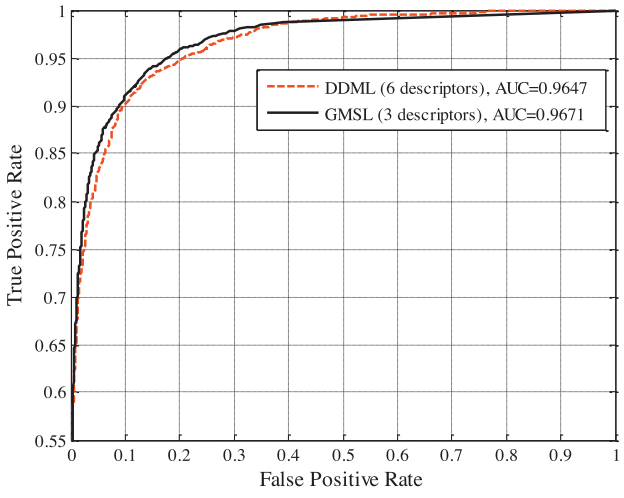

Fig. 9. Comparison of ROC curves and AUCs.

Table 7
Performance comparisons between our GMSL and other metric learning on Pubfig faces.

Methods	Euclidean	LMNN [11]	ITML [9]	DML-eig [10]	LDML [5]	KissMe [37]	SubSML [8]	GMSL
Accuracy (%)	72.5	73.5	69.3	77.4	77.6	77.6	77.3	**78.5**

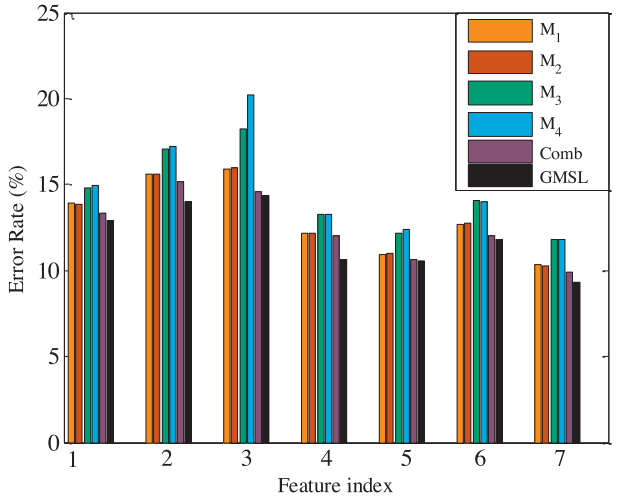

Fig. 10. Comparison with single metric and combination of four metrics. 1: LBP; 2: SIFT; 3: Attribute; 4: LBP + SIFT; 5: LBP + Attribute; 6: SIFT + Attribute; and 7: LBP + SIFT + Attribute.

Fig. 11. Some intra and inter pairs of faces in PubFig.

nation of four metrics learned individually. Fig. 10 shows the error rate of LFW for different feature types in restricted setting by using four metrics and their combination, respectively. We can observe that the proposed GMSL by integrating multiple metrics learned simultaneously has a better performance. The single metric $\mathbf{M}_1$ and $\mathbf{M}_2$ obtain better performance than $\mathbf{M}_3$ and $\mathbf{M}_4$. The direct combination of four metrics is not very useful for performance improvement. This demonstrates the importance of learning latent metrics simultaneously as a merit of our GMSL.

4.4. Test results on PubFig faces

The Public Figures (PubFig) data shares similar property with LFW for unconstrained face verification [26]. It consists of 58,797 images from 200 people. Some samples from PubFig are shown as Fig. 11.

For performance evaluation, 20,000 pairs from 140 people are divided into 10 folds, and each fold contains 1000 same and 1000 not same pairs. Like LFW, 10-fold cross validation is adopted for evaluation. The average accuracies of several popular metric learning methods are reported. The comparison results between our method and state-of-the-art metric learning methods on PubFig under the restricted setting are shown in Table 7, in which some

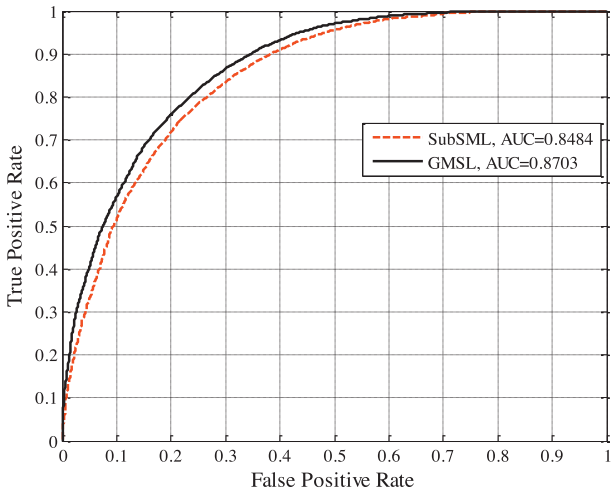

Fig. 12. ROC curves and AUCs of SubSML and GMSL.

results are from [37]. The accuracy 78.5% of our GMSL is about 1% improvement than KissMe [37]. To compare with SubSML which is closely related with our method, we also present the ROC curves in Fig. 12, and the effectiveness of our method is obviously demonstrated.

5. Conclusion

In this paper, we proposed to learn a metric swarm in local patches and enforce a vectorized similarity space reconstruction for general classification problem and unconstrained face verification. Specifically, in the proposed GMSL, we aim at obtaining a metric swarm by learning local patch based sub-metrics simultaneously with a regularized metric learning model. The dual problem of GMSL is denoted as a quadratic programming problem, which is efficiently solved by FISTA algorithm via an alternative optimization algorithm. Then the local patch sub-metrics can be represented by the dual solution. With the solved metric swarm, the sample pairs are transformed into a vectorized similarity space (metric swarm space) via an established joint similarity function, where a SVM-like classification can be easily implemented in the represented space for verification tasks. Experiments on several benchmark UCI datasets preliminarily demonstrate its effectiveness in general classification problem. Further experiments on real-world LFW and PubFig faces datasets under restricted setting demonstrate the best performance of our GMSL for unconstrained face verification.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 61401048, the Fundamental Research Funds for the Central Universities and the Hong Kong Scholar Program under Grant XJ2013044.

References

[1] G.B. Huang, M. Ramesh, T. Berg, and E.L. Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environment. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[2] J. Lu, Y.P. Tan, Regularized locality perserving projections and its extensions for face recognition, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (2010) 958–963.

[3] J. Lu, G. Wang, W. Deng, K. Jia, Reconstruction based metric learning for unconstrained face verification, IEEE Trans. Inf. Forensics Secur. 10 (2015) 79–89.

[4] H. Ling, S. Soatto, N. Ramanathan, D. Jacobs, Face verification across age progression using discriminative methods, IEEE Trans. Inf. Forensics Secur. 5 (2010) 82–91.

[5] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: Proceedings of International Conference on Computer Vision, ICCV, 2009, pp. 498–505.

[6] Y. Taigman, L. Wolf, T. Hassner, Multiple one-shots for utilizing class label information, in: Proceedings of British Machine Vision Conference, BMVC, 2009, pp. 1–12.

[7] H.V. Nguyen, L. Bai, Cosine similarity metric learning for face verification, in: Proceedings of Asian Conference on Computer Vision, ACCV, 2011, pp. 709–720.

[8] Q. Cao, Y. Ying, P. Li, Similarity metric learning for face recognition, in: Proceedings of International Conference on Computer Vision, ICCV, 2013, pp. 4321–4328.

[9] J. Davis, B. Kulis, P. Jain, S. Sra, I. Dhillon, Information theoretic metric learning, in: Proceedings of International Conference on Machine Learning, ICML, 2007.

[10] Y. Ying, P. Li, Distance metric learning with eigenvalue optimization, J. Mach. Learn. Res. 13 (2012) 1–26.

[11] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Proceedings of International Conference on Neural Information Processing Systems, NIPS, 2006.

[12] C. Shen, J. Kim, L. Wang, Scalable large-margin Mahalanobis distance metric learning, IEEE Trans. Neural Netw 219 (2010) 1524–1530.

[13] W. Bian, D. Tao, Constrained empirical risk minimization framework for distance metric learning, IEEE Trans. Neural Netw. Learn. Syst. 23 (2012) 1194–1205.

[14] F. Wang, W. Zuo, L. Zhang, D. Meng, D. Zhang, A kernel classification framework for metric learning, IEEE Trans. Neural Netw. Learn. Syst. 26 (9) (2015) 1950–1962.

[15] S. Parameswaran, K. Weinberger, Large margin multi-task metric learning, in: Proceedings of International Conference on Neural Information Processing Systems, NIPS, 2010.

[16] G. Niu, B. Dai, M. Yamada, M. Sugiyama, Information-theoretic semi-supervised metric learning via entropy regularization, in: Proceedings of International Conference on Machine Learning, ICML, 2012.

[17] S. Hoi, W. Liu, S. Chang, Semi-supervised distance metric learning for collaborative image retrieval, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR, 2008.

[18] D. Kedem, S. Tyree, K. Weinberger, F. Sha, G. Lanckriet, Nonlinear metric learning, in: Proceedings of International Conference on Neural Information Processing Systems, NIPS, 2012.

[19] T. Xia, D. Tao, T. Mei, Y. Zhang, Multiview spectral embedding, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 40 (2010) 1438–1446.

[20] B. Xie, Y. Mu, D. Tao, K. Huang, M-SNE: multiview stochastic neighbor embedding, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 41 (2011) 1088–1096.

[21] H. Wang, J. Yuan, Collaborative multifeature fusion for transductive spectral learning, IEEE Trans. Cybern. 45 (2014) 465–475.

[22] Y. Xiao, B. Liu, Z. Hao, L. Cao, A similarity-based classification framework for multiple instance learning, IEEE Trans. Cybern. 44 (2013) 500–515.

[23] D.T. Nguyen, C.D. Nguyen, R. Hargraves, L.A. Kurgan, mi-DS: multiple-instance learning algorithm, IEEE Trans. Cybern. 43 (2012) 143–154.

[24] Z. Cui, W. Li, D. Xu, S. Shan, X. Chen, Fusing robust face region descriptors via multiple metric learning for face recognition in the wild, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR, 2013, pp. 3554–3561.

[25] J. Hu, J. Lu, J. Yuan, Y.P. Tan, Large margin multi-metric learning for face and kinship verification in the wild, in: Proceedings of Asian Conference on Computer Vision, ACCV, 2014, pp. 252–267.

[26] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: Proceedings of International Conference on Computer Vision, ICCV, 2009.

[27] J. Hu, J. Lu, Y.P. Tan, Discriminative deep metric learning for face verification in the wild, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 1875–1882.

[28] A. Beck, M. Teboulle, A fast iterative shrinkage thresholding algorithms for linear inverse problems, SIAM J. Imaging Sci. 2 (2009) 183–202.

[29] Y. Ying, K. Huang, C. Campbell, Sparse metric learning via smooth optimization, in: Proceedings of International Conference on Neural Information Processing Systems, NIPS, 2009.

[30] A. Frank and A. Asuncion, UCI Machine Learning Repository [Online], available: http://archive.ics.edu/ml.

[31] L. Wolf, T. Hassner, Y. Taigman, Similarity scores based on background samples, in: Proceedings of Asian Conference on Computer Vision, ACCV, 2010, pp. 88–97.

[32] M. Kan, S. Shan, D. Xu, X. Chen, Side-information based linear discriminant analysis for face recognition, in: Proceedings of British Machine Vision Conference, BMVC, 2011.

[33] N. Pinto, D. Cox, Beyond simple features: a large scale feature search approach to unconstrained face recognition, in: Proceedings of International Conference on Automatic Face and Gesture Recognition, 2011.

[34] Q. Wang, W. Zuo, L. Zhang, P. Li, Shrinkage expansion adaptive metric learning, in: Proceedings of European Conference on Computer Vision, ECCV, 2014.

[35] G.B. Huang, H. Lee, E.G. Learned-Miller, Learning hierarchical representations for face verification with convolutional deep belief networks, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR, 2012, pp. 2518–2525.

[36] X. Cai, C. Wang, B. Xiao, X. Chen, J. Zhou, Deep nonlinear metric learning with independent subspace analysis for face verification, in: Proceedings of ACM International Conference on Multimedia, ACM MM, 2012, pp. 749–752.

[37] M. Kostinger, M. Hirzer, P. Wohlhart, P.M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, CVPR, 2012.