Joint Prototype and Metric Learning for Set-to-Set Matching: Application to Biometrics

Mengjun Leng, Panagiotis Moutafis, and Ioannis A. Kakadiaris
Computational Biomedicine Lab
Department of Computer Science, University of Houston
4800 Calhoun Rd. Houston, TX, 77004
{mleng2, pmoutafis, ioannisk}@uh.edu

Abstract

In this paper, we focus on the problem of image set classification. Since existing methods utilize all available samples to model each image set, the corresponding time and storage requirements are high. Such methods are also susceptible to outliers. To address these challenges, we propose a method that jointly learns prototypes and a Mahalanobis distance. The prototypes learned represent the gallery image sets using fewer samples, while the classification accuracy is maintained or improved. The distance learned ensures that the notion of similarity between sets of images is reflected more accurately. Specifically, each gallery set is modeled as a hull spanned by the learned prototypes. The prototypes and distance metric are alternately updated using an iterative scheme. Experimental results using the YouTube Face, ETH-80, and Cambridge Hand Gesture datasets illustrate the improvements obtained.

1. Introduction

Image set identification has been an active field of research for more than a decade [4, 23, 3, 1, 5, 22]. This problem is often encountered in biometric applications including video-based face recognition [25], person re-identification using multi-camera networks [21], and video surveillance [16]. The goal in these tasks is to identify the gallery subject that corresponds to a probe. Unlike traditional identification, though, the gallery and probes are defined using sets of samples for each subject.

Our focus is video-based face recognition, where each video is modeled as a set of images. Using video streams for recognition provides additional information which can improve recognition accuracy [4, 23, 3]. However, it also introduces new challenges. For instance, modeling the similarity between sets of vectors is a difficult task. Existing methods define a model for each image set and then learn a distance that accurately measures the similarity between them. These approaches can be grouped into four categories: (i) parametric, (ii) subspace, (iii) statistical, and (iv) hull-based. In the first category, each image set is usually assumed to follow a Gaussian [5] or a mixture of Gaussians [1]. The Kullback-Leibler divergence is then used to measure the distance between two sets. Recently, Huang *et al*. [10] proposed a mixture model that combines the mean, covariance matrix, and a Gaussian distribution to represent each image set. The heterogeneous representations are fused via a Hybrid Euclidean-and-Riemannian Metric Learning approach. Methods in this category make strong assumptions concerning the distribution of the data which may not always be true. *Subspace-based methods* model each image set using either a single linear subspace [7, 6, 22] or a non-linear mixture of subspaces [18, 12]. Each subspace can be viewed as a point on a manifold, and the dissimilarity between the image sets is defined using the corresponding geodesic distance [6]. However, such methods require large datasets with dense sampling to learn the manifold [17]. *Statistical methods* use properties of the data to represent each set. Wang *et al*. [17] consider the covariance matrix as points on a Riemannian manifold. To measure the matrix to matrix distance, the Log Euclidean Distance is used. This method was extended by using multi-order statistics (*i.e*., mean, covariance matrix, and tensor) by Lu *et al*. [15]. Localized Multi-Kernel Metric Learning combines the multi-order statistics to learn a distance with better discriminative properties. Relying only on a few statistical properties, though, may ignore significant information in the data. *Hull-based methods* model each image set using affine hulls [3] or other types of reduced affine hulls [9]. The set-to-set distance is defined as the distance between the nearest points between pairs of hulls. To better measure the distance between sets under this model, Zhu *et al*. [24] introduced distance metric learning for set-to-set matching. More recently, they extended

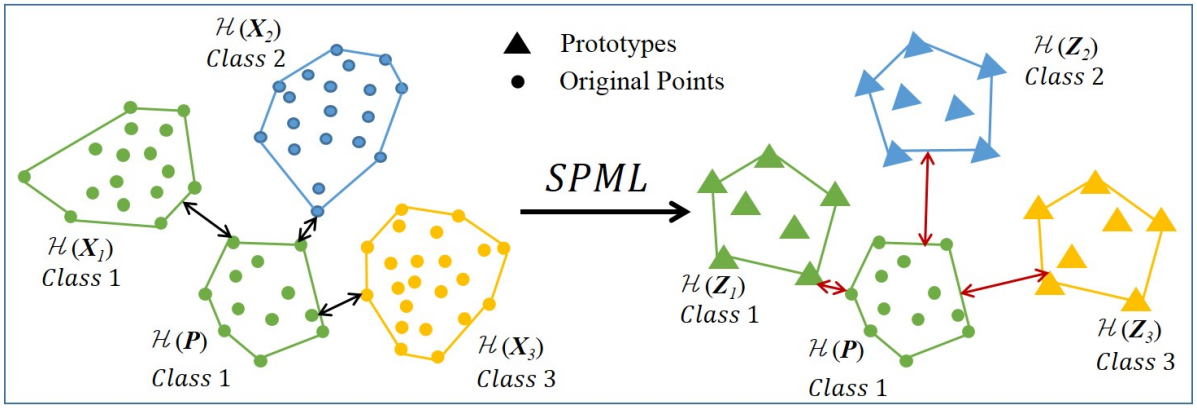

Figure 1. Illustration of SPML. (L): The $\mathcal{H}(\boldsymbol{X}_1)$, $\mathcal{H}(\boldsymbol{X}_2)$, and $\mathcal{H}(\boldsymbol{X}_3)$ denote three classes in the gallery, while $\mathcal{H}(\boldsymbol{P})$ denotes a probe set. The $\mathcal{H}(\boldsymbol{X}_1)$ and $\mathcal{H}(\boldsymbol{P})$ belong to the same class. (R): The $\mathcal{H}(\boldsymbol{Z}_1)$, $\mathcal{H}(\boldsymbol{Z}_2)$, and $\mathcal{H}(\boldsymbol{Z}_3)$ denote the prototypes learned to represent the corresponding classes in the gallery. As indicated, fewer samples are used. SPML learns the prototypes and a corresponding Mahalanobis distance in such a way that the distances between similar sets are "smaller", while the distances between dissimilar sets are "larger".

the hull-based model using an Image Set Collaborative Reconstruction (ISCRC) approach [25]. This approach considers the correlations between different gallery sets. Even though affine hull-based methods offer a good model for set-to-set matching, they appear to be sensitive to outliers. In summary, even though the methods discussed above address the problem of matching sets of images, challenges remain. The storage and processing costs increase greatly for large scale applications. Since the useful information is contained only in a small fraction of the data, errors in pre-processing and redundancies can degrade the recognition performance. With video-based face recognition there are thousands of frames and in many the target face changes slightly or the detection fails. Considering all samples is computationally expensive and also degrades the recognition performance.

To address this gap, we propose a Set-to-set Prototype and Metric Learning framework (SPML). Our approach extends the method of Köstinger *et al*. [13] to set-to-set matching. Specifically, prototypes with discriminative properties and a distance metric for image-set-based classification are jointly learned. The objective of the prototype learning component of our framework is to represent the gallery image set by using fewer templates while maintaining or improving the recognition performance. Each gallery image set is then modeled as a hull spanned by the prototypes learned. To accurately reflect the notion of similarity when matching a probe with the learned prototypes, a Mahalanobis distance metric is jointly learned. We cast the optimization problem using a single loss function that jointly learns the prototypes and metric learning. Specifically, it brings similar image sets "closer" to each other, while pushing dissimilar ones "far away", as illustrated by Fig. 1.

Our contribution is a method with these advantages: (i) it uses fewer prototypes to represent each image set in the gallery, reducing the computational cost and storage requirement; (ii) it increases the robustness of the hull model; and (iii) it can be used with any hull model and any distance metric learning objective function.

The rest of the paper is organized as follows: in Sec. 2 we offer a brief introduction on hull distances; in Sec. 3 we describe the proposed method; in Sec. 4 we present the experimental results; and Sec. 5 concludes the paper.

2. Background

In this section, we introduce the notation used throughout the paper and review concepts concerning affine hulls that will help the reader understand the proposed method. Specifically, we focus on the definition of different affine hulls used to model image sets, and their corresponding distance measures.

Affine Hull: An image set was first modeled as an affine hull by Cevikalp *et al*. [3]. Let $\boldsymbol{\mathcal{G}} = \{(\boldsymbol{X}_i, y_i)\}$ be a gallery, where $\boldsymbol{X}_i$ is the i^{th} image set, and y_i is its corresponding class label. Each image set $\boldsymbol{X}_i = [\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, ..., \boldsymbol{x}_i^{N_i}] \in \mathbb{R}^{d \times Ni}$ can be represented as a $\mathbb{R}^{d \times Ni}$ matrix, where $\boldsymbol{x}_i^m$ denotes the d-dimensional feature vector for the m^{th} image in $\boldsymbol{X}_i$, and N_i denotes the number of images in $\boldsymbol{X}_i$. In the hull model [3] each image set is modeled as a subspace spanned by the sample vectors it contains. For instance, let $\mathcal{H}(\boldsymbol{X}_i)$ denote the affine hull generated by $\boldsymbol{X}_i$. Then, $\mathcal{H}(\boldsymbol{X}_i)$ is defined as follows:

$$\mathcal{H}(\boldsymbol{X}_i) = \left\{ \sum_{m=1}^{N_i} \alpha_i^m \boldsymbol{x}_i^m \mid \sum_{m=1}^{N_i} \alpha_i^m = 1, \alpha_i^m \in \boldsymbol{\mathcal{R}} \right\}, \quad (1)$$

where m indicates the index of samples of the $\boldsymbol{X}_i$ set. The notation $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, ..., \alpha_i^{N_i}]^T$ is used to denote the co-

efficient vector for $\mathcal{H}(\boldsymbol{X}_i)$. Depending on the data at hand there are different restrictions for the coefficients α_i^m. For the general model of affine hull, $\boldsymbol{\mathcal{R}}$ is set to $(-\infty, +\infty)$. However, some outlying samples might cause the hulls to overlap. To address this problem, the range of the coefficients can be restricted (*e.g.*, set $\boldsymbol{\mathcal{R}} = [0, 1]$), resulting in a *convex hull*. The l_p norm of $\boldsymbol{\alpha}_i$ can be bounded from above (*i.e.*, $\boldsymbol{\mathcal{R}} = \left\{\alpha_i^m \mid \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma\right\}$), to obtain a *regularized affine hull* [23].

Set-to-Set Distance: The dissimilarity between two image sets $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ is measured by the square distance between the nearest points of these two hulls:

$$\mathcal{D}^2(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left((\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j)^T(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j)\right). \tag{2}$$

To increase the class-separation between image sets, a Mahalanobis metric $\boldsymbol{M}$ was introduced by Zhu *et al*. [24]:

$$\begin{aligned} &\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{X}_j) = (\boldsymbol{X}_i\hat{\boldsymbol{a}}_i - \boldsymbol{X}_j\hat{\boldsymbol{a}}_j)^T\boldsymbol{M}(\boldsymbol{X}_i\hat{\boldsymbol{a}}_i - \boldsymbol{X}_j\hat{\boldsymbol{a}}_j) \\ &(\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{a}}_j) = arg \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{X}_j), \end{aligned} \tag{3}$$

where $\boldsymbol{M}$ is a positive semi-definite matrix.

3. Method

In this section, we describe the SPML framework which extends the affine hull model of Eq. (3). Specifically, our objective is to jointly learn: (i) prototypes to represent each gallery image set, and (ii) a corresponding Mahalanobis distance.

Loss function: The proposed loss function $\mathcal{L}$ is minimized in $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{M}$:

$$(\boldsymbol{\mathcal{Z}}, \boldsymbol{M}) = arg \min_{\boldsymbol{\mathcal{Z}}, \boldsymbol{M}} \mathcal{L}\left(\boldsymbol{X}, \boldsymbol{\mathcal{Z}}, \boldsymbol{M}\right), \tag{4}$$

where $\boldsymbol{\mathcal{Z}} = \{(\boldsymbol{Z}_i, y_i)\}$ is a gallery defined using learned prototypes $\boldsymbol{Z}_i = [\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, ..., \boldsymbol{z}_i^K] \in \mathbb{R}^{d\times K}$, with $K < N_i$. Specifically, for each gallery hull $\mathcal{H}(\boldsymbol{X}_i)$ a new prototype hull spanning $\boldsymbol{Z}_i$ is learned:

$$\mathcal{H}(\boldsymbol{Z}_i) = \{\sum_{m=1}^{K} \beta_i^m \boldsymbol{z}_i^m | \sum_{m=1}^{K} \beta_i^m = 1, \beta_i^m \in \boldsymbol{\mathcal{R}}\}, \tag{5}$$

where $\boldsymbol{\beta}_i = [\beta_i^1, \beta_i^2, ..., \beta_i^K]^T$ denotes the coefficient vector for $\mathcal{H}(\boldsymbol{Z}_i)$. The prototypes $\boldsymbol{Z}_i$ should be able to represent the original image sets $\boldsymbol{X}_i$, while having better discriminative properties. The Mahalanobis distance $\boldsymbol{M}$ should increase the class separation between the image sets. The proposed loss function $\mathcal{L}$ is a variant of the Large Margin Nearest Neighbors (LMNN) approach [19] and is given by:

$$\mathcal{L}\left(\boldsymbol{X}, \boldsymbol{\mathcal{Z}}, \boldsymbol{M}\right) = \sum_{\boldsymbol{\mathcal{G}}} \mathcal{L}_i(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}}, \boldsymbol{M}). \tag{6}$$

Each term $\mathcal{L}_i$ is respectively defined as:

$$\begin{aligned} \mathcal{L}_i(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}}, \boldsymbol{M}) =& (1-\mu)\sum_{\boldsymbol{\mathcal{S}}_i} \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) \\ &+\mu\sum_{\boldsymbol{\mathcal{V}}_i} [2\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) - \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_l)]_+, \end{aligned} \tag{7}$$

where $\boldsymbol{\mathcal{S}}_i$ contains the indices of the κ-nearest prototype sets to $\boldsymbol{X}_i$ labeled as y_i (*i.e.*, target neighbors), $\boldsymbol{\mathcal{V}}_i = \{(j, l) | j \in \boldsymbol{\mathcal{S}}_i$, and $l : y_l \neq y_i\}$, $[x]_+ = max(x, 0)$, and μ determines the trade-off between the two terms. The objective of the first term is to pull target neighbors (*i.e.*, $\boldsymbol{Z}_j$) "closer", while the objective of the second term is to push impostors (*i.e.*, $\boldsymbol{Z}_l$) "far away". The LMNN function was selected due to its robustness. Other loss functions could have been selected instead.

Minimizing $\mathcal{L}$: The prototype gallery $\boldsymbol{\mathcal{Z}}$ and the Mahalanobis matrix $\boldsymbol{M}$ are learned via minimizing Eq. 4 using an EM-like approach [13, 24]. Specifically, gradient descent is employed to update $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{M}$ alternately. The neighborhood information (*i.e.*, $\boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{\mathcal{V}}_i$) is updated accordingly.

Update $\boldsymbol{M}$: The partial derivative of $\mathcal{L}$ regarding $\boldsymbol{M}$ is given by:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\boldsymbol{M}} &= \sum_{\boldsymbol{\mathcal{G}}} \frac{\partial\mathcal{L}_i}{\partial\boldsymbol{M}} \\ &= \sum_{\boldsymbol{\mathcal{G}}} \left((1-\mu)\sum_{\boldsymbol{\mathcal{S}}_i} \boldsymbol{C}_{ij} + \mu\sum_{\boldsymbol{\mathcal{V}}_{i+}} (2\boldsymbol{C}_{ij} - \boldsymbol{C}_{il})\right), \end{aligned} \tag{8}$$

where $\boldsymbol{C}_{ij} = (\boldsymbol{X}_i\hat{\boldsymbol{a}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{b}}_j)(\boldsymbol{X}_i\hat{\boldsymbol{a}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{b}}_j)^T$, and $\boldsymbol{\mathcal{V}}_{i+} = \left\{(j, l) \mid 2\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) - \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_l) > 0\right\}$. That is, $\boldsymbol{\mathcal{V}}_{i+}$ is a subset of $\boldsymbol{\mathcal{V}}_i$, containing the index pairs for which the hinge loss in $\mathcal{L}_i$ is larger than zero. The rule to update $\boldsymbol{M}$ at the $(t+1)^{th}$ iteration is given by:

$$\boldsymbol{M}^{t+1} = \boldsymbol{M}^t - \eta_M \frac{\partial\mathcal{L}(\boldsymbol{\mathcal{Z}}^t, \boldsymbol{M}^t)}{\partial\boldsymbol{M}^t}, \tag{9}$$

where η_M is the learning rate. To ensure that $\boldsymbol{M}$ is positive semi-definite, the updated $\boldsymbol{M}$ is projected onto its nearest positive semi-definite matrices as in [8].

Update $\boldsymbol{\mathcal{Z}}$: Each prototype set $\boldsymbol{Z}_k \in \boldsymbol{\mathcal{Z}}$ is optimized independently. The partial derivative of the loss function $\mathcal{L}$ regarding $\boldsymbol{Z}_k$ is:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{Z}_k} = \sum_{\boldsymbol{\mathcal{G}}} \frac{\partial\mathcal{L}_i}{\partial\boldsymbol{Z}_k}. \tag{10}$$

Since $\boldsymbol{Z}_k$ is sometimes considered a target neighbor or an impostor for different $\boldsymbol{X}_i$, the corresponding partial derivatives vary. Specifically, if $\boldsymbol{Z}_k$ is a target neighbor (*i.e.*,

$k \in \boldsymbol{\mathcal{S}}_i$), the partial derivative is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{Z}_k} =& -2(1-\mu)\sum_{k\in\boldsymbol{\mathcal{S}}_i}\boldsymbol{M}(\boldsymbol{X}_i\hat{\boldsymbol{a}}_i-\boldsymbol{Z}_k\hat{\boldsymbol{b}}_k)\hat{\boldsymbol{b}}_k^T \\ &-4\mu\sum_{(k,l)\in\boldsymbol{\mathcal{V}}_{i+}}\boldsymbol{M}(\boldsymbol{X}_i\hat{\boldsymbol{a}}_i-\boldsymbol{Z}_k\hat{\boldsymbol{b}}_k)\hat{\boldsymbol{b}}_k^T.\end{aligned} \tag{11}$$

If $\boldsymbol{Z}_k$ is an impostor that violates the predefined margin (*i.e.*, $(j,k) \in \boldsymbol{\mathcal{V}}_{i+}$), its partial derivative is given by:

$$\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{Z}_k} = 2\mu\sum_{(j,k)\in\boldsymbol{\mathcal{V}}_{i+}}\boldsymbol{M}(\boldsymbol{X}_i\hat{\boldsymbol{a}}_i-\boldsymbol{Z}_k\hat{\boldsymbol{b}}_k)\hat{\boldsymbol{b}}_k^T. \tag{12}$$

In all other cases, $\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{Z}_k} = 0$. The rule to update $\boldsymbol{Z}_k$ at the $(t+1)^{th}$ iteration is given by:

$$\boldsymbol{Z}_k^{t+1} = \boldsymbol{Z}_k^t - \eta_{\mathcal{Z}}\frac{\partial \mathcal{L}_i(\boldsymbol{\mathcal{Z}}^t, \boldsymbol{M}^t)}{\partial \boldsymbol{Z}_k}, \tag{13}$$

where $\eta_{\mathcal{Z}}$ is the learning rate for $\boldsymbol{\mathcal{Z}}$.

Update $\mathcal{D}_M^2(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$: Once $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{M}$ have been updated the corresponding distance and neighborhood information should be redefined. The Mahalanobis metric $\boldsymbol{M}$ can be decomposed into $\boldsymbol{L}^T\boldsymbol{L}$ via Cholesky decomposition. The distance between $\boldsymbol{X}_i$ and $\boldsymbol{Z}_j$ can be written as:

$$\begin{aligned}\mathcal{D}_M^2(\boldsymbol{X}_i, \boldsymbol{Z}_j) &= [\boldsymbol{L}(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)]^T\boldsymbol{L}(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j) \\ (\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{b}}_j) &= arg\min_{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_j}\|\boldsymbol{L}(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)\|_2^2.\end{aligned} \tag{14}$$

It is equivalent to search for the nearest points on two hulls in the projected space defined by $\boldsymbol{L}$. The coefficients $(\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{b}}_j)$ can be computed using different constraints (*e.g.*, affine hull, convex hull, SNAP, and RNP). In this paper, the RNP [23] is used due to its robustness:

$$\begin{aligned}(\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{b}}_j) =& arg\min_{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_j}\|\boldsymbol{L}(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)\|_2^2 \\ &+\lambda_1\|\boldsymbol{\alpha}_i\|_2^2 + \lambda_2\|\boldsymbol{\beta}_j\|_2^2.\end{aligned} \tag{15}$$

To minimize Eq. 15 the closed-form solution offered by [24] is employed. Once $\mathcal{D}_M^2(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$ has been updated, the neighborhood relationships $(\boldsymbol{\mathcal{S}}_i, \boldsymbol{\mathcal{V}}_i)$ can be redefined accordingly.

Implementation: An overview of the training procedure is offered by Algorithm 1.

Line 1: The matrix $\boldsymbol{M}$ is initialized using an identity matrix of the corresponding dimensions. The prototypes can be initialized in many ways, such as selecting random samples or clustering the original image set. The values of the initial learning rates η_M and $\eta_{\mathcal{Z}}$ are set empirically.

Line 3: In our implementation, the stopping condition is defined as the union of three criteria: (i) the relative change of $\mathcal{L}$ is smaller than a threshold $\omega_{\mathcal{L}}$ using a window of five iterations; (ii) both learning rates are smaller than a threshold ω_η; or (iii) there are no impostors.

Lines 4-6, 8, 10-12, 14: If the update overshoots (*i.e.*, $\mathcal{L}^{t+1} > \mathcal{L}^t$), the learning rate is reduced by a factor of σ_r to increase the stability of the algorithm. If $\boldsymbol{M}$ and $\boldsymbol{\mathcal{Z}}$ are updated successfully, the corresponding learning rates are increased by a factor σ_g to speed up the convergence. The values of σ_r and σ_g are set empirically.

Algorithm 1 Set-based Prototypes and Metric Learning

Input: $\boldsymbol{\mathcal{G}}$
Output: $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{M}$

1: Initialize $\boldsymbol{M}_0$, $\boldsymbol{\mathcal{Z}}_0$, η_M, $\eta_{\mathcal{Z}}$
2: **procedure** $(\boldsymbol{\mathcal{Z}}, \boldsymbol{M})$ =*SPML*($\boldsymbol{\mathcal{G}}$)
3: **while** convergence criterion is not met **do**
4: **while** $\mathcal{L}^{t+1} > \mathcal{L}^t$ **do**
5: $\eta_{\mathcal{Z}} = (1-\sigma_r)\eta_{\mathcal{Z}}$
6: **end while**
7: Update $\boldsymbol{\mathcal{Z}}$ (Eq. (13))
8: $\eta_{\boldsymbol{\mathcal{Z}}} = (1+\sigma_g)\eta_{\mathcal{Z}}$
9: Update $\mathcal{D}_M^2(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$ (Eq. (14))
10: **while** $\mathcal{L}^{t+1} > \mathcal{L}^t$ **do**
11: $\eta_M = (1-\sigma_r)\eta_M$
12: **end while**
13: Update $\boldsymbol{M}$ (Eq. (9))
14: $\eta_M = (1+\sigma_g)\eta_M$
15: Update $\mathcal{D}_M^2(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$ (Eq. (14))
16: **end while**
17: **end procedure**

4. Experiments

In this section, we discuss important implementation details and describe the datasets used in our experiments. Finally, we present the corresponding results.

4.1. Datasets

The ETH-80 [14], Cambridge Hand Gesture Dataset(CHG) [11] and YouTube Face (YTF) [20] datasets were selected to assess the performance of the proposed SPML in three tasks (*i.e.*, object categorization, gesture recognition, and video-based face identification, respectively).

ETH-80: This dataset comprises eight categories, where each category contains 10 objects. For each object, 41 images from different views are captured to form an image set. Following [17], the original images are resized to 20×20 and the concatenated pixel values are used as features. For all experiments, five objects are randomly sampled from each category to form the gallery, while the rest are used as probes. A 10-fold cross-validation protocol is used to

report the average performance.

CHG: This dataset comprises 900 image sequences for nine types of gestures. These gestures result from the combination of three hand shapes and three motions. For each class (*i.e*., gesture), there are 100 image sequences, which include five illumination conditions, 10 arbitrary motions, and two subjects (*i.e*., left and right hands). The number of images in each sequence varies from 37 to 119. Each sequence is used to form a set in our experiments. Following [4], the original image is resized to 20×20 and the concatenated pixel values are used as features. For all experiments, 20 sequences are randomly sampled from each class to form the gallery, while the remaining 80 are used as probes. A 10-fold cross-validation protocol is used to report the average performance.

YTF: This dataset contains $3,425$ videos captured from $1,595$ subjects. Our goal is to simulate a face identification task. However, for most subjects only a single video is provided and thus these videos cannot be used to evaluate the identification performance. A subset of 59 subjects was selected for which five or more videos are available. For each video, the number of valid image frames (*i.e*., a face is detected) varies from 48 to $2,157$. This dataset comes with three feature descriptors: Local Binary Patterns (LBP), Center-Symmetric LBP (CSLBP) and Four-Patch LBP (FPLBP). For all experiments, four videos are randomly selected from each subject to form the gallery, while the rest are used as probes. A 10-fold cross-validation protocol is used to report the average performance.

Feature processing: To reduce the noise and avoid over-fitting, Principal Component Analysis (PCA) is applied to all the features. For ETH-80 and CHG the length of the feature vectors was selected so that 90% of the total variation is retained. For the YTF dataset, the feature length was arbitrarily set to 100. To reduce large intra-class variations, these features were projected onto an intra-class subspace following the procedure described in [2].

4.2. Parameter Settings

In this section, we discuss the parameter settings for all algorithms used in our experiments. To conduct a fair comparison, all important parameters are tuned empirically according to their original papers.

SPML: Each prototype set is initialized using k-means clustering, while the number of prototypes used is set to ten. The trade-off parameter μ (Eq. (7)) is set to 0.5 to weight equally the "pull" and "push" terms. The convergence threshold $\omega_{\mathcal{L}}$ is set to 0.01. The learning rates η_M and $\eta_{\mathcal{Z}}$ are set to 0.01, while the learning rate threshold ω_η is set to 10^{-7}. The growth and reduction rates σ_g and σ_r (see Algorithm 1) are set to 0.05 and 0.5, respectively. To include enough neighborhood information for training and avoid over-fitting, the number of target neighbors is set to two, three, and five for ETH-80, CHG, and YTF, respectively. The regularization parameters λ_1 and λ_2 (Eq. (15)) are set to 10. A nearest neighbor classifier is used in test.

Table 1. Summary of results for Experiment 1. The values denote the average rank-1 identification accuracy (%).

Method	ETH-80	CHG	YTF		
			LBP	CSLBP	FPLBP
RNP	78.00	35.74	52.24	45.52	**53.13**
SSDML	80.50	36.79	56.42	51.79	52.39
ISCRC	65.00	34.88	52.24	39.40	48.21
SPML	**86.00**	**40.17**	**62.39**	**52.54**	50.90

ISCRC [25]: This algorithm was selected because it is one of the most recent methods for image set classification using hull-based models. It is also the only method that reduces the number of samples used to span the hull. The authors offer online code [25]. The default settings were used as mentioned in the original paper.

SSDML [24]: This is one of the most recent algorithms on image set classification that employs distance metric learning for set-to-set classification. The code is available at the authors' website [24]. The regularization parameters λ_1 and λ_2 are set to 10 as in our SPML. The number of similar sets is set to three, five, and three for ETH-80, CGH, and YTF, respectively. The number of dissimilar sets is set to 30 for all the datasets. A nearest neighbor classifier is used in the test.

RNP [23]: This algorithm is used to model the distance for the aforementioned methods. Hence, its performance is used as a baseline in our experiments. A closed-form implementation is provided by Zhu *et al*. [24] online. The regularization parameters λ_1 and λ_2 are set to 10. A nearest neighbor classifier is used in the test.

4.3. Experimental Results

Experiment 1: The objective of this experiment is to compare the classification performance of SPML with state-of-the-art approaches. Specifically, for all datasets the number of prototypes used to represent each gallery set was set to 10. To reduce the computational cost on YTF the samples per set in the probe were reduced to 100 using k-means clustering. For ETH-80 and CHG, the original probe sets were used. An overview of the results is offered in Table 1. As illustrated, SPML appears to outperform all methods for ETH-80, CHG, and two of three features for YTF. RNP learns an unsupervised distance and does not fully utilize the labels of the training data. SSDML learns a distance metric, but the reduction in the number of samples per set appears to degrade its performance. ISCRC uses dictionary learning to compress the image set. However, the fitting

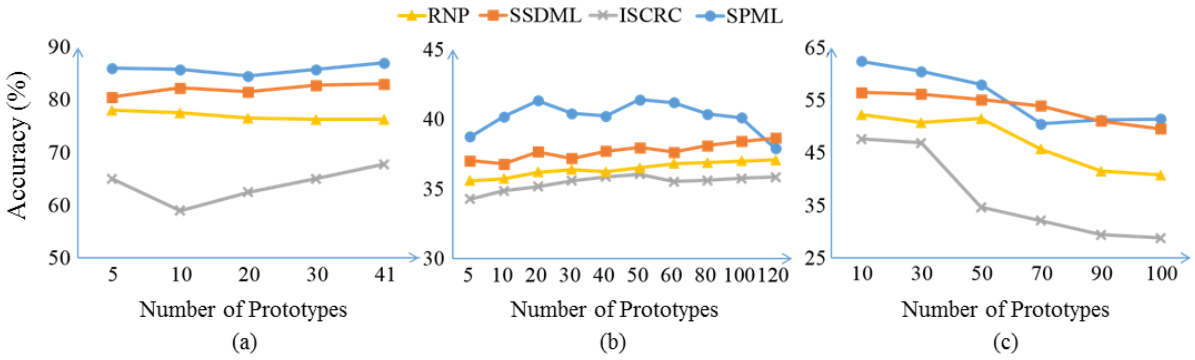

Figure 2. Average rank-1 identification accuracy using different number of prototypes (Experiment 2): (a) results obtained for ETH-80, (b) results obtained for CHG, and (c) results obtained for YTF using LBP features.

does not appear to work well and the resulting identification accuracy is the lowest. Finally, SPML utilizes the training data more effectively to compress the input information into fewer prototypes. The inherent ability to perform a reduction in the number of samples per gallery set gives the edge to SPML over other methods.

Experiment 2: The objective of this experiment is to assess the impact of the number of prototypes used on the identification performance. For ETH-80 and CHG, the number of prototypes used ranges from five to the maximum possible. For YTF, the corresponding number of prototypes ranges from 10 to 100. The LBP feature was used because it yielded the best accuracy in Experiment 1 for three out of four algorithms. Some image sets contain fewer samples than the target number of prototypes to be learned. In such cases, the number of prototypes was set to the number of samples in the original set. An overview of the results is depicted in Fig. 2. For ETH-80, SPML outperforms all methods in all cases. Its performance appears to be robust when different numbers of prototypes are used. CHG contains more image sequences and larger variations, which allows us to study the impact of the number of prototypes in a more challenging setting. SPML ranks first in all but one case (*i.e.*, when the number of prototypes is set to 120). Using all the samples in the image set seems to result in over-fitting. However, our method ranks second and improves the baseline performance (*i.e.*, RNP). YTF contains more subjects and offers more reliable evidence concerning the expected sensitivity of our method when the number of prototypes is varied. The typical image set contains 200 to 300 images. As illustrated, all methods appear to work better when fewer prototypes are used. Specifically, for 50 or more prototypes the performance of all the algorithms decreases. This indicates that the information in large scale image sets is redundant for identification tasks. The best performance is achieved by our method when the number of prototypes is set to 10. In all other cases it ranks first or second. In summary, SPML appears to be robust in the number of prototypes used for small scale image set identification. It can compress the information using few prototypes for large scale applications yielding increased accuracy while reducing the computational cost.

Experiment 3: The objective of this experiment is to assess the impact of outliers on the identification performance for set-to-set matching. We define three protocols similar to the work of Cevikalp *et al*. [3]: (i) outliers induced in the gallery set (OG), (ii) outliers induced in the probe (OP), and (iii) outliers induced in both (OGP). The *original* results correspond to the performance obtained when no outliers are induced. For all protocols, 5% of outlier samples were added to the corresponding image sets by randomly sampling from other classes. This experiment was conducted using ETH-80, setting the number of prototypes to 10 (as in Experiment 1) and 43 (*i.e.*, full image set plus outliers). The obtained results are depicted in Fig. 3. SPML outperforms all algorithms in all cases. The accuracy appears to marginally drop when the outliers are added in the setting with the 43 prototypes (see left part of Fig. 3). However, the relative changes are much smaller for SPML compared to those of ISCRC and SSDML. Our understanding is that SPML learns more robust prototypes. When 10 prototypes are used, the accuracy of SPML appears to remain the same for OG and OGP (see Fig. 3(b)). Similarly, the performance of SSDML remains almost the same when the outliers are induced in the gallery. In summary, SPML appears to be robust in the presence of outliers, especially when fewer prototypes are used to represent the gallery image sets.

Experiment 4: The objective of this experiment is to assess the impact of different initialization approaches on the identification accuracy. Random sampling and k-means clustering were used as two different initialization strategies. The YTF dataset was selected because it contains richer information compared to the other datasets. The results for ISCRC are omitted as it uses its own strategy to reduce the number of samples. The numbers of prototypes were set to 10, 30, and 50 to analyze the impact of initial-

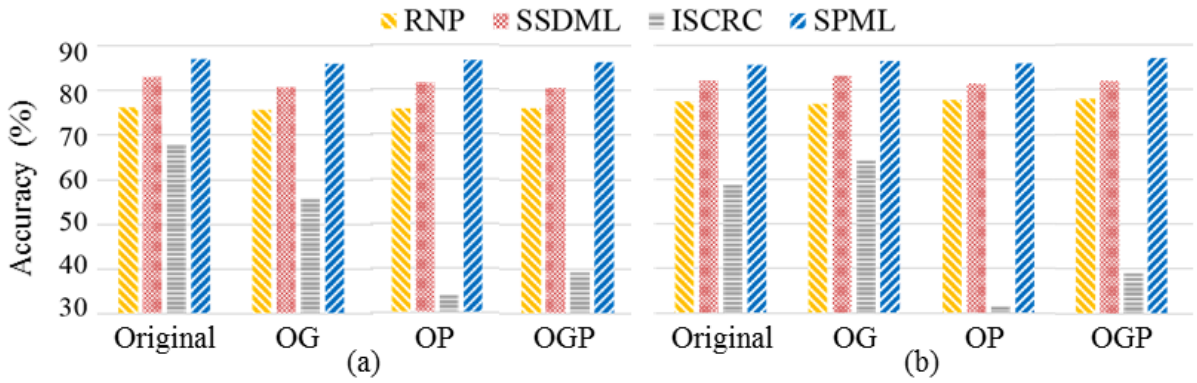

Figure 3. Average rank-1 identification accuracy obtained for different settings (Experiment 3). *Original* denotes the group of results obtained with no outliers. *OG*, *OP* and *OGP* denote the group of results obtained when there are outliers in the gallery, probe, and both gallery and probe, respectively. (a): Results obtained with full image set. (b): Results obtained with 10 prototypes per set.

ization under different settings. An overview of the results is offered in Table 2. Note that SPML always outperforms the other methods regardless of the initialization approach used. Also, the results for RNP and SSDML using k-means clustering always outperform those obtained from random sampling. This is expected as the random sampling strategy results in loss of important information. SPML learns optimal prototypes in the training phase and fully utilizes the available information. Consequently, it achieves comparable results for both initialization approaches. When the number of prototypes is small, k-means clustering appears to offer a better initialization.

Experiment 5: The objective of this experiment is to evaluate the impact of the number of prototypes on the test time cost, which is vital for real-time applications. The obtained results for YTF are reported in Table 3 when 10, 30, 70, and 90 prototypes are used. ISCRC needs to reconstruct each probe using the gallery, which greatly increases the time cost. The other methods rely on the same framework and any differences occur due to different programming practices and the different distribution of the prototypes and distance metrics learned. We observe that by reducing the number of prototypes from 90 to 10 results in a reduction of the test time cost by 50%. As illustrated in Experiment 2, using 10 prototypes yields the best identification accuracy. Hence, we conclude that SPML has the potential to increase identification accuracy and significantly reduce the test time cost.

Table 2. Summary of results for Experiment 4. The values denote average rank-1 identification accuracy (%) for YTF using the LBP features.

Initialization	Method	Number of Prototypes		
		10	30	50
k -means Clustering	RNP	52.24	50.75	51.49
	SSDML	56.42	56.12	55.07
	SPML	**62.39**	**60.45**	**57.91**
Random Sampling	RNP	48.96	49.85	49.85
	SSDML	55.07	55.07	54.78
	SPML	**59.25**	**60.90**	**58.66**

5. Conclusion

In this paper, we proposed a method that jointly learns a reduced number of prototypes and a distance metric for image set classification. As demonstrated, the proposed approach can fully utilize the training data to compress the image set, while learning a distance metric tailored to set-to-set matching. The experimental results indicate that SPML can use a few prototypes to represent each image set. Hence, it reduces the storage requirements and test time cost, while improving the identification accuracy. The corresponding sensitivity analysis indicates that our method is robust to the number of prototypes used, presence of outliers in the gallery and probe, and the prototypes initialization strategy. Joint prototype and distance metric learning

Table 3. Summary of results for Experiment 5. The values denote the average testing time (*s*) for YTF using the LBP features.

Method	Number of Prototypes			
	10	30	70	90
RNP	**50.88**	69.55	95.97	116.32
SSDML	63.03	74.06	116.13	130.13
ISCRC	294.80	847.62	917.19	948.55
SPML	53.32	**63.98**	**86.31**	**115.11**

for set-to-set identification can be employed with other hull models and distance metric learning objective functions. Despite the many advantages, the current form of SPML addresses closed-set identification only. In future work, we plan to extend it to address the tasks of verification and open-set identification.

6. Acknowledgments

This research was funded in part by the US Army Research Lab (W911NF-13-1-0127) and the UH Hugh Roy and Lillie Cranz Cullen Endowment Fund. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of the sponsors.

References

[1] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *Proc. CVPR*, pages 581–588, San Diego, CA, June 20-25 2005.

[2] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *Proc. ICCV*, pages 2408–2415, Sydney, Australia, December 3-6 2013.

[3] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *Proc. CVPR*, pages 2567–2573, San Francisco, CA, Jun. 13-18 2010.

[4] S. Chen, C. Sanderson, M. T. Harandi, and B. C. Lovell. Improved image set classification via joint sparse approximated nearest subspaces. In *Proc. CVPR*, pages 452–459, Portland, Oregon, June 25-27 2013.

[5] a. J. F. G. Shakhnarovich and T. Darrell. Face recognition from long-term observations. In *Proc. ECCV*, pages 851–865, Copenhagen, Denmark, May 27 - June 2 2002.

[6] J. Hamm and D. Lee. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *Proc. ICML*, pages 376–383, Helsinki, Finland, July 5-9 2008.

[7] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *Proc. CVPR*, pages 2705–2712, Colorado Springs, CO, June 21-23 2011.

[8] N. J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear algebra and its applications*, 103:103–118, 1988.

[9] Y. Hu, A. Mian, and R. Owens. Face recognition using sparse approximated nearest points between image sets. *PAMI*, 34(10):1992 – 2004, 2012.

[10] Z. Huang, R. Wang, S. Shan, and X. Chen. Hybrid euclidean-and-riemannian metric learning for image set classification. In *Proc. ACCV*, pages 1–8, Singapore, Nov. 1-5 2014.

[11] T. Kim, S. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *Proc. CVPR*, pages 1–8, Minneapolis, MN, Jun. 2007.

[12] T.-K. Kim, O. Arandjelovic, and R. Cipolla. Boosted manifold principal angles for image set-based recognition. *PR*, 40(9):2475–2484, 2007.

[13] M. Kostinger, P. Wohlhart, P. Roth, and H. Bischof. Joint learning of discriminative prototypes and large margin nearest neighbor classifiers. In *Proc. ICCV*, pages 3112–3119, Sydney, Australia, Dec. 3-6 2013.

[14] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proc. CVPR*, pages II–409, Madison, WI, June 18-20 2003.

[15] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *Proc. ICCV*, pages 329–336, Sydney, Australia, December 3-6 2013.

[16] M. Uzair, A. Mahmood, A. Mian, and C. McDonald. A compact discriminative representation for efficient image-set classification with application to biometric recognition. In *Proc. ICB*, pages 1–8, Madrid, June 4-7 2013.

[17] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. CVPR*, pages 2496–2503, Providence, Rhode Island, June 16-21 2012.

[18] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *Proc. CVR*, pages 1–8, Anchorage, AK, June 23-28 2008.

[19] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proc. NIPS*, pages 1473–1480, Vancouver, Canada, December 4-7 2006.

[20] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. CVPR*, pages 529–534, Providence, RI, June 20–25 2011.

[21] Y. Wu, M. Minoh, and M. Mukunoki. Collaboratively regularized nearest points for set based recognition. In *Proc. BMVC*, pages 1–8, Bristol, UK, Sept. 9-13 2013.

[22] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Proc. FG*, pages 318 – 323, Nara, Japan, Apr. 14–16 1998.

[23] M. Yang, P. Zhu, L. Van Gool, and L. Zhang. Face recognition based on regularized nearest points between image sets. In *Proc. FG*, pages 1–7, Shanghai, China, April 22-26 2013.

[24] P. Zhu, L. Zhang, W. Zuo, and D. Zhang. From point to set: extend the learning of distance metrics. In *Proc. ICCV*, pages 2664 – 2671, Sydney, Australia, December 3-6 2013.

[25] P. Zhu, W. Zuo, L. Zhang, S. Shiu, and D. Zhang. Image set based collaborative representation for face recognition. *TIFS*, 9(7):1120 – 1132, 2014.