Joint Prototype and Metric Learning for Image Set Classification: Application to Video Face Identification

Mengjun Leng, Panagiotis Moutafis, Ioannis A. Kakadiaris*

Computational Biomedical Lab
Department of Computer Science, University of Houston
4800 Calhoun Road, Houston, TX, 77004

Abstract

In this paper, we address the problem of image set classification, where each set contains a different number of images acquired from the same subject. In most of the existing literature, each image set is modeled using all its available samples. As a result, the corresponding time and storage costs are high. To address this problem, we propose a joint prototype and metric learning approach. The prototypes are learned to represent each gallery image set using fewer samples without affecting the recognition performance. A Mahalanobis metric is learned simultaneously to measure the similarity between sets more accurately. In particular, each gallery set is represented as a regularized affine hull spanned by the learned prototypes. The set-to-set distance is optimized via updating the prototypes and the Mahalanobis metric in an alternating manner. To highlight the importance of representing image sets using fewer samples, we analyzed the corresponding test time complexity with respect to the number of images used per set. Experimental results using YouTube Celebrity, YouTube Faces, and ETH-80 datasets illustrate the efficiency on the task of video face recognition, and object categorization.

Keywords: Image Set Classification, Metric Learning, Prototype Learning, Video Face Recognition
2015 MSC: 00-01, 99-00

1. Introduction

Image set classification has been an active research field for more than twenty years [1, 2, 3, 4, 5, 6, 7, 8, 9]. The task is to assign each probe image set to its corresponding gallery subject. Templates stored in the gallery are also sets of images. Both gallery and probe sets contain various numbers of images, describing the same subject. In the

*Corresponding author
Email address: ioannisk@uh.edu (Ioannis A. Kakadiaris)

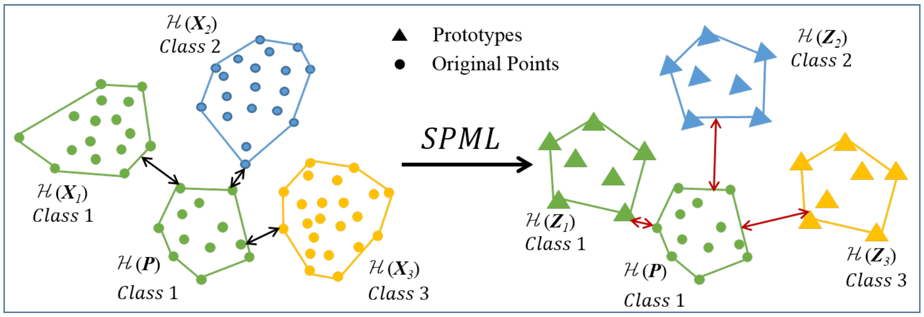
$\mathcal{H}(\boldsymbol{X}_2)$
Class 2
▲ Prototypes
● Original Points
$\mathcal{H}(\boldsymbol{Z}_2)$
Class 2
$\mathcal{H}(\boldsymbol{X}_1)$
Class 1
SPML
$\mathcal{H}(\boldsymbol{Z}_1)$
Class 1
$\mathcal{H}(\boldsymbol{P})$
Class 1
$\mathcal{H}(\boldsymbol{X}_3)$
Class 3
$\mathcal{H}(\boldsymbol{P})$
Class 1
$\mathcal{H}(\boldsymbol{Z}_3)$
Class 3

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 gallery sets from three different classes, 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 for the corresponding gallery sets. As illustrated, there are fewer samples in prototype presentation. After the process of SPML, distances between similar sets are "smaller", while the distances between dissimilar sets are "larger".

field of biometrics, many applications can be formulated as an image set classification problem, such as video-based face recognition [1], gesture recognition [10], and person re-identification across camera networks [2]. Compared with traditional single image classification, the set-based approach provides richer information with multiple samples. Hence, more reliable results are expected. However, it also introduces several new challenges: First, not all the information provided is useful for the task at hand. There is information redundancy or even noise, especially for large scale image sets. Second the within-set variations are large (e.g., different views, illumination conditions, sensors). As a result, building a proper model is crucial. Third, the computational and storage cost are increased significantly with the rapid growth of data to be processed. For example, some videos could be thousands of frames long.

To solve the image set classification problem, a straightforward approach is to model the set-to-set distance. The smaller the distance is, the more similar two image sets will be. According to how this distance is modeled, existing literature can be grouped into three categories: (i) subspace model, (ii) statistical model, and (iii) affine hull model.

Subspace Model: Methods in this category can be further grouped into two sub-categories: single subspace model and multi-subspace model. In the single subspace model, each image set is modeled as a single linear subspace [11, 3, 4, 12, 5] and can be treated as a point on a Grassmann manifold [3, 4]. Different mutual subspace distances were defined based on the principal angles between subspaces. Linear discriminative analysis [11], non-linear manifold kernels [3, 4], sparse dictionary learnng [12], and direct manifold-to-manifold mappings [5] are employed to optimize the distances. However, the single subspace model cannot reflect the importance of different local variations under different scenarios. In the multi-subspace model, each image set is modeled as a mixture of several subspaces [6, 7, 13, 8]. These subspaces can be constructed using clustering algorithms (e.g., k-means clustering [6], hierarchical agglomerative clustering [7], and Maximum Linear Patches [13, 8]). The set-to-set distance is defined as the distance between the closest pair of local subspaces. It can represent the complex local variations in a better way. However, computing a multi-subspace model is very expensive and a large amount of data is needed.

Statistical Model: Statistical characteristics are used to model the image sets. It can be further divided into two sub-categories: parametric and non-parametric. In the parametric statistical model, an image set is either modeled as a single Gaussian distribution [9] or a mixture of Gaussian [14]. The Kullback-Leibler divergence [9] or kernel based distance [14] are used to measure the distance between two sets. Methods in this category make strong assumptions concerning the distribution of the data which may not always be true. In non-parametric statistical model, each image set is described using its statistical properties: mean [15, 16], covariance matrix [17, 16], and other higher order statistics [15]. The distance is measured either in a Euclidean space [15, 16] or on a Riemannian manifold [17, 16]. The manifold-to-manifold dimensionality reduction [18] was developed to reduce the cost of computing a high dimensional Riemannian manifold. Multi-metric learning [15, 16] was employed to combine different properties together. The non-parametric statistic model relies only on a few statistical properties. As a result, it is robust, but may ignore significant local variation in the data.

Affine Hull Model: Each image set is modeled as an affine hull [19] or different kinds of reduced affine hull [19, 20, 21]. The geodesic distance between two hulls is then employed to measure the dissimilarity between sets. Mahalanobis metric is employed [22] for a more accurate dissimilarity measurement. More recently, the correlations between different gallery sets were taken into consideration [2, 23]. Although the hull-based approaches have a better tolerance on intra-class variation, the global data structure is weakly characterized. In addition, it is computationally expensive, especially when there is a large number of images in each set. In summary, even though there is a plethora of algorithms developed to address the image set classification problem, most of them only focus on exploring more discriminative similarity measurements. Very few efforts were focused on reducing high time/storage cost and information redundancy introduced by large scale image sets.

To address this gap, we extend the method of Köstinger *et al.* [24] to set-to-set matching and propose the Set-based Prototype and Metric Learning framework (SPML). Groups of discriminative prototypes and a Mahalanobis metric are jointly learned for image set classification. The prototype learning seeks to represent the gallery image sets with fewer templates, while maintaining or improving the recognition performance. The metric learning seeks to tailor a more accurate set-to-set similarity measurement based on the learned prototypes. We formulate the learning problem in a single loss function, and optimize the prototypes and Mahalanobis metric simultaneously. After processing by our SPML, a probe image set lies closer to those gallery prototype sets from the same subject, and further from those gallery prototype sets from different subjects, as illustrated in Fig. 1.

Parts of this work has appeared in our conference version [25]. In this paper, we offer three major extensions: (i) we present a time complexity analysis on existing distance models to highlight our motivation; (ii) we provide more detailed discussions and comparisons with methods from different categories; (iii) we include additional sensitivity analyses carried out to explore different aspects of the proposed algorithm.

The rest of the paper is organized as follows: In Sec. 2, we discuss related works.

Table 1: Overview of the notations used in this paper. Matrices are denoted by bold upper-case letters, vectors by bold lower-case letters, sets by bold upper-case letters in Calligraphy, and scalars by non-bold English or Greek letters. The table fits in camera ready version.

Symbol	Description
$\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, ..., \boldsymbol{x}_{i,N_i}] \in \mathbb{R}^{d \times N_i}$	the i^{th} image set containing N_i images
$\boldsymbol{x}_{i,m} \in \mathbb{R}^d$, $m \in [1, N_i]$	feature vector of the m^{th} image in image set $\boldsymbol{X}_i$
$\boldsymbol{Z}_i = [\boldsymbol{z}_{i,1}, \boldsymbol{z}_{i,2}, ..., \boldsymbol{z}_{i,K}] \in \mathbb{R}^{d \times K}$	the prototype representation of $\boldsymbol{X}_i$, containing K prototypes
$\boldsymbol{z}_{i,k} \in \mathbb{R}^d$, $k \in [1, K]$	feature vector of the k^{th} prototype
$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_{N_p}] \in \mathbb{R}^{d \times N_p}$	a probe image set containing N_p images
$\boldsymbol{p}_m \in \mathbb{R}^d$, $m \in [1, N_p]$	feature vector of the m^{th} image in probe image set
$\boldsymbol{\mathcal{G}} = \{(\boldsymbol{X}_i, y_i) \mid i \in [1, N]\}$	a gallery with N image sets, where y_i is the label of $\boldsymbol{X}_i$
$\boldsymbol{\mathcal{Z}} = \{(\boldsymbol{Z}_i, y_i) \mid i \in [1, N]\}$	a set containing prototype representation of all image sets in the gallery
$\boldsymbol{M} \in \mathbb{R}^{d \times d}$	the Mahalanobis matrix
$\mathcal{H}(*)$	a regularized affine hull spanned by $*$

In Sec. 3 we introduce the mathematical model of the proposed framework. In Sec. 4 we discuss the implementation of our framework and the testing time complexity. In Sec. 5 we present the experimental settings and results; In Sec. 6 we summarize the limitations of our proposed framework; Sec. 7 concludes the paper.

2. Related Work

In this section, we offer a brief introduction on algorithms that are closely related to our work. In particular, our work is built on the regularized nearest points (RNP) method [21], set-to-set distance metric learning (SSDML) [22], and the prototype learning for large margin nearest neighbor classifiers [24]. For the convenience of discussion, an overview of the notations used in this paper is summarized in Table 1.

In RNP, Yang *et al.* [21] proposed to model an image set $\boldsymbol{X}_i$ as a regularized affine hull (RAH), spanned by all its samples:

$$\mathcal{H}(\boldsymbol{X}_i) = \left\{ \boldsymbol{X}_i \boldsymbol{\alpha}_i \middle| \sum_{m=1}^{N_i} \alpha_{i,m} = 1, \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma \right\}, \tag{1}$$

with a regularization on the l_p norm of the the combination coefficient $\|\boldsymbol{\alpha}\|_{l_p} < \sigma$, where $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,m}]^T$. The distance between two image sets $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ is then defined as the geodesic distance between $\mathcal{H}(\boldsymbol{X}_i)$ and $\mathcal{H}(\boldsymbol{X}_j)$,

$$\begin{aligned} \mathcal{D}^2(\boldsymbol{X}_i, \boldsymbol{X}_j) = \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left[(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j)^T (\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{X}_j\boldsymbol{\alpha}_j) \right] \\ s.t.\ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1,\ \|\boldsymbol{\alpha}_j\|_{l_p} < \sigma_2, \sum_{m=1}^{N_i} \alpha_{i,m} = 1, \sum_{m=1}^{N_j} \alpha_{j,m} = 1, \end{aligned} \tag{2}$$

By relaxing $\sum_{m=1}^{N_i} \alpha_{i,m} = 1$ and $\sum_{m=1}^{N_j} \alpha_{j,m} = 1$ to $\sum_{m=1}^{N_i} \approx 1$ and $\sum_{m=1}^{N_j} \alpha_{j,m} \approx 1$ and using the Lagrangian formulation, Eq. (2) with $l_p = 2$ can be integrated as

$$\begin{aligned} &\mathcal{D}^2(\boldsymbol{X}_i, \boldsymbol{X}_j) \\ &= \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left(\|\boldsymbol{u} - \hat{\boldsymbol{X}}_i \boldsymbol{\alpha}_i - \hat{\boldsymbol{X}}_j \boldsymbol{\alpha}_j\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}_i\|_2^2 + \|\boldsymbol{\alpha}_j\|_2^2 \right), \end{aligned} \tag{3}$$

where $\boldsymbol{u} = [\mathbf{0}; \mathbf{1}; \mathbf{1}]$, $\hat{\boldsymbol{X}}_i = \left[\boldsymbol{X}_i; \mathbf{1}^T; \mathbf{0}^T\right]$, $\hat{\boldsymbol{X}}_j = \left[-\boldsymbol{X}_j; \mathbf{0}^T; \mathbf{1}^T\right]$, and the column vectors $\mathbf{0}$ and $\mathbf{1}$ have the appropriate sizes associated with their corresponding context. Although the regularization can effectively restrict the expansion of the hull area, the natural geodesic distance might not reflect the dissimilarity for the task at hand properly. To tailor a more accurate set-to-set distance, Zhu *et al.* [22] extended the Mahalanobis distance metric learning [26] to the geodesic distance between hulls:

$$\begin{aligned} &\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{X}_j) = (\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{X}_j \hat{\boldsymbol{\alpha}}_j)^T \boldsymbol{M} (\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{X}_j \hat{\boldsymbol{\alpha}}_j) \\ &(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\alpha}}_j) = arg \min_{\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j} \left[(\boldsymbol{X}_i \boldsymbol{\alpha}_i - \boldsymbol{X}_j \boldsymbol{\alpha}_j)^T \boldsymbol{M} (\boldsymbol{X}_i \boldsymbol{\alpha}_i - \boldsymbol{X}_j \boldsymbol{\alpha}_j) \right] \\ &s.t. \ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1, \ \|\boldsymbol{\alpha}_j\|_{l_p} < \sigma_2, \sum_{m=1}^{N_i} \alpha_{i,m} = 1, \sum_{m=1}^{N_j} \alpha_{j,m} = 1, \end{aligned} \tag{4}$$

where $\boldsymbol{M}$ is a positive semi-definite matrix to be learned. It can be learned using any distance metric learning model. In both RNP and SSDML, the restricted affine hull is spanned by all the samples in the image set. This is not only computationally expensive, but also sensitive to outliers. In single-shot image classification, Köstinger *et al.* [24] proposed to reduce and optimize the templates used for each subject, and a distance metric is learned jointly. In this paper, we extend this idea to set-to-set matching. In particular, we build a framework, which can jointly learn a prototype representation and a Mahalanobis distance metric for the geodesic distance between hulls.

3. Method

In this section, we describe the SPML framework. Specifically, our objectives are to learn jointly: (i) a prototype representation for each gallery image set, using fewer samples, and (ii) a corresponding Mahalanobis distance metric with better discriminative property for set-to-set matching.

3.1. Mathematical Model

In the prototype representation, each gallery image set $\boldsymbol{X}_i$ is then represented as an RAH spanned by the prototypes:

$$\mathcal{H}(\boldsymbol{Z}_i) = \left\{ \boldsymbol{Z}_i \boldsymbol{\beta} \middle| \sum_{k=1}^{K} \beta_k = 1, \|\boldsymbol{\beta}\|_{l_p} < \sigma \right\}, \tag{5}$$

where $\boldsymbol{Z}_i = [\boldsymbol{z}_{i,1}, \boldsymbol{z}_{i,2}, ..., \boldsymbol{z}_{i,K}] \in \mathbb{R}^{d \times K}$ denotes for the prototype set containing K prototypes($K < N_i$). To distinguish between the representation coefficients of the original RAH $\mathcal{H}(\boldsymbol{X}_i)$, $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_K]^T$ is used to denote the representation coefficients of the prototype RAH $\mathcal{H}(\boldsymbol{Z}_i)$. The prototypes and the Mahalanobis distance metric are optimized by minimizing a loss function across the whole gallery:

$$\begin{aligned}(\boldsymbol{\mathcal{Z}}, \boldsymbol{M}) &= arg \min_{\boldsymbol{\mathcal{Z}}, \boldsymbol{M}} \mathcal{L}\left(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{Z}}, \boldsymbol{M}\right) \\ &= arg \min_{\boldsymbol{\mathcal{Z}}, \boldsymbol{M}} \sum_{\boldsymbol{\mathcal{G}}} \mathcal{L}_i(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}}, \boldsymbol{M}).\end{aligned} \tag{6}$$

The proposed loss function $\mathcal{L}$ is a variant of the Large Margin Nearest Neighbors (LMNN) approach [27], and the loss on each gallery set $\boldsymbol{X}_i$ is defined as:

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

The objective of the first term is to pull target neighbors (i.e., $\boldsymbol{Z}_j$) “closer”, where target neighbors denote the k-nearest prototype sets to $\boldsymbol{X}_i$ and labeled as y_i. All the indices of the target neighbors are contained in $\boldsymbol{\mathcal{S}}_i$. The objective of the second term is to push impostors of $\boldsymbol{X}_i$ (i.e., $\boldsymbol{Z}_l$) “far away”, where $\boldsymbol{\mathcal{V}}_i = \{(j, l) | j \in \boldsymbol{\mathcal{S}}_i$ and $y_l \neq y_i\}$,
$[x]_+ = max(x, 0)$. The trade-off between the pull and push terms is determined by $\mu \in [0, 1]$. The LMNN function was selected due to its robustness. Other loss functions could have been selected instead. The distance used in Eq. (7) is the Mahalanobis distance between restricted affine hulls, specifically,

$$\begin{aligned}&\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) = (\boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{\beta}}_j)^T \boldsymbol{M} (\boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{\beta}}_j) \\ &(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_j) = arg \min_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j} \left[(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)^T \boldsymbol{M} (\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)\right] \\ &s.t. \ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1, \ \|\boldsymbol{\beta}_j\|_{l_p} < \sigma_2, \sum_{m=1}^{N_i} \alpha_{i,m} = 1, \sum_{k=1}^{K} \beta_{j,k} = 1.\end{aligned} \tag{8}$$

However, Mahalanobis distance under other hull models [19, 20] can also be employed instead.

3.2. Optimization

The prototype gallery $\boldsymbol{\mathcal{Z}}$ and the Mahalanobis matrix $\boldsymbol{M}$ are optimized via solving Eq. (8) in an EM-like manner. Specifically, gradient descent is employed to update $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{M}$ in an alternating manner.

$\boldsymbol{M}$ *Step*: In this step, $\boldsymbol{M}$ is updated using gradient descent with the prototype $\boldsymbol{\mathcal{Z}}$ fixed. At the $(t+1)^{th}$ iteration, the $\boldsymbol{M}$ is then updated via

$$\boldsymbol{M}^{t+1} = \boldsymbol{M}^t - \eta_{\boldsymbol{M}} \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{M}^t}, \tag{9}$$

where $\eta_{\boldsymbol{M}}$ is the learning rate. The partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{M}$ is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}} &= \sum_{\boldsymbol{\mathcal{G}}} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{M}} \\ &= \sum_{\boldsymbol{\mathcal{G}}} \left[(1-\mu) \sum_{\boldsymbol{\mathcal{S}}_i} \boldsymbol{C}_{ij} + \mu \sum_{\boldsymbol{\mathcal{V}}_{i+}} (2\boldsymbol{C}_{ij} - \boldsymbol{C}_{il}) \right],\end{aligned} \tag{10}$$

where,

$$\begin{aligned}\boldsymbol{C}_{ij} &= (\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j \hat{\boldsymbol{\beta}}_j)(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j \hat{\boldsymbol{\beta}}_j)^T \\ \boldsymbol{\mathcal{V}}_{i+} &= \left\{ (j,l) \mid 2\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) - \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_l) > 0 \right\}.\end{aligned} \tag{11}$$

The representation coefficients $(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_j)$ are calculated from Eq. (8), and $\boldsymbol{\mathcal{V}}_{i+}$ is a subset of $\boldsymbol{\mathcal{V}}_i$, containing the index pairs (j,l) for which the hinge loss in $\mathcal{L}_i$ is larger than zero. To ensure that $\boldsymbol{M}$ is positive semi-definite, the updated $\boldsymbol{M}$ is projected onto its nearest positive semi-definite matrices as described in [28].

$\boldsymbol{\mathcal{Z}}$ *Step*: In $(t+1)^{th}$ iteration, each prototype set $\boldsymbol{Z}_k^{t+1} \in \boldsymbol{\mathcal{Z}}^{t+1}$ is optimized independently by:

$$\boldsymbol{Z}_k^{t+1} = \boldsymbol{Z}_k^t - \eta_{\boldsymbol{\mathcal{Z}}} \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{Z}_k}, \tag{12}$$

where $\eta_{\boldsymbol{\mathcal{Z}}}$ is the learning rate for $\boldsymbol{\mathcal{Z}}$. The partial derivative of the loss function $\mathcal{L}$ with respect to $\boldsymbol{Z}_k$ is the summation of partial derivative of loss on each gallery set:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_k} = \sum_{\boldsymbol{\mathcal{G}}} \frac{\partial \mathcal{L} i}{\partial \boldsymbol{Z}_k}. \tag{13}$$

Since $\boldsymbol{Z}_k$ is considered to be a target neighbor for some of the gallery sets $\boldsymbol{X}_i$, but an impostor for a different $\boldsymbol{X}_i$, the corresponding partial derivatives vary. Specifically, when $\boldsymbol{Z}_k$ is treated as a target neighbor (i.e., $k \in \boldsymbol{\mathcal{S}}_i \,\mathrm{and}\, (k,l) \in \boldsymbol{\mathcal{V}}_{i+}$), its partial derivative is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{Z}_k} =& -2(1-\mu) \sum_{k \in \boldsymbol{\mathcal{S}}_i} \boldsymbol{M}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_k \hat{\boldsymbol{\beta}}_k) \hat{\boldsymbol{\beta}}_k^T \\ & - 4\mu \sum_{(k,l) \in \boldsymbol{\mathcal{V}}_{i+}} \boldsymbol{M}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_k \hat{\boldsymbol{\beta}}_k) \hat{\boldsymbol{\beta}}_k^T.\end{aligned} \tag{14}$$

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

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{Z}_k} = 2\mu \sum_{(j,k) \in \boldsymbol{\mathcal{V}}_{i+}} \boldsymbol{M}(\boldsymbol{X}_i \hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_k \hat{\boldsymbol{\beta}}_k) \hat{\boldsymbol{\beta}}_k^T. \tag{15}$$

In all other cases,

$$\frac{\partial \mathcal{L} i}{\partial \boldsymbol{Z}_k} = 0. \tag{16}$$

Update Neighborhood: Once $\boldsymbol{\mathcal{Z}}$ or $\boldsymbol{M}$ has been updated, the corresponding distance and neighborhood relationship should be refined. The Mahalanobis metric $\boldsymbol{M}$ can be decomposed via Cholesky decomposition $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$. The distance between $\boldsymbol{X}_i$ and $\boldsymbol{Z}_j$ in Eq. (8) can be written as:

$$\begin{aligned}
&\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j) = \Big[\boldsymbol{L}(\boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{\beta}}_j)\Big]^T \boldsymbol{L}(\boldsymbol{X}_i\hat{\boldsymbol{\alpha}}_i - \boldsymbol{Z}_j\hat{\boldsymbol{\beta}}_j)\\
&(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_j) = arg\min_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j} \|\boldsymbol{L}(\boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{Z}_j\boldsymbol{\beta}_j)\|_2^2 \qquad (17)\\
&s.t.\ \|\boldsymbol{\alpha}_i\|_{l_p} < \sigma_1,\ \|\boldsymbol{\beta}_j\|_{l_p} < \sigma_2.
\end{aligned}$$

It is equivalent to first project $\boldsymbol{X}_i$ and $\boldsymbol{Z}_j$ into a new space defined by $\boldsymbol{L}$, and then calculates the geodesic distance between the two hulls in the projected space. Using the same trick in Eq. (3), Eq. (17) can be formulated as

$$\begin{aligned}
&\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j)\\
&= \min_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_j} \Big(\|\boldsymbol{u}' - \hat{\boldsymbol{X}}'_i\boldsymbol{\alpha}_i - \hat{\boldsymbol{Z}}'_j\boldsymbol{\beta}_j\|_2^2 + \lambda_1\|\boldsymbol{\alpha}_i\|_2^2 + \lambda_2\|\boldsymbol{\beta}_j\|_2^2\Big), \qquad (18)
\end{aligned}$$

where $\boldsymbol{u} = [\ \boldsymbol{0};\ \boldsymbol{1};\ \boldsymbol{1}\ ]$, $\hat{\boldsymbol{X}}'_i = \big[\ \boldsymbol{L}\boldsymbol{X}_i\ ;\ \boldsymbol{1}^T\ ;\ \boldsymbol{0}^T\ \big]$, and $\hat{\boldsymbol{Z}}'_j = \big[-\boldsymbol{L}\boldsymbol{Z}_j; \boldsymbol{0}^T; \boldsymbol{1}^T\big]$. The column vectors $\boldsymbol{0}$ and $\boldsymbol{1}$ have the appropriate sizes associated with their corresponding contexts. Eq. (18) has a closed-form solution. It can also be solved using the fast solver in [21] to update $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{Z}_j)$. Once $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$ has been updated, the neighborhood relationships $(\boldsymbol{\mathcal{S}}_i, \boldsymbol{\mathcal{V}}_i)$ can be refined accordingly.

4. Discussion

In this section, we first discuss the implementation of the training and testing procedures for the proposed framework; and then analyze the testing time complexity, with respect to the number of images per gallery set.

4.1. Implementation

Training: The training process is conducted to optimize prototype $\boldsymbol{\mathcal{Z}}$ and the Mahalanobis matrix $\boldsymbol{M}$. An overview of the training procedure is offered by Algorithm 1.

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

Line 3 (Convergence criteria): 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 (Adaptive Learning rate): If the update overshoots (i.e., $\mathcal{L}^{t+1} >$

Algorithm 1 Set-based Prototypes and Metric Learning

```
Input: $\mathcal{G}$
Output: $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{M}$
Initialize $\boldsymbol{M}_0$, $\boldsymbol{\mathcal{Z}}_0$, $\eta_{\boldsymbol{M}}$, $\eta_{\boldsymbol{\mathcal{Z}}}$
procedure $(\boldsymbol{\mathcal{Z}}, \boldsymbol{M})$ =SPML($\mathcal{G}$)
    while convergence criterion is not met do
        while $\mathcal{L}^{t+1} > \mathcal{L}^t$ do
            $\eta_{\boldsymbol{\mathcal{Z}}} = (1 - \sigma_r)\eta_{\boldsymbol{\mathcal{Z}}}$
        end while
        Update $\boldsymbol{\mathcal{Z}}$ (Eq. (13))
        $\eta_{\boldsymbol{\mathcal{Z}}} = (1 + \sigma_g)\eta_{\boldsymbol{\mathcal{Z}}}$
        Update $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$ (Eq. (18))
        while $\mathcal{L}^{t+1} > \mathcal{L}^t$ do
            $\eta_{\boldsymbol{M}} = (1 - \sigma_r)\eta_{\boldsymbol{M}}$
        end while
        Update $\boldsymbol{M}$ (Eq. (9))
        $\eta_{\boldsymbol{M}} = (1 + \sigma_g)\eta_{\boldsymbol{M}}$
        Update $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{\mathcal{Z}})$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{V}}_i$ (Eq. (18))
    end while
end procedure
```

$\mathcal{L}^t$), 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.

Testing: In the testing, a probe image set $\boldsymbol{P}$ is compared with all the prototype sets in the gallery, and calculate their distances $\mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{P}, \boldsymbol{Z}_i), i \in [1, N]$. The probe is classified to

$$y_i = arg \min_i \mathcal{D}^2_{\boldsymbol{M}}(\boldsymbol{P}, \boldsymbol{Z}_i), \tag{19}$$

the same subject with its closest prototype gallery set.

4.2. Testing Time Complexity

We compare the testing time complexity of the proposed SPML with its most related hull based algorithms: SSDML and RNP. The elementary operation of the testing process (Eq. (19)) is calculating the distance between a gallery set and a probe set.

RNP: There are two ways to compute the distance between two sets (Eq. (3)): (i) a closed-form solution, and (ii) a fast solver. For the closed-form solution, the time complexity is $\mathcal{O}\left((N_i + N_q)^3\right)$, where N_i and N_q denote the number of images in the current gallery set and probe image set, respectively. For the alternate fast solver, the time complexity is $\mathcal{O}\left(dl(N_i + N_q)\right)$, where l is the number of iterations and d is the feature dimensionality.

SSDML: An extra step of mapping features to the learned space is added to RNP. Its

Table 2: A summary of the datasets used in the experiments. Numbers listed in the table are computed based on our protocol and processing. There are some differences from the statistics of the original release that are explained in Sec. 5.1. The table fits in camera ready version.

Dataset	Task	Num. of Subjects	Num. of Image Sets per Gallery Sub./Obj.	Num. of Probes	Num. of images per Sets
ETH-80	Object Categorization	8	5	40	41
YTC	Video Face Identification	47	5	$1,621$	$13-349$
YTF	Video Face Identification	59	4	67	$48-2{,}157$

time complexity is $\mathcal{O}\left(d^3(N_i+N_q)\right)$. The overall time complexity is $\mathcal{O}\left((N_i+N_q)^3\right)$ for the closed-form solution and $\mathcal{O}\left((d^3+dl)(N_i+N_q)\right)$ for the fast solver.

SPML: The distance between two sets is computed in the same manner with SSDML, by replacing the gallery set with the prototype set. The overall time complexity reduce to $\mathcal{O}\left((K+N_q)^3\right)$ and $\mathcal{O}\left((d^3+dl)(K+N_q)\right)$ for the closed-form solution and the fast solver, respectively, where the number of prototypes per set K is smaller than N_i. Based on the analysis above, the testing time is increasing linearly or in a cubic manner with the number of images per gallery set. Representing the gallery image set using fewer samples will reduce the testing time significantly.

5. Experiments

In this section, we designed experiments to evaluate the proposed SPML framework. The following information is provided: (i) datasets, (ii) baselines with corresponding settings, and (iii) experimental results.

5.1. Datasets

The ETH-80 [29], YouTube Celebrity (YTC) [30] and YouTube Face (YTF) [31] datasets were selected to assess the performance of the proposed SPML in object categorization and video-based face identification. The basic information about the employed datasets is summarized in Table 2.

ETH-80: This dataset comprises objects from 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. In all the experiments, five objects are randomly sampled from each category to form the gallery, while the rest are used as probes. We repeat the random spliting of gallery and probe ten times and report the average performance.

YTC: This dataset contains low resolution video sequences of 47 subjects from YouTube. For each subject, the number of videos provided varies from 15 to 106. Following [22], the face area is detected frame by frame, resized to 30×30, and the concatenated pixel values are used as features. For each video, the number of valid image frames (i.e., a face is detected) varies from 13 to 349. In all the experiments, four videos are randomly sampled from each subject to form the gallery, while the rest are used as probes.

Table 3: A summary of the algorithms compared in the experiments.

Algorithm	Literature Source	Category
CDL [17]	Wang *et al. CVPR' 12*	Statistical
GDL [12]	Harandi *et al. ICCV' 13*	Subspace
RNP [21]	Yang *et al. FG' 13*	Hull
SSDML [22]	Zhu *et al. ICCV' 13*	Hull
ISCRC [23]	Zhu *et al. TIFS' 14*	Hull

We repeat the random spliting of gallery and probe ten times and report the average performance.

YTF: This dataset contains $3{,}425$ videos captured from $1{,}595$ different subjects. Our objective is to simulate a face identification task. However, for most subjects only a single video is available. As a result, these videos cannot be used to evaluate the identification performance. To this end, 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). In all the experiments, four videos are randomly sampled from each subject to form the gallery, while the rest are used as probes. We repeat the random spliting of gallery and probe ten times and report the average performance.

Feature Processing: Principal Component Analysis (PCA) is applied to all the features for two reasons: (i) to reduce the noise in the features; and (ii) to avoid the over-fitting introduced by high dimensional features. Except for RNP, all the algorithms we evaluated use either distance metric learning (i.e., CDL, SSDML, and SPML) or dictionary learning (i.e., GDL, and ISCRC). Moutafis et al. [32] have illustrated experimentally that distance metric learning algorithms suffer from overfitting when high dimensional features are being used, because the parameters that need to learn increase quadratically with the feature dimensionality. Similar reasons also apply to dictionary learning algorithms. In particular, the length of the feature vectors is arbitrarily set to 100 as default in our experiments. A sensitivity analysis on the feature length is presented in Experiment 2. To reduce large intra-class variations, these PCA reduced features were projected onto an intra-class subspace following the procedure described in [33].

5.2. Baselines and Settings

In this section, we discuss the algorithms we compared in the experiments and their corresponding parameter settings. To conduct a fair comparison, all parameters are tuned according to the instructions in the original papers. In particular, we split the gallery set into gallery and validation via random sampling one image set per subject/object. All the tuning is conducted to achieve the highest identification rate in the validation set. All the parameters are initialized using the default values. All the changes are described in the following paragraphs. A summary of the selected algorithms is provided in Table 3. In particular, we select recently published algorithms

Table 4: Summary of results for Experiment 1. The values denote the mean(%) and standard deviation(%) of rank-1 identification rate when only 10 prototypes are used.

Method	ETH-80		YTC		YTF					
					LBP		FPLBP		CSLBP	
	mean	std.	mean	std.	mean	std.	mean	std.	mean	std.
CDL-lda	82.50	6.22	60.07	4.69	36.27	4.28	31.34	2.49	35.37	4.63
CDL-pls	80.75	8.07	60.14	2.96	37.31	5.04	30.90	3.89	35.22	5.43
GDL	82.50	5.47	**64.82**	3.09	51.34	5.13	46.57	5.20	50.03	4.81
RNP	78.75	5.51	64.11	2.11	52.24	3.20	**53.13**	7.10	45.52	4.63
SSDML	83.25	5.71	61.63	2.51	56.42	5.16	52.39	3.43	51.79	4.11
ISCRC	67.50	4.74	47.84	2.71	52.24	4.69	48.21	3.60	39.40	3.80
SPML	**85.75**	6.43	61.67	3.00	**62.39**	6.11	50.90	4.40	**52.54**	2.56

from the three categories discussed in Sec. 1.

Statistical model: Covariance Discriminative Learning (CDL) [17] is selected due to its stable performance reported in the literature. In particular, it represents each image set using its covariance matrix, and the set-to-set distance is calculated on the Riemannian manifold. Two versions of implementations based on Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS) are provided by Wang *et al.* [34], referred to as CDL-lda and CDL-pls. The parameters for this method are kept the same as default settings in the code after cross-validation.

Subspace model: Grassmann dictionary learning (GDL) [12] is one of the most recent algorithms that uses the linear subspace model. In particular, it extends the dictionary learning and sparse coding into the subspace model. The implementation is provided by Harandi *et al.* online [35]. In our experiments, the orthogonal representation of linear subspace is computed using Singular Value Decomposition. In each experiment, the order of the subspace is kept the same as the number of prototypes used. The number of atoms in the dictionary is set to be 20, 150, and 232 for ETH-80, YTC, and YTF, respectively based on cross-validation. The rest of the parameters are kept the same as the default settings after cross-validation.

Hull model: Except for RNP and SDML (introduced in Sec. 2), the image set based collaborative representation and classification (ISCRC) [23] method is also included. As RNP is used to model the distance for SSDML, ISCRC, and the proposed SPML, its performance is used as a baseline in our experiments. The implementations of all above algorithms is provided by Zhu *et al.* online [36]. For RNP the regularization parameters λ_1 and λ_2 are set to 10 based on the result of cross-validation. A nearest neighbor classifier is used for testing. For SSDML, the numbers of similar and dissimilar sets are set to be three and 30, accordingly.

SPML: For our proposed algorithm, the default settings are provided here. Each prototype set is initialized using k-means clustering, while the number of prototypes used is set to 10. As an LMNN-like objective function is employed, the related parameters are set to follow the original implementation of LMNN. The trade-off parameter μ

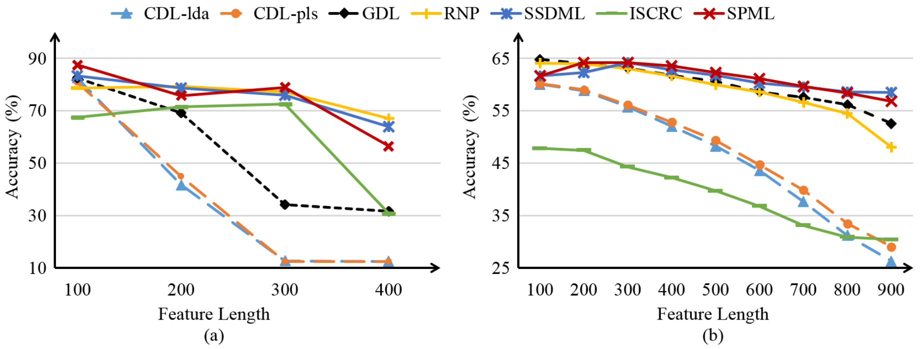

Figure 2: Average rank-1 identification accuracy obtained using different length of features (Experiment 2): (a) results obtained in the ETH-80 dataset; (b) results obtained in the YTC dataset. Results for the hull-based algorithms are presented using solid lines. Results for statistics-based algorithms are presented in short dashed lines. Results for the subspace-based algorithms are presented in long dashed lines. The feature length covers from 100 to the original length without reduction with a step of 100.

(Eq. (7)) is set to 0.5 to equally weight the "pull" and "push" terms. The convergence threshold $\omega_{\mathcal{L}}$ is set to 0.01. The learning rates $\eta_{\boldsymbol{M}}$ and $\eta_{\boldsymbol{\mathcal{Z}}}$ are initialized to 0.01. The learning rate threshold ω_η is set to 10^{-7}. The number of neighbors is set to three. The growth and reduction rates σ_g and σ_r (see Algorithm 1) are set to 0.05 and 0.5, respectively. Following the settings of we used for RNP, the regularization parameters λ_1 and λ_2 (Eq. (18)) are set to 10.

5.3. 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 is set to 10. To reduce the computational cost on YTF the samples per set in the probe are reduced to 100 using *k*-means clustering. For ETH-80 and YTC, the original probe sets are used. An overview of the results is offered in Table 4. As illustrated, SPML appears to outperform all methods for ETH-80, and two out of three features for YTF. The statistic-based CDL needs enough samples to estimate the covariance matrix. The subspace-based GDL can embed the information into a low order subspace. In the order of 10, it seems to perform worse than our proposed SPML. The RNP learns an unsupervised distance and thus 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 does not appear to work well. 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. For YTC, and FPLBP features in YTF, SPML failed to improve the baseline provided by RNP. Further analysis is provided in Experiment 2 and Experiment 6.

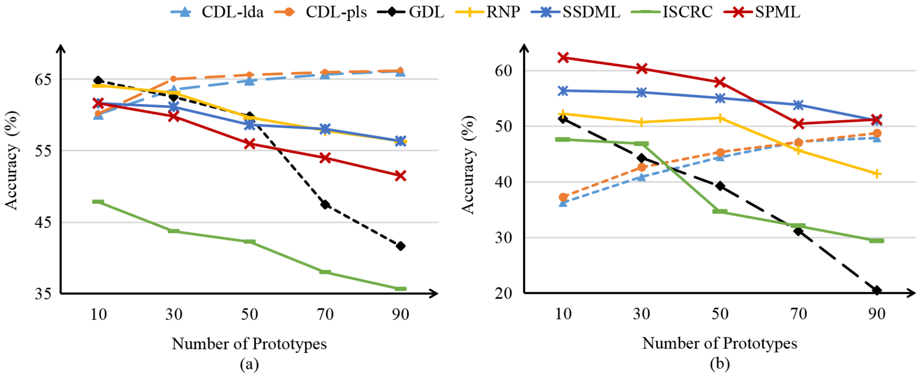

Figure 3: Average rank-1 identification accuracy obtained using different numbers of prototypes (Experiment 3): (a) results obtained in the YTC dataset; (b) results obtained in the YTF dataset using LBP features. Results for the hull-based algorithms are presented using solid lines. Results for statistics-based algorithms are presented in short dashed lines. Results for the subspace-based algorithms are presented in long dashed lines.

Experiment 2: The objective of this experiment is to assess the impact of the feature length on the identification performance. An overview of the results is depicted in Fig. 3. The ETH-80 dataset and YTC dataset are used to assess the performance. The feature length used ranges from 100 to the original length, with a step of 100. In ETH-80, the performance of all algorithms is decreasing with the increase of the feature length. There are several reasons: (i) longer feature contains more noise; (ii) longer feature corresponds to more parameters to learn for each model. It may under-fit due to lack of training data or over-fit due to the complex model. It can be also observe that the hull-based methods (i.e., solid lines) are more robust to the changes of feature length. The statistic-based methods (i.e., long dashed lines) suffer the most from high dimensional features. Similar patterns can be observed from the results obtained in YTC. The performance of SPML and SSDML starts to decrease after the feature length of 300. With a feature length of 200 and 300 SPML can achieve comparable performance with the highest one.

Experiment 3: The objective of this experiment is to assess the impact of the number of prototypes used on the identification performance. An overview of the results is depicted in Fig. 3. The results are obtained in YTC and YTF datasets, because they contain more subjects and provide larger image sets. The number of prototypes used ranges from 10 to 90 with a step of 20. All other settings keep the same with Experiment 1. In YTF, the LBP feature was used because it yielded the best accuracy in Experiment 1 for six out of seven 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. In general, the SPML achieves the highest performance using only 10 prototypes. This indicates its effectiveness in compressing the available information using few prototypes. In YTF, the SPML appears to outperform all methods in four out of five cases. In YTC, the performance

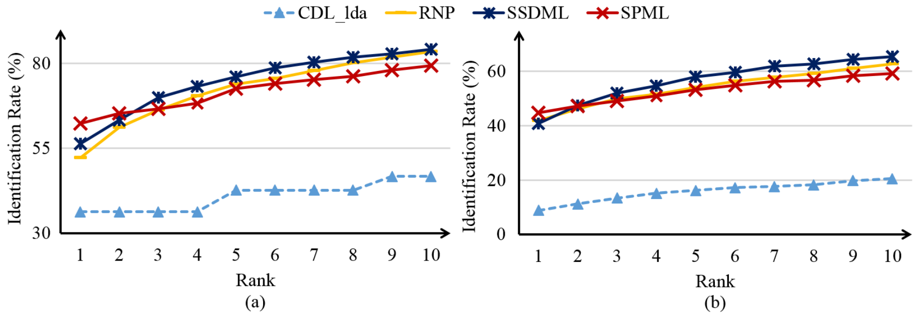

Figure 4: The CMC curves obtained with different numbers of subjects in gallery (Experiment 4): (a) results obtained in our default settings of YTF; (b) results obtained with the expanded gallery set. Results for the hull-based algorithms are presented using solid lines. Results for statistics-based algorithms are presented in short dashed lines. The classification of CDL-pls and ISCRC are not applicable for the CMC curves.

of SPML is on average when few prototypes are used. Better performance is expected with longer features as illustrated in Experiment 2. It can also be observed that the performance of CDL (short dashed lines in Fig. 3) keeps increasing with the increase of the number of prototypes used, while the performance of all other algorithms is decreasing. Since CDL relies on a statistical model, it needs enough samples to estimate the covariance matrix. The subspace-based GDL (the long dashed line in Fig. 3)is not suitable to represent an image set using a high order subspace. For hull-based SPML, ISCRC, SSDML, and RNP (solid lines in Fig. 3), spanning a large amount of vectors would cause overlap of inter-class hulls. It may also result in over-fitting for SPML when a large number of prototypes is used.

Experiment 4: The objective of this experiment is to assess the impact of different numbers of subjects in the gallery. In particular, the cumulative match characteristic curve (CMC) is employed to assess the performance. The gallery set is expanded by add some of the removed subjects which contain four videos. All these samples are added to the gallery set without matching samples in the query set. This is due to the restriction that three neighbors are needed in our training process. Therefore, four videos per subject should be ensured in the gallery for training. A summary of the results is depicted in Fig. 4. The CDL-pls and ISCRC are not applicable for CMC. Their classification is based on all samples from a certain class instead of a single sample. As indicated, the performance of all algorithms drops after expanding the gallery set. Because it increasing the possibility of a wrong match. We can also observe that the proposed SPML can only outperform RNP and SSDML before rank 3. One of the reason is that we only take into consideration the three nearest neighbors in our objective functions. As a result, it can only optimize the first three recalls. This is one of the limitations of our proposed algorihtm. It can be addressed by embedding other distance metric learning objective functions without the local neighborhood constraints.

Experiment 5: The objective of this experiment is to assess the impact of different initialization approaches on the identification accuracy. In particular, we compare two

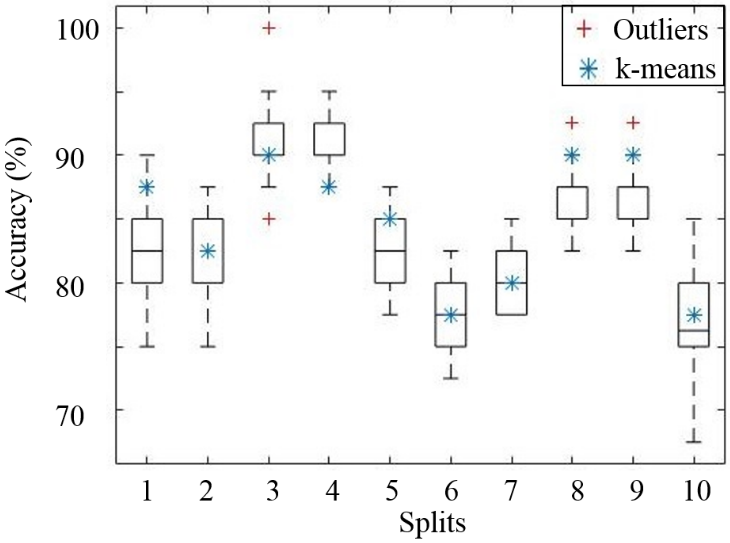

Figure 5: Depicted are boxplots for the rank-1 identification rate obtained in different splits of gallery-probe settings (Experiment 5). In each split, the accuracy is computed 30 times using different random initialization. The red plus-symbol denotes for the outlier results in random initialization, and the blue star-symbol denotes the rank-1 identification rate obtained using k-means initialization.

initialization approaches: (i) k-means clustering and (ii) random sampling. In particular, for each gallery-probe split, the algorithm was tested using 30 different random initialization settings. This experiment was conducted using the ETH-80 dataset, setting the number of prototypes to 10 (as in Experiment 1). A summary of the results is depicted in Fig. 5. The blue star-symbol denotes the rank-1 identification rate obtained using k-means initialization. The red plus-symbol denotes the extreme results treated as outliers. As indicated, in every split, the performance of random initialization can be worse or better than k-means initialization. However, in five out of 10 cases the performance of k-means initialization is better than the average performance of random sampling; and in four out of 10 cases they are comparable. This indicates that k-means clustering offers a better initialization for the proposed SPML. This is expected as the random sampling strategy results in loss of important information.

Experiment 6: The objective of this experiment is to analyze the failure case (i.e., using FPLBP features in TYF with 10 prototypes) reported in Experiment 1. In particular, we look into updates of the objective value, training and testing accuracy on each iteration. The results are summarized in Fig. 6(a). To compare it with a successful case, the corresponding performance obtained using LBP in the YTF dataset is also provided in Fig. 6(b). As illustrated, in both cases the objective value and training accuracy converge very fast. For the FPLBP feature, its testing accuracy keeps decreasing with the updates. For the LBP feature, its testing accuracy increases slowly with the updates, and finally achieves a 5% improvement. One possible reason is that

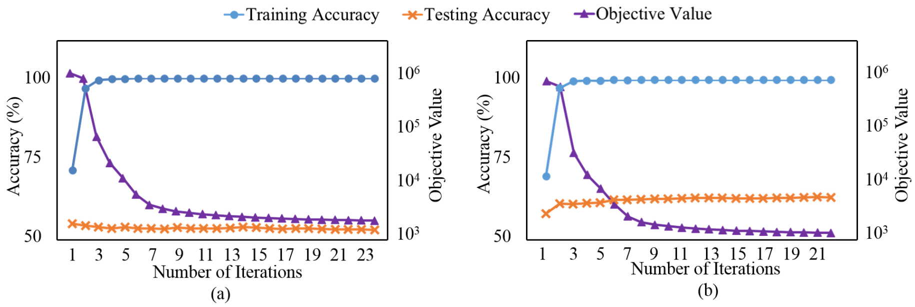

Figure 6: Convergence property of objective value, training and testing accuracy (Experiment 6): (a) results obtained in YTF using FPLBP features; (b) results obtained in YTF using LBP features. In both figures, the left axis indicates accuracy (training and testing); the right axis indicates the objective value. All numbers reported are the average values.

the learning process overfits from the first iteration. To verify this interpretation, we decrease the learning rate of both prototype learning and metric learning (i.e., $\eta_{\boldsymbol{M}}$ and $\eta_{\boldsymbol{\mathcal{Z}}}$) to 0.001, while keeping all other settings fixed. The obtained results are depicted in Fig. 7. As illustrated, the testing accuracy improves in the first five iterations, and starts to decrease from the sixth iteration. This observation verified our interpretation that it was overfitting from the first iteration in the failure case ($\eta_{\boldsymbol{M}}, \eta_{\boldsymbol{\mathcal{Z}}} = 0.01$). This overfitting is caused by the large initial learning rate. Although the overfitting at the first iteration can be avoided by decreasing the initial learning rate, it still suffers from overfitting after the convergence (Fig. 7). The small learning rate will also result in a long convergence time. A practical way to address this problem is to split a validation set from gallery to cross-validate a proper initial learning rate and a stop criterion.

Experiment 7: The objective of this experiment is to analyze the contributions of two learning procedures in SPML: set-based metric learning (SML) and set-based prototype learning (SPL). In SML, the results of k-means initialization are used as fixed prototypes. The Mahalanobis metric is learned by minimizing Eq. 13. The updating rules are kept the same as described in Eq. 9. In SPL, the Mahalanobis $\boldsymbol{M}$ is fixed as an identity matrix. The prototypes are learned by minimizing Eq. 13. The corresponding updating rules are kept the same as described in Eq. 9. In particular, we observe the testing accuracy of SPML, SML, and SPL at each iteration. Corresponding results are summarized in Fig. 8. Results are obtained using LBP features in YTF datasets. All the settings are kept the same as Experiment 1. As illustrated, the SPL converges very fast (only 13 iterations). However, the testing accuracy does not fit well with the prototype learning procedure. The testing accuracy keeps decreasing after the second iteration. The SML is much more stable than the SPL. The testing accuracy increases in the first few iterations and then starts to decrease slowly. The combined SPML, on the other hand, keeps increasing the testing accuracy on each update. As a result, there is no single step that makes SPML work. Both the prototype learning and metric learning process contribute to performance of SPML.

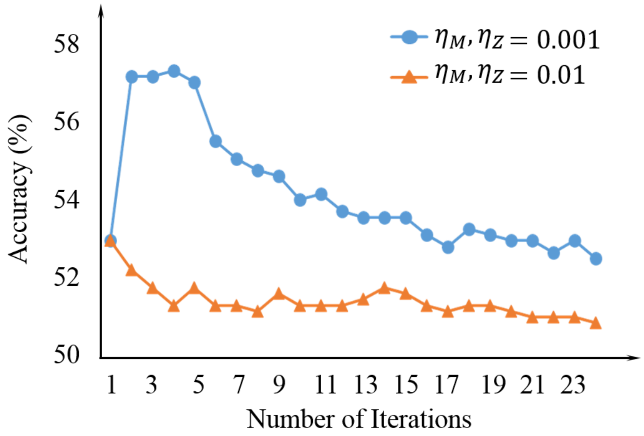

Figure 7: Convergence property of testing accuracy (Experiment 6) obtained using a reduced learning rate.

6. Limitation

In this section, we summarize the limitations of the proposed PML. First, the way we learn the prototypes makes it impossible to extend the algorithm to work with an unseen subject directly. The prototypes can only be learned for the training image sets. As a result, it is not applicable for tasks like verification. Second, it is required by the LMNN-like objective function (Eq. (7)) that the number of image sets per subject in the gallery to be at least $k + 1$, where k is the number of neighbors used. Third, the optimization only takes into consideration the k nearest matches. As a result, the rank-k' identification rate may not be competitive when k' is larger than the number of neighbors used. Finally, it is a complex model that inherits all parameters from LMNN and RNP.

7. 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 statistical test on random initialization indicates that our method works better with k-means initialization. Independent tests on set-based prototype learning and metric learning show that these two processes work together to improve the performance of SPML. We also investigated a failure case and found an overfitting issue. Despite its many advantages, the current form of SPML can be further improved in the following aspects: (i) address the overfitting issues using regularization, (ii) extend it to unseen

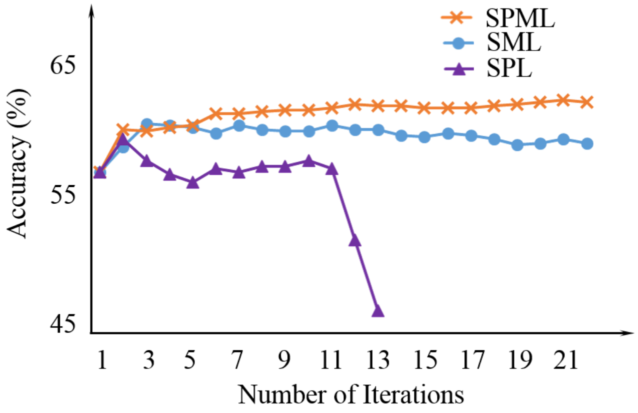

Figure 8: Convergence property of testing accuracy (Experiment 7) obtained using SPML, SML, and SPL. Results are obtained using LBP features in YTF.

subjects, (iii) leverage prototype and metric learning in a different way, and (iv) embed different objective functions into this framework.

8. 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. Yamaguchi, K. Fukui, K. Maeda, Face recognition using temporal image sequence, in: Proc. IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 318–323.

[2] Y. Wu, M. Minoh, M. Mukunoki, Collaboratively regularized nearest points for set based recognition, in: Proc. 24th British Machine Vision Conference, Bristol, UK, 2013, pp. 1–8.

[3] J. Hamm, D. Lee, Grassmann discriminant analysis: A unifying view on subspace-based learning, in: Proc. International Conference on Machine learning, Helsinki, Finland, 2008, pp. 376–383.

[4] M. T. Harandi, C. Sanderson, S. Shirazi, B. C. Lovell, Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, 2011, pp. 2705–2712.

[5] Z. Huang, R. Wang, S. Shan, X. Chen, Projection metric learning on Grassmann manifold with application to video based face recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 140–149.

[6] A. Hadid, M. Pietikainen, From still image to video-based face recognition: An experimental analysis, in: Proc. IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, 2004, pp. 813–818.

[7] W. Fan, D. Yeung, Locally linear models on face appearance manifolds with application to dual-subspace based classification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, 2006, pp. 1384–1390.

[8] R. Wang, X. Chen, Manifold discriminant analysis, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Miami,FL, 2009, pp. 429–436.

[9] G. Shakhnarovich, J. Fisher, T. Darrell, Face recognition from long-term observations, in: Proc. European Confernce on Computer Vision, Copenhagen, Denmark, 2002, pp. 851–865.

[10] S. Chen, C. Sanderson, M. Harandi, B. Lovell, Improved image set classification via joint sparse approximated nearest subspaces, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, 2013, pp. 452–459.

[11] T.-K. Kim, J. Kittler, R. Cipolla, Discriminative learning and recognition of image set classes using canonical correlations, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1005–1018.

[12] M. Harandi, C. Sanderson, C. Shen, B. Lovell, Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 3120–3127.

[13] R. Wang, S. Shan, X. Chen, W. Gao, Manifold-manifold distance with application to face recognition based on image set, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1–8.

[14] W. Wang, R. Wang, Z. Huang, S. Shan, X. Chen, Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 2048–2057.

[15] J. Lu, G. Wang, P. Moulin, Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 329–336.

[16] Z. Huang, R. Wang, S. Shan, X. Chen, Hybrid Euclidean-and-Riemannian metric learning for image set classification, in: Proc. Asian Conference on Computer Vision, Singapore, Singapore, 2014.

[17] R. Wang, H. Guo, L. Davis, Q. Dai, Covariance discriminative learning: A natural and efficient approach to image set classification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 2496–2503.

[18] M. Harandi, M. Salzmann, R. Hartley, From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices, in: Proc. European Confernce on Computer Vision, Zürich, Switzerland, 2014, pp. 17–32.

[19] H. Cevikalp, B. Triggs, Face recognition based on image sets, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 2567–2573.

[20] Y. Hu, A. Mian, R. Owens, Sparse approximated nearest points for image set classification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, 2011.

[21] M. Yang, P. Zhu, L. Van Gool, L. Zhang, Face recognition based on regularized nearest points between image sets, in: Proc. IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, Shanghai, China, 2013, pp. 1–7.

[22] P. Zhu, L. Zhang, W. Zuo, D. Zhang, From point to set: Extend the learning of distance metrics, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 2664 – 2671.

[23] P. Zhu, W. Zuo, L. Zhang, S. Shiu, D. Zhang, Image set-based collaborative representation for face recognition, IEEE Transactions on Information Forensics and Security 9 (7) (2014) 1120 – 1132.

[24] M. Köstinger, P. Wohlhart, P. Roth, H. Bischof, Joint learning of discriminative prototypes and large margin nearest neighbor classifiers, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 3112–3119.

[25] M. Leng, P. Moutafis, I. Kakadiaris, Joint prototype and metric learning for set-to-set matching: Application to biometrics, in: Proc. International Conference on Biometrics: Theory, Applications and Systems, Arlington, VA, 2015, pp. 1–8.

[26] E. Xing, M. Jordan, S. Russell, A. Ng, Distance metric learning, with application to clustering with side-information, in: Proc. 18th Annual Conference on Neural Information Processing Systems, Vancouver, Canada, 2002, pp. 505–512.

[27] K. Weinberger, L. Saul, Distance metric learning for large margin nearest neighbor classification, The Journal of Machine Learning Research 10 (1) (2009) 207–244.

[28] N. Higham, Computing a nearest symmetric positive semidefinite matrix, Linear algebra and its applications 103 (1) (1988) 103–118.

[29] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 1–8, madison, Wisconsin.

[30] M. Kim, S. Kumar, V. Pavlovic, H. Rowley, Face tracking and recognition with visual constraints in real-world videos, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1–8.

[31] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2011, pp. 529–534.

[32] P. Moutafis, M. Leng, I. A. Kakadiaris, An overview and empirical comparison of distance metric learning methods, IEEE Transactions on Cybernetics PP (99) (2016) 1–14.

[33] Q. Cao, Y. Ying, P. Li, Similarity metric learning for face recognition, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 2408–2415.

[34] W. Wang. Covariance discriminant learning [online] (Oct. 2015).

[35] M. Harandi. Dictionary learning and sparse coding on Grassmann manifolds [online] (Oct. 2015).

[36] P. Zhu. Set-to-set distance metric learning [online] (Aug. 2015).