Similarity Learning on an Explicit Polynomial Kernel Feature Map for Person Re-Identification

Dapeng Chen [†], Zejian Yuan [†], Gang Hua[‡], Nanning Zheng[†], Jingdong Wang [§]
[†] Xi'an Jiaotong University [‡]Stevens Institute of Technology [§]Microsoft Research

Abstract

In this paper, we address the person re-identification problem, discovering the correct matches for a probe person image from a set of gallery person images. We follow the learning-to-rank methodology and learn a similarity function to maximize the difference between the similarity scores of matched and unmatched images for a same person. We introduce at least three contributions to person re-identification. First, we present an explicit polynomial kernel feature map, which is capable of characterizing the similarity information of all pairs of patches between two images, called soft-patch-matching, instead of greedily keeping only the best matched patch, and thus more robust. Second, we introduce a mixture of linear similarity functions that is able to discover different soft-patch-matching patterns. Last, we introduce a negative semi-definite regularization over a subset of the weights in the similarity function, which is motivated by the connection between explicit polynomial kernel feature map and the Mahalanobis distance, as well as the sparsity constraint over the parameters to avoid over-fitting. Experimental results over three public benchmarks demonstrate the superiority of our approach. [1]

1. Introduction

Person re-identification refers to a task of associating the person through different camera views located at different physical sites. In the real case that the camera views are significantly disjoint making the temporal transition time between cameras vary greatly, the temporal information is not enough to approach the problem. Thus a lot of efforts [2, 5, 14, 37] have been devoted to investigating the solutions through appearance information.

Existing works tackle this problem from two paths. The first one is to design a visual descriptor to handle inter-camera differences in lighting conditions, changes in object orientation and object pose. The second path is to learn a similarity function to suppress inter-camera variations, which our work belongs to.

In this paper, we learn a similarity function over a pairwise feature formed by concatenating the patch descriptors of a probe image and a gallery image, with the goal maximizing the difference between the similarity score between an image A and an image B about the same person and that between the image A and any image C about a different person. We follow the learning-to-rank methodology and adopt the triplet loss function.

Our key contributions to the person re-identification problem lie in three aspects. First, we explore the second-polynomial kernel but adopt an explicit feature map instead of the kernel value, to formulate a linear similarity function. The benefit is the ability of characterizing the similarities of all pairs of patches between two images, called soft-patch-matching instead of only keeping the best matched patch in one image for each patch in the other image as done in patch matching [38, 37].

Second, we introduce a latent similarity function, a mixture of linear similarity functions, which is capable of mining various soft-patch-matching patterns. Last, we introduce two regularizers: a subset of the weights in the similarity function forms a negative semi-definite matrix, motivated by the connection between the explicit polynomial kernel feature map and the Mahalanobis distance, and a sparsity constraint for the weights, which makes each component in the mixture of similarity functions aligned with a common function to avoid over-fitting.

2. Related Works

The pipeline of a person re-identification system often consists of two main steps: (1) extracting features from images; (2) measuring the similarity between images. Some works emphasized on feature design [34, 9, 2, 24, 5], and some other works focused on similarity function learning [12, 31, 14, 28, 17, 20, 29, 28, 21, 26, 3, 40].

In the feature extraction step, methods that focus on feature design often try to tackle the person re-identification problem by seeking a very stable and distinctive feature representation. For example, Ma *et al.* [24] present the

[1] This work was done when Dapeng Chen was an intern at Microsoft Research.

person image via covariance descriptors that is robust to illumination change and background variation, while Zhao *et al.* [38] learn the distinct salience feature to distinguish the correct matched person from others. Farenzena *et al.* [9] further consider symmetric and asymmetric prior of human body, to integrate different local feature from different body parts. Cheng *et al.* [5] employ pre-learned pictorial structure model to more accurately localize the body parts.

In contrast, methods that focus on similarity learning usually extract the features in a more straightforward way: most of them extract color or textural histograms from predefined image regions in a "block" shape or "strip" shape [36, 14, 17, 29, 20, 41]; some methods further encode the region descriptors to form high level images features [25, 21]. Our method is compatible with both region based features or encoded features.

In the similarity measuring step, feature design based methods usually employ off-the-shelf distance metrics, such as Euclidean distance [9], Bhattacharyya distance [5], and covariance distance [24, 1], etc. Meanwhile, how to learn a proper similarity measurement is studied in different perspectives. Gray *et al.*[12] employ boosting to a select a subset of optimal features for matching. Prosser *et al.*[31] and Zheng *et al.*[41] stress the importance of loss function and describe the triplet relation between samples. They don't compare the direct similarity score between correct matched and incorrect matched pairs, but are only interested in the rank of these scores that reflects how likely they match to a given query image.

Recently, Mahalanobis distance learning has been applied for re-identification problem [28, 17, 14, 6], where the distance metrics are optimized in either a discriminative fashion [28, 6] or a generative fashion [17]. As Mahalanobis distance can implicitly model the transition in feature space between two camera views, these methods achieve better performance than the similarity functions directly learnt in the original feature space. Li *et al.* [21] further extend the metric learning. They proposed the Locally-Adaptive Decision Function (LADF) to jointly models a distance metric and a locally adaptive thresholding rule.

In this paper, we focus on the second step and develop a new similarity function. The effectiveness of our similarity function stems from the feature representation, the explicit polynomial kernel feature map of concatenated descriptors of image pairs. The purpose of utilization explicit feature map is distinguished from existing explicit kernel work [33, 27]. They derive explicit feature maps to speed up nonlinear kernel machines, while we utilize the explicit polynomial kernel feature map to characterize the image pairs.

With obtained feature, our method can be compared with methods based on patch matching [37, 38]. For each patch, these methods greedily search the corresponding patch in adjacent space and only keep the maximum matching score as the similarity. Meanwhile, our features for image pair maintain the matches of all the possible patch pairs, and the matching criterion is to be learnt from data.

Our method is also related to the methods that learn multiple similarity functions. For example, Li *et al.* [20] learn a mixture of experts, where samples were softly distributed into different experts (similarity function) via a gating function. Ma *et al.* [26] divide the data according to the additional camera position information, and utilize multiple task learning to learn specific distance metric for each camera pair. We learn multiple similarity functions in a latent fashion to discover different matching patterns. The strategy is inspired by latent SVM, which is originally developed for object detection, to learn a mixture of object templates. We modify it for similarity measuring, in order to increase the model's discriminative ability.

3. Formulation

In this section, we first introduce the similarity function $f(\mathbf{x}_1, \mathbf{x}_2)$ for image descriptors $\mathbf{x}_1$ and $\mathbf{x}_2$, then discuss the necessary regularization strategies associated with the similarity function, and finally formulate the objective function for person re-identification. The flowchart of our method is illustrated in Figure 1.

3.1. Similarity Function

We formulate our similarity function by performing an explicit kernel feature map on the concatenated vector $\mathbf{z} = [\mathbf{x}_1^\top \ \mathbf{x}_2^\top]^\top$. The feature map is written as $\phi(\mathbf{z}) = \phi(\mathbf{x}_1, \mathbf{x}_2)$. It is known that there is an explicit feature map for second-order polynomial kernel $k(\mathbf{z}_1, \mathbf{z}_2) = (\mathbf{z}_1^\top \mathbf{z}_2)^2$, that is $\phi(\mathbf{z}) = \mathrm{vec}(\mathbf{z}\mathbf{z}^\top) = [\mathrm{vec}(\mathbf{x}_1\mathbf{x}_1^\top)^\top \ \mathrm{vec}(\mathbf{x}_2\mathbf{x}_1^\top)^\top \ \mathrm{vec}(\mathbf{x}_1\mathbf{x}_2^\top)^\top \ \mathrm{vec}(\mathbf{x}_2\mathbf{x}_2^\top)^\top]^\top$. Here $\mathrm{vec}(\mathbf{A})$ is an operation that vectorizes the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ to be a vector $\mathbf{a} \in \mathbb{R}^{mn \times 1}$. To make the function be symmetric, *i.e.* $f(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_2, \mathbf{x}_1)$, which is natural for the similarity function, we redefine

$$\phi(\mathbf{x}_1, \mathbf{x}_2) = [\mathrm{vec}(\mathbf{x}_1\mathbf{x}_1^\top + \mathbf{x}_2\mathbf{x}_2^\top)^\top \ \mathrm{vec}(\mathbf{x}_2\mathbf{x}_1^\top + \mathbf{x}_1\mathbf{x}_2^\top)^\top]^\top. \quad (1)$$

This feature map takes into account the relations between the feature values from the same positions and different positions: $x_{1d}x_{2d}$ and $x_{1d}x_{2d'}$, where x_{1d} is the dth component of $\mathbf{x}_1$, x_{2d} and $x_{2d'}$ are similarly defined. In the situation when the feature $\mathbf{x}$ is a patch-wise descriptor of an image (each entry or subvector corresponds to a block of the image), $\mathrm{vec}(\mathbf{x}_1\mathbf{x}_2^\top)$ (and $\mathrm{vec}(\mathbf{x}_2\mathbf{x}_1^\top)$) can be viewed as a concatenation of cross-patch similarities of two images, where the cross-patch similarity is a vector formed by vectorizing the outer-product of the patch features.

In other words, it matches each patch in one image with all the patches in the other image and all the matching scores are attained as the descriptor, which we call soft-patch-matching, instead of only keeping the best-matched

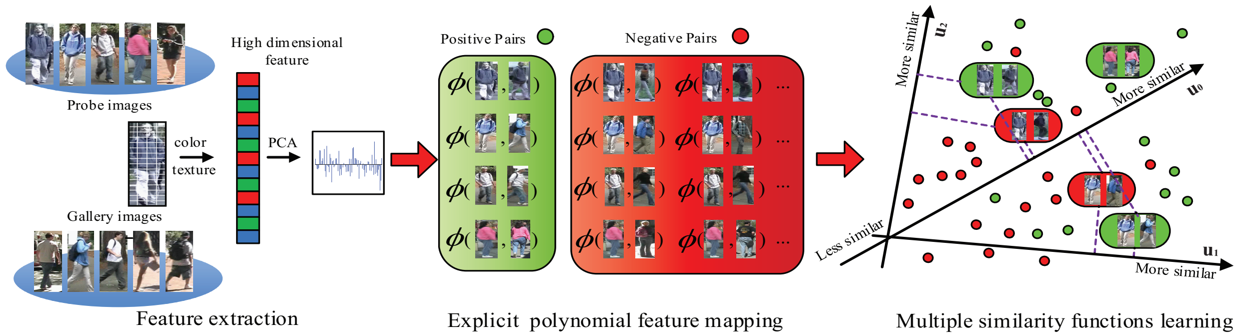

Figure 1: **Flow chart of the proposed method.** We extract the features to represent the images, utilize the explicit polynomial-kernel feature map to represent image pairs, and train a mixture of similarity functions to discover multiple matching patterns.

score. This still holds even when the descriptor $\mathbf{x}$ is transformed through linear dimension reduction, because we have $\mathrm{vec}((\mathbf{P}^\top\mathbf{x}_1)(\mathbf{P}^\top\mathbf{x}_2)^\top) = \mathrm{vec}(\mathbf{P}^\top(\mathbf{x}_1\mathbf{x}_2^\top)\mathbf{P})$, which is equivalent to first performing $\mathbf{x}_1\mathbf{x}_2^\top$ then left multiplying $\mathbf{P}^\top$ and right multiplying $\mathbf{P}$.

The similarity function, $f(\mathbf{x}_1, \mathbf{x}_2)$, is usually linear with respect to the mapped feature $\phi(\mathbf{x}_1, \mathbf{x}_2)$. To handle different soft-patch-matching patterns, we make a nonlinear extension using a latent formulation,

$$f(\mathbf{x}_1, \mathbf{x}_2) = \max_{h=1,\dots H} f_h(\mathbf{x}_1, \mathbf{x}_2), \qquad (2)$$

where $f_h(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{w}_h^\top \phi(\mathbf{x}_1, \mathbf{x}_2)$. Intuitively, the latent formulation aims to discover H representative patterns $\{\mathbf{w}_h\}_{h=1}^H$, and uses the most similar pattern to evaluate the similarity for a pair $(\mathbf{x}_1, \mathbf{x}_2)$ in terms of the inner product.

3.2. Regularization

We propose two regularization over $\mathbf{w}_h$ for each latent linear function. The first regularization is motivated by the connection between explicit polynomial kernel feature map and the Mahalanobis distance as $(\mathbf{x}_1 - \mathbf{x}_2)^\top \mathbf{M}(\mathbf{x}_1 - \mathbf{x}_2) = \mathrm{vec}(\mathbf{M})^\top \mathrm{vec}(\mathbf{x}_1\mathbf{x}_1^\top + \mathbf{x}_2\mathbf{x}_2^\top - \mathbf{x}_1\mathbf{x}_2^\top - \mathbf{x}_2\mathbf{x}_1^\top)$. We rearrange $\phi(\mathbf{x}_1, \mathbf{x}_2) = [\phi^1(\mathbf{x}_1, \mathbf{x}_2),\ \phi^2(\mathbf{x}_1, \mathbf{x}_2)]$, where $\phi^1(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{vec}(\mathbf{x}_1\mathbf{x}_1^\top + \mathbf{x}_2\mathbf{x}_2^\top - \mathbf{x}_1\mathbf{x}_2^\top - \mathbf{x}_2\mathbf{x}_1^\top)$ and $\phi^2(\mathbf{x}_1, \mathbf{x}_2) = \mathrm{vec}(\mathbf{x}_1\mathbf{x}_2^\top + \mathbf{x}_2\mathbf{x}_1^\top)$. Accordingly, $\mathbf{w}_h$ is written as $[\mathbf{w}_h^1,\ \mathbf{w}_h^2]$, and the linear function $f_h(\mathbf{x}_1, \mathbf{x}_2)$ is :

$$f_h(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{w}_h^1)^\top \phi^1(\mathbf{x}_1, \mathbf{x}_2) + (\mathbf{w}_h^2)^\top \phi^2(\mathbf{x}_1, \mathbf{x}_2). \qquad (3)$$

The first half component $(\mathbf{w}_h^1)^\top \phi^1(\mathbf{x}_1, \mathbf{x}_2)$ is related to Mahalanobis distance, as it is equivalent to $(\mathbf{x}_1 - \mathbf{x}_2)^\top \mathrm{mat}(\mathbf{w}_h^1)(\mathbf{x}_1 - \mathbf{x}_2)$. Here, $\mathrm{mat}(\mathbf{a})$ is the inverse operation of $\mathrm{vec}(\mathbf{A})$ that recovers the matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ from $\mathbf{a} \in \mathbb{R}^{mn\times 1}$. We impose the negative semi-definite regularization over $\mathbf{w}_h^1$: $\mathrm{mat}(\mathbf{w}_h^1) \preceq 0$, as we utilize the negative Mahalanobis distance to measure the similarity.

The second regularization is motivated by the assumption that different matching patterns share a common component, which describes a general matching pattern for all the image pairs. We decompose $\mathbf{w}_h = \mathbf{u}_h + \mathbf{u}_0$, and align the weights of the H similarity functions to a common weight vector $\mathbf{u}_0$. The alignment is imposed by a sparsity regularization: $\sum_{h=1}^H \|\mathbf{u}_h\|_1$. In addition, we also impose the sparsity regularization $\|\mathbf{u}_0\|_1$, which is widely used for feature selection.

3.3. Objective function

The training data for person re-identification can be transformed as follows. Given a set of probe images $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, each image $\mathbf{x}_n$ is associated with two sets of gallery images: a positive set $\mathcal{X}_n^+$ composed of the images about the same person with $\mathbf{x}_n$ and a negative set $\mathcal{X}_n^-$ composed of the images about different persons. As the re-identification problem is usually formulated as a ranking problem, we utilize the triplet loss function:

$$\ell_{\mathrm{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = [f(\mathbf{x}_i, \mathbf{x}_k) - f(\mathbf{x}_i, \mathbf{x}_j) + 1]_+, \qquad (4)$$

where $\mathbf{x}_j \in \mathcal{X}_i^+$ and $\mathbf{x}_k \in \mathcal{X}_i^-$. Intuitively, this means that given a probe image $\mathbf{x}_i$, a gallery image belonging to a same person $\mathbf{x}_j \in \mathcal{X}_i^+$ should be scored higher than a image belonging to a different person $\mathbf{x}_k \in \mathcal{X}_i^-$ at least by a margin 1. The whole loss is:

$$\mathcal{L}(f) = \sum_{i=1}^N \sum_{\mathbf{x}_j \in \mathcal{X}_i^+, \mathbf{x}_k \in \mathcal{X}_i^-} \ell_{\mathrm{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k). \qquad (5)$$

With the regularization, the objective function for person re-identification is given as:

$$\min_{\mathbf{u}_0,\dots,\mathbf{u}_H} \quad \mathcal{L}(\mathbf{u}_0, \dots, \mathbf{u}_H) + \lambda \textstyle\sum_{h=0}^H \|\mathbf{u}_h\|_1 \qquad (6)$$

$$\text{s.t.} \quad \mathrm{M}(\mathbf{u}_h) \preceq 0, h = 0, 1, \dots, H. \qquad (7)$$

where $\mathrm{M}(\mathbf{u}_h) = \mathrm{mat}(\mathbf{u}_h^1)$. $\mathbf{u}_h^1$ is the first half part of $\mathbf{u}_h$. As $\mathbf{w}_h = [\mathbf{w}_h^1, \mathbf{w}_h^2] = [\mathbf{u}_h^1 + \mathbf{u}_0^1, \mathbf{u}_h^2 + \mathbf{u}_0^2]$, constraint 7 also derives $\mathrm{mat}(\mathbf{w}_h^1) \preceq 0$.

4. Optimization

We concatenate the $(H+1)$ weight vectors $\{\mathbf{u}\}_{h=0}^H$ into a single vector $\mathbf{v} = [\mathbf{u}_0^\top\ \mathbf{u}_1^\top \cdots \mathbf{u}_H^\top]^\top$. Accordingly, we reformulate the similarity function as,

$$f(\mathbf{x}_1, \mathbf{x}_2; \mathbf{v}) = \max_{h=1,2,\cdots,H} f(\mathbf{x}_1, \mathbf{x}_2, h; \mathbf{v}) \qquad (8)$$

where $f(\mathbf{x}_1, \mathbf{x}_2, h; \mathbf{v})$ is defined as:

$$f(\mathbf{x}_1, \mathbf{x}_2, h; \mathbf{v}) = \mathbf{v}^\top \psi(\mathbf{x}_1, \mathbf{x}_2, h). \tag{9}$$

$\psi(\mathbf{x}_1, \mathbf{x}_2, h)$ is a vector with the same length to $\mathbf{v}$ and its entries are zeros except that the two subvectors corresponding to $\mathbf{u}_0$ and $\mathbf{u}_h$ are both equal to $\phi(\mathbf{x}_1, \mathbf{x}_2)$:

$$\begin{aligned}&\psi(\mathbf{x}_1, \mathbf{x}_2, h)\\ =&[\psi^0(\mathbf{x}_1, \mathbf{x}_2, h)^\top \psi^1(\mathbf{x}_1, \mathbf{x}_2, h)^\top \cdots \psi^H(\mathbf{x}_1, \mathbf{x}_2, h)^\top]^\top,\end{aligned} \tag{10}$$

where $\psi^0(\mathbf{x}_1, \mathbf{x}_2, h) = \psi^h(\mathbf{x}_1, \mathbf{x}_2, h) = \phi(\mathbf{x}_1, \mathbf{x}_2)$ and $\psi^k(\mathbf{x}_1, \mathbf{x}_2, h) = \mathbf{0}, \forall k \neq 0, h$.

Considering the formulation 6 and 7, we have the following observations: the negative semi-definite regularization in the constraint 7 defines a closed convex set; the sparse term ($\|\mathbf{v}\|_1$) in the objective 6 is convex; but the loss term $\mathcal{L}(\mathbf{v})$ in 6 is not guaranteed to be convex. The uncertainty is from the content $f(\mathbf{x}_i, \mathbf{x}_k) - f(\mathbf{x}_i, \mathbf{x}_j)$ within $\ell_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$, where $f(\mathbf{x}_i, \mathbf{x}_j)$ is nonlinear.

For optimization, we first introduce an auxiliary function for $\ell_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ (Equation 4), that is:

$$\ell'_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, h_{ij}) = [f(\mathbf{x}_i, \mathbf{x}_k) - f(\mathbf{x}_i, \mathbf{x}_j, h_{ij}) + 1]_+,$$

where h_{ij} is a specified latent value for positive pair $\mathbf{x}_i, \mathbf{x}_j$. $\ell'_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, h_{ij})$ is convex with respect to $\mathbf{v}$. At the same time, $\ell_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \leqslant \ell'_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, h_{ij})$ is inferred from Equation 8. Based on $\ell'_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, h_{ij})$, we further introduce an auxiliary function for $\mathcal{L}(\mathbf{v})$, that is;

$$\mathcal{L}'(\mathbf{v}) = \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in \mathcal{X}_i^+, \mathbf{x}_k \in \mathcal{X}_i^-} \ell'_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, h_{ij}^{(t+1)}). \tag{11}$$

Accordingly, $\mathcal{L}'(\mathbf{v})$ is a convex function and bounds $\mathcal{L}(\mathbf{v})$. This justifies optimizing objective function 6 by employing $\mathcal{L}'(\mathbf{v})$. In practice, an EM-like iterative optimization algorithm is applied to alternatively optimizing $\{h_{ij}\}$ for positive pairs $\{(\mathbf{x}_i, \mathbf{x}_j)\}_{j \in \mathcal{X}_i^+}$ and $\mathbf{v}$ from the convex optimization with $\mathcal{L}'(\mathbf{v})$. Both steps decrease the objective function and can achieve the convergence. The whole algorithm is summarized in Algorithm 1.

4.1. Latent positive variable estimation

Let $\mathbf{v}^t$ be the estimated weight vector at the iteration t. The $(t+1)$ iteration first estimates the hidden variables $\{h_{ij}\}$ for positive pairs $\{(\mathbf{x}_i, \mathbf{x}_j)\}_{j \in \mathcal{X}_i^+}$:

$$h_{ij}^{(t+1)} = \arg \max_{h=1,\cdots,H} f(\mathbf{x}_i, \mathbf{x}_j, h; \mathbf{v}^t). \tag{12}$$

4.2. Weight vector update

The loss function $\mathcal{L}'(\mathbf{v})$ is convex with respect to $\mathbf{v}$. Thus, the optimization problem is convex, and the objective function can be written as

$$\min_{\mathbf{v}} g_1(\mathbf{v}) + g_2(\mathbf{v}) + g_3(\mathbf{v}). \tag{13}$$

The three terms are written as below. The loss term is $g_1(\mathbf{v}) = \mathcal{L}'(\mathbf{v})$. The sparsity term is $g_2(\mathbf{v}) = \lambda \|\mathbf{v}\|_1$. The semi-definite constraint term is written as $g_3(\mathbf{v}) = \infty\delta[\mathbf{v} \notin \mathcal{C}]$, where $\mathcal{C}$ is a closed convex set defined from the constraint 7: $\mathcal{C} = \{\mathbf{v} | \, \mathrm{M}(\mathbf{u}_h) \preceq 0, h = 0, 1, \ldots, H\}$.

We propose to adopt the alternating direction method of multipliers (ADMM) and optimize an equivalent problem

$$\begin{aligned}\min_{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3} \quad & g_1(\mathbf{v}_1) + g_2(\mathbf{v}_2) + g_3(\mathbf{v}_3)\\ \text{s.t.} \quad & \mathbf{v}_1 = \mathbf{v}_2 = \mathbf{v}_3.\end{aligned} \tag{14}$$

Through introducing Lagrange multipliers $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, we obtain the augmented Lagrangian,

$$\begin{aligned}\Phi(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \; & g_1(\mathbf{v}_1) + g_2(\mathbf{v}_2) + g_3(\mathbf{v}_3)\\ & + \rho\boldsymbol{\mu}_1^\top(\mathbf{v}_1 - \mathbf{v}_3) + \frac{\rho}{2}\|\mathbf{v}_1 - \mathbf{v}_3\|_2^2\\ & + \rho\boldsymbol{\mu}_2^\top(\mathbf{v}_2 - \mathbf{v}_3) + \frac{\rho}{2}\|\mathbf{v}_2 - \mathbf{v}_3\|_2^2\end{aligned} \tag{15}$$

where $\rho > 0$ is a scaling parameter. The ADMM algorithm in the scaled form consists of the iterations,

$$\begin{aligned}\mathbf{v}_1^{k+1} &= \arg\min_{\mathbf{v}_1} g_1(\mathbf{v}_1) + \frac{\rho}{2}\|\mathbf{v}_1 - (\mathbf{v}_3^k - \boldsymbol{\mu}_1^k)\|_2^2 \\ \mathbf{v}_2^{k+1} &= \arg\min_{\mathbf{v}_2} g_2(\mathbf{v}_2) + \frac{\rho}{2}\|\mathbf{v}_2 - (\mathbf{v}_3^k - \boldsymbol{\mu}_2^k)\|_2^2\\ \mathbf{v}_3^{k+1} &= \arg\min_{\mathbf{v}_3} g_3(\mathbf{v}_3) + \frac{\rho}{2}\|\mathbf{v}_3 - \frac{1}{2}(\mathbf{v}_1^{k+1} + \mathbf{v}_2^{k+1} + \boldsymbol{\mu}_1^k + \boldsymbol{\mu}_2^k)\|_2^2\\ \boldsymbol{\mu}_1^{k+1} &= \boldsymbol{\mu}_1^k + \mathbf{v}_1^{k+1} - \mathbf{v}_3^{k+1}\\ \boldsymbol{\mu}_2^{k+1} &= \boldsymbol{\mu}_2^k + \mathbf{v}_2^{k+1} - \mathbf{v}_3^{k+1}\end{aligned} \tag{16}$$

The updates of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ are represented below.

Update $\mathbf{v}_1$. We use the subgradient method to optimize the problem 16. Let $h_{ik} = \max_{h=1,\cdots,H} f(\mathbf{x}_i, \mathbf{x}_k, h; \mathbf{v}_1)$ for all the negative pairs $\{(\mathbf{x}_i, \mathbf{x}_k)\}_{k \in \mathcal{X}_i^-}$. Then the triplet loss $\ell'_{\text{triplet}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, h_{ij}^{(t+1)}) = [\mathbf{v}_1\psi(\mathbf{x}_i, \mathbf{x}_k, h_{ik}) - \mathbf{v}_1\psi(\mathbf{x}_i, \mathbf{x}_j, h_{ij}^{(t+1)}) + 1]_+$. A subgradient of the problem 16 consists of two parts, written as $\boldsymbol{\nabla} = \boldsymbol{\nabla}_1 + \boldsymbol{\nabla}_2$, where $\boldsymbol{\nabla}_1 = \sum_{i=1}^{N} \sum_{\mathbf{x}_j \in \mathcal{X}_i^+, \mathbf{x}_k \in \mathcal{X}_i^-} (\psi(\mathbf{x}_i, \mathbf{x}_k, h_{ik}) - \psi(\mathbf{x}_i, \mathbf{x}_j, h_{ij}^{(t+1)}) \delta[\mathbf{v}_1\psi(\mathbf{x}_i, \mathbf{x}_k, h_{ik}) - \mathbf{v}_1\psi(\mathbf{x}_i, \mathbf{x}_j, h_{ij}^{(t+1)}) + 1 > 0]$ and $\boldsymbol{\nabla}_2 = \rho(\mathbf{v}_1 - \mathbf{v}_3^k + \boldsymbol{\mu}_1^k)$. The resulting algorithm is an iterative algorithm, alternatively updating h_{ik} for the negative pairs and updating $\mathbf{v}_1$ by $\mathbf{v}_1 \leftarrow \mathbf{v}_1 - \alpha\boldsymbol{\nabla}$ with α being the learning rate. The result at the convergence of the iterative algorithm is output as $\mathbf{v}_1^{k+1}$.

Update $\mathbf{v}_2$. We adopt soft-thresholding to update $\mathbf{v}_2$. The following presents the update of the jth element $v_{2,j}$ of $\mathbf{v}_2$

$$v_{2,j}^{k+1} = \begin{cases} v_{3,j}^k - \mu_{2,j}^k - \frac{\lambda}{\rho}, & \text{if } v_{3,j}^k - \mu_{2,j}^k \geqslant \frac{\lambda}{\rho} \\ 0, & \text{if } |v_{3,j}^k - \mu_{2,j}^k| < \frac{\lambda}{\rho} \\ v_{3,j}^k - \mu_{2,j}^k + \frac{\lambda}{\rho}, & \text{if } v_{3,j}^k - \mu_{2,j}^k \leqslant -\frac{\lambda}{\rho} \end{cases} \tag{17}$$

where $v_{3,j}, \mu_{2,j}$ are the jth element of $\mathbf{v}_3$ and $\boldsymbol{\mu}_2$ respectively.

Algorithm 1 The main algorithm.

```
Input: Dateset $\mathcal{D} = \{\mathbf{x}_i, \mathcal{X}_i^+, \mathcal{X}_i^-\}_{i=1}^n$,
        Initialize $h_{ij}^1$ for each positive pairs,
Output: weight $\mathbf{v}$
for $t = 1, 2, ..., T$ do
    Randomly initialize $\mathbf{v}_3^0$
    for $k = 0, ..., K-1$ (until convergence) do
        Update $\mathbf{v}_1^{k+1}$ by solving Equation 16
        Update $\mathbf{v}_2^{k+1}$ by soft thresholding in Equation 17
        Update $\mathbf{v}_3^{k+1}$ by projection in Equation 18
    end for
    $\mathbf{v}^t \leftarrow \mathbf{v}_3^K$
    Infer $h_{ij}^{(t+1)}$ for positive pairs based on Equation 12
end for
$\mathbf{v} \leftarrow \mathbf{v}^T$
```

Update $\mathbf{v}_3$. $\mathbf{v}_3$ is updated through a simple projection:

$$\mathbf{v}_3^{k+1} = \Pi_{\mathcal{C}}[\frac{1}{2}(\mathbf{v}_1^{k+1} + \mathbf{v}_2^{k+1} + \boldsymbol{\mu}_1^k + \boldsymbol{\mu}_2^k)]. \qquad (18)$$

The projection is obtained by first computing $\bar{\mathbf{v}}_3^{k+1} = \frac{1}{2}(\mathbf{v}_1^{k+1} + \mathbf{v}_2^{k+1} + \boldsymbol{\mu}_1^k + \boldsymbol{\mu}_2^k)$, then updating a part of entries in $\bar{\mathbf{v}}_3^{k+1}$ that corresponding to the first subvector $\mathbf{u}_h^1$ of each weight vector $\mathbf{u}_h$ (recall that $\mathbf{v} = [\mathbf{u}_0^\top\ \mathbf{u}_1^\top \cdots \mathbf{u}_H^\top]^\top$) through cropping the positive eigenvalues of $\mathrm{M}(\mathbf{u}_h^1)$, and finally obtaining $\mathbf{v}_3^{k+1}$.

5. Discussions

Similarity function. The relative distance comparison approach (PRDC) [41], which learns the Mahalanobis distance to align the relative similarities, and the locally-adaptive decision function (LADF) approach [21], which learns a second-order symmetric function, are close to the proposed linear similarity function and can be cast into the explicit polynomial kernel feature map. The joint Bayesian approach to face recognition [4], similar to [21], also uses a second-order symmetric function, but is learnt in a generative manner.

Our approach is different from them in several aspects, including the triplet loss function, the mixture formulation for nonlinear extension, as well as the sparsity regularization. Table 1 compares the performance of the three methods and ours on the VIPER dataset with 316 gallery images. In particular, Joint-Bayesian is implemented by ourselves following details in [4] and utilize the same feature descriptors with LADF and our method.

Latent formulation and regularization. The latent formulation has also been investigated in other applications, such as object detection [10]. Our approach is the first to study it in the person re-identification problem. Differently, our approach aligns each latent similarity function to a common function, and adopts the sparsity regularization to avoid over-fitting. It should be noted that the latent formulation has also been studied in machine learning, such as multi-class latent locally linear support vector machines [11]. The regularization scheme resembles clustered SVM [13] that uses a similar alignment scheme but a L_2 regularizer, regularized multi-task learning [8], and so on.

Table 1: Comparing our method and related similarity functions on the VIPER dataset, the size of the gallery set is 316.

methods	r=1	r=5	r=10	r= 20	r=50
PRDC[41]	15.7	38.4	53.9	70.1	- -
Joint-Bayesian[4]	27.1	60.4	74.8	87.2	97.3
LADF[21]	30.0	64.7	79.0	91.3	97.2
Ours	36.8	70.4	83.7	91.7	97.8

6. Experiments

In this section, we evaluate the proposed similarity learning approach for the person re-identification task on three widely-used datasets: VIPER [12], GRID [23] and CAVIAR4REID [2], as well as for the face verification task on the LFW dataset [15] using the binary loss function.

6.1. Setup

Visual descriptors and preprocessing. We follow [21] to use the high level feature based on patch color descriptors for the VIPER dataset, and the block feature similar to [29] for the GRID and CAVIR4REID datasets. We observe that the high-level descriptor is better on the viper dataset (with about a 2% improvement in terms of the rank-1 matching rate), and the block feature is more suitable for the GRID and CAVIR4REID datasets. We employ principal component analysis (PCA) to reduce the dimension. To limit the impact of co-occurrence [16], we do a whitening process by dividing each dimension by the inverse of the square root of the corresponding eigenvalue. The resulting feature vectors are normalized so that its L_2 norm is 1.

Parameter setting. The coefficient λ of the $L1$ sparsity regularization in Equation 6 and the penalty variable ρ in Equation 15 are determined via cross validation. To achieve better performance, we empirically set the PCA-reduced dimension and the number of latent functions for the VIPER GRID and CAVIR4REID datasets to be $\{100, 45, 80\}$ and $\{6, 3, 2\}$, respectively. The number of outer iterations is 5, the maximum iteration number for the ADMM algorithm is 15, and the iteration number for updating $\mathbf{v}_1$ is set as 10. We initialize the latent variable h_{ij} for positive pairs $(\mathbf{x}_i, \mathbf{x}_j)$ by clustering the positive samples using the spherical k-means algorithm [7] which utilizes dot product to measure the inter-sample similarities.

Evaluation scheme. We adopt a single-shot evaluation. The persons in each dataset are separated into the training set and testing set so that each person appears only in the

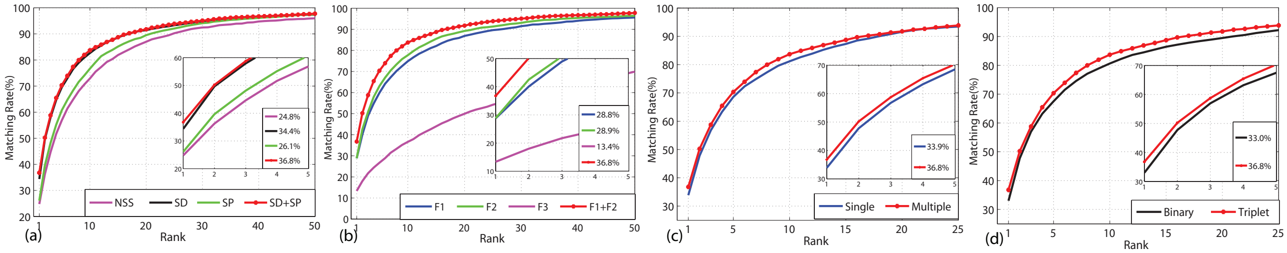

Figure 2: **Empirical analysis:** The average CMC curves for analyzing the effect of (a) regularization. (b) explicit polynomial-kernel feature map. (c) latent formulation. (d) the loss function. All the experiments are run 10 times with the same partitions. The size of the gallery set is 316.

Methods	Regularization	Feature	H	Loss function	r=1	r=5	r=10	r=20	r=25	r= 50
NSS	None	$\phi(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	24.8	56.8	72.9	87.0	90.3	96.0
SP	Sparse(SP)	$\phi(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	26.1	60.7	76.7	89.5	92.1	97.4
SD	Semi-definite(SD)	$\phi(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	**34.4**	**69.5**	**82.7**	91.4	93.3	97.4
F1	SP+SD	$\phi^1(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	28.8	50.8	74.8	86.9	89.2	95.6
F2	SP+SD	$\phi^2(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	28.9	63.0	77.7	89.2	91.3	96.4
F3	SP+SD	$\phi^3(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	13.4	26.7	36.6	49.9	54.2	69.9
Single	SP+SD	$\phi(\mathbf{x}_1, \mathbf{x}_2)$	0	triplet	33.9	68.6	81.2	**91.7**	**93.5**	**97.6**
Binary	SP+SD	$\phi(\mathbf{x}_1, \mathbf{x}_2)$	6	binary	33.0	67.5	80.6	89.5	92.2	96.6
Ours	SP+SD	$\phi(\mathbf{x}_1, \mathbf{x}_2)$	6	triplet	**36.8**	**70.4**	**83.7**	**91.7**	**93.9**	**97.8**

Table 2: **Performance of different configuration:** The top-n matching rate of the methods with different configurations about regularization strategy, explicit polynomial kernel feature map, the number of similarity funcion H and the loss function. All the experiments are run 10 times with the same partitions. The size of the gallery set is 316.

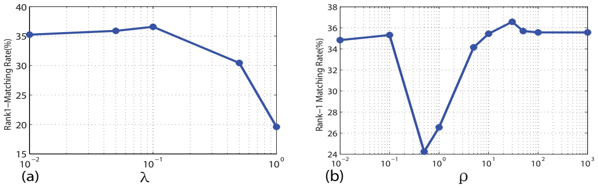

Figure 3: The rank-1 matching rate with respect to (a) parameter λ when ρ is fixed to be 30. (b) parameter ρ when λ is fixed to be 0.1.

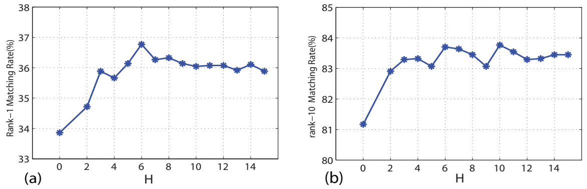

Figure 4: The influence of the number of multiple-function H on (a) rank-1 matching rate. (b) rank-10 matching rate.

training set or the testing set. We partition the testing set into two sets: the probe set and the gallery set. The gallery set contains one image, and the probe set contains one image (VIPER, GRID) or multiple images (CAVIR4REID). The results are evaluated by cumulative matching characteristic (CMC) curves [12], an estimate of the expectation of finding the correct match in the top n matches. The final results are averaged over ten random runs.

6.2. Empirical analysis

We empirically analyze how various components in our approach affect the performance. We use the results obtained from the VIPER dataset to show the analysis result.

The effect of regularization. There are two regularization schemes in the proposed approach, sparsity and negative semi-definite. We report four kinds of results with respect to regularization: without regularization (NSS), only with sparsity regularization (SP), with negative semi-definite regularization and the $L2$ regularization replacing the sparsity regularization (SD), and with both regularization(SP+SD). The CMC curves and quantitative results are reported in Figure 2(a) and Table 2. It can be observed that the semi-definite projection takes a major contribution that improves the rank-1 matching rate from 24.8% to 34.4%, while sparsity is also indispensable that further improves the result of L_2 regularization from 34.4% to 36.8%. Both sparsity and semi-definite regularization consistently improve the NSS in all range. The gain from regularization has also been emphasized in [36], where they impose the L_2 norm of the coefficients to improve PCCA[28], while we show that sparsity regularization is more effective for our method.

The effect of explicit polynomial-kernel feature map. To show the effectiveness of the proposed polynomial-kernel feature map $\phi(\mathbf{x}_1, \mathbf{x}_2)$, we construct three variants of our methods F1, F2 and F3, which are obtained by replacing $\phi(\mathbf{x}_1, \mathbf{x}_2)$ with $\phi^1(\mathbf{x}_1, \mathbf{x}_2)$, $\phi^2(\mathbf{x}_1, \mathbf{x}_2)$ and $\phi^3(\mathbf{x}_1, \mathbf{x}_2)$. Among them, $\phi^1(\mathbf{x}_1, \mathbf{x}_2)$ and $\phi^2(\mathbf{x}_1, \mathbf{x}_2)$ are two parts of proposed polynomial feature after regularization, $\phi^3(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2|$ measures the first order absolute difference. Denoting our method as F1+F2, we compare F1+F2 with F1, F2 and F3 in the Figure 2(b). From the Figure, we can see F1, F2 and F1+F2 significantly outperform F3, which indicates the second order correlation used here can represent much more useful information for similarity measuring. Furthermore, we find that F1 and F2 perform almost the

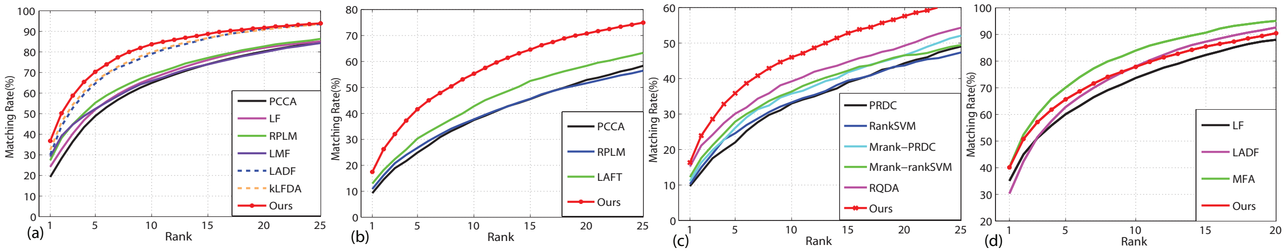

Figure 5: **Comparison with other methods:** CMC curves of our method and other competing methods on (a) the VIPER dataset with 316 gallery images. (b) the VIPER dataset with 532 gallery images. (c) the GIRD dataset with 900 gallery images. (d) the CAVIER4REID dataset with 36 gallery images.

Table 3: The rank-n matching rates(%) for comparison with other methods on the VIPER dataset. The size of the gallery set is 316.

Methods	r = 1	r = 5	r = 10	r = 20	r = 50
PCCA [28]	19.3	48.9	64.9	80.3	- -
LF [29]	24.2	52.0	67.1	82.0	94.1
RPLM [14]	27.3	55.3	69.0	82.7	95.1
LMF [39]	29.1	52.3	66.0	79.9	- -
LADF [21]	30.0	64.7	79.0	**91.3**	**97.2**
kLFDA [36]	**32.3**	**65.8**	**79.7**	90.9	- -
Ours	**36.8**	**70.4**	**83.7**	**91.7**	**97.8**

Table 4: The rank-n matching rates(%) for comparison with other methods on the VIPER dataset. The size of the gallery set is 532.

Methods	r = 1	r = 5	r = 10	r = 20
PCCA [28]	9.3	24.9	37.4	52.9
RPML [14]	10.9	26.7	37.7	51.6
LAFT [20]	**12.9**	**30.3**	**42.7**	**58.0**
Ours	**17.4**	**41.6**	**55.3**	**70.8**

same at rank 1, but their collaboration can improve the results from about 29% to 36.8%, which indicates the complementary properties of the two parts.

The effect of the latent formulation. Latent formulation introduces H similarity functions to account for different matching patterns and learns one similarity function to prevent over-fitting. To evaluate how much our method can benefit from learning multiple functions, we reduce our method to learn a single matching function by only learning the common weight $\mathbf{u}_0$ with $H = 0$, denoted as "Single". The comparison between Single and original one are demonstrated in Figure 2(c). The latent formulation can improve the rank-1 matching rate from 33.8% to 36.7%. Besides, we show how the number of similarity function can influence the performance in Figure 4.

The effect of the loss function. To investigate influence of choosing different loss functions, we compare our method with triplet loss and with binary loss. The binary loss is the counterpart of the triplet loss in Equation 4:

$$\ell_{\text{binary}}(\mathbf{x}_i, \mathbf{x}_j) = [1 - (f(\mathbf{x}_i, \mathbf{x}_j) + b)y_{ij}]_+, \qquad (19)$$

where y_{ij} equals 1 or -1 indicating whether $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to a same identity, and b is the threshold value. For binary loss, as the number of negative samples is much larger than that of the positive samples, we adopt the hard negative sample mining strategy [10]. The overall performance of triplet loss is better than binary loss as shown in Figure 2(d). The difference between triplet loss and binary loss is that triplet loss utilizes the identity information to optimize the relative similarity according to the query images, while binary loss tries to differentiate all the correct matches from incorrect matches.

Table 5: The rank-n matching rates(%) for comparison with other methods on the GRID dataset. The size of the gallery set is 900.

Methods	r = 1	r = 5	r = 10	r = 15	r = 20
PRDC [41]	9.7	22.0	33.0	40.0	44.3
RankSVM [31]	10.2	24.6	33.3	39.4	43.7
MRank-PRDC[23]	11.1	26.1	35.8	41.8	46.6
MRank-rankSVM[23]	12.2	27.8	36.3	42.2	46.6
RQDA [22]	**15.2**	**30.1**	**39.2**	**44.7**	**49.3**
Ours	**16.3**	**35.8**	**46.0**	**52.8**	**57.6**

Table 6: The rank-n matching rates(%) for comparison with other methods on the CAVIAR4REID dataset. The size of the gallery set is 36.

Methods	r = 1	r = 5	r = 10	r = 20
LF [29]	35.2	59.9	73.7	88.8
PCCA-χ^2_{rbf} [36]	33.2	65.9	81.9	95.2
LADF [21]	30.3	62.8	**78.0**	**92.6**
MFA-χ^2 [36]	**40.2**	**70.2**	**83.9**	**95.1**
Ours	**40.1**	**65.6**	**78.0**	90.5

The effect of hyper-parameters. We study the influence of two parameters, the parameter λ of the $L1$ sparsity regularization in Equation 6 and the penalty parameter ρ in Equation 15. We show how the performance changes with respect to λ in Figure 4 (a) by fixing $\rho = 30$, and show the influence of ρ in Figure 4 (b) by fixing $\lambda = 0.1$. It can be seen that both too large λ and too small λ lead to inferior performances, and the influence of ρ is a little complex. We use cross validation to select the two parameters.

6.3. Results

VIPER. The VIPER dataset contains 632 persons. For each person there are two 48 × 128 images taken from camera A and B under different viewpoints, poses and illumination conditions. Two protocols were used for the evaluation: randomly selecting 316 persons to form the training set and the remaining 316 persons to form the testing set; and randomly selecting 100 persons to form the training set and the remaining 532 persons to form the testing set. We evaluate the performance using both protocols, and present the results in Table 3 and Table 4. Besides, we also plot the CMC curves in Figure 5. It can be seen that our method significantly outperforms other state-of-arts under both the two protocols.

GRID. The GRID dataset consists of person images captured from 8 disjoint camera views installed in a busy underground station. The probe set contains 250 persons, and the gallery set contains 1025 persons where 775 persons do not match any person in the probe set. We process the images by resizing the images into 300 × 100. We divide each image evenly into 150 overlapped patches, and describe each patch using the concatenation of HSV and LAB histograms (24 bin for each channel), a LBP descriptor with 8 bins, and a SIFT feature with 8 bins. All the patch features are concatenated together to form the image feature, which is further reduced to a 45-dimensional feature through PCA.

We conduct the experiments directly over the 10 partitions provided by the GRID dataset, where 125 image pairs are used for training, 125 image pairs and 775 irrelevant images are used for testing. Figure 5 (c) and Table 5 show the CMC curves and matching rates for our method and recent published results under the same protocol. One can see that our method consistently achieves the best results and that the superiority becomes significant with the larger rank.

CAVIAR4REID. The images in the CAVIAR4REID dataset are cropped from 26 sequences captured by two cameras in a shopping center. It contains 72 persons in total, and for each person there are 10 or 20 images collected from one or two video sequence. We resize the images into 90 × 30, and divide each image into 45 overlapped patches. Each patch is described by the feature same to the one used for the GRID dataset. All the patch features are concatenated together to form the image feature, which is further reduced to a 80-dimensional feature.

We use the same experimental protocol with [36, 21], where 36 persons are used for training and other 36 persons are used for testing. Our approach performs the second best from Table 6. The reason might be that the number of identities in the training set is very small. It is valuable to integrate the kernel trick used in MFA [36], which performs the best in this dataset, into our approach to further improve the performance.

6.4. Results for face verification

Face verification is a binary classification problem to tell whether two images belong to a same person. We use the binary loss in Equation 19 to replace the triplet loss in Equation 4, and study the performance for face verification on the LFW dataset [15]. We follow the *image-restricted, no outside* data protocol [15], in which the dataset is split into 10 folds with each containing 300 positive pairs and 300 negative pairs. The classifier is learnt from 9 folds and tested over the remaining 1 fold. No external data for strong face alignment, feature extraction or recognition model training is allowed. We do the evaluation by 10-fold cross-validation and report the performance in terms of mean accuracy and standard error. We employ the 67584-dimensional fisher vector introduced in [32] to represent the face image, and reduce it to be a 150-dimensional vector via PCA.

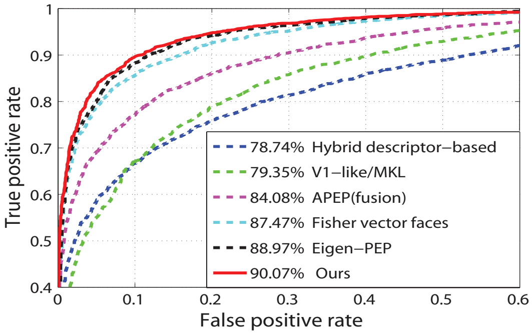

Figure 6: **Comparison with the state-of-the-arts**: ROC curves of our method and the state-of-the-art techniques on the LFW dataset, under restricted, no outside data protocol. The accuracy is marked before the name of each approach.

Table 7: Comparing our method and its variants on the LFW dataset. The variants includes using different pairwise features and regularization strategies as introduced in the text. The results are in terms of mean accuracy and standard error.

Methods	Accuracy	Methods	Accuracy
F1	87.95 ± 1.17	NSS	85.48 ± 1.34
F2	88.43 ± 1.39	SP	87.42 ± 1.65
F3	83.22 ± 0.98	SD	89.23 ± 1.13
Single	89.15 ± 1.17	Ours	90.07 ± 1.07

We compare our approach with the competing methods, including the hybrid descriptor-based method [35], V1-like/MLK [30], APEP (fusion) [18], the fisher vector face [32] and Eigen-PEP [19]. As shown in Figure 6, our method outperforms the current state-of-the-arts. Particularly, our approach gets the 2.6% improvement over the fisher vector face [32] that uses the same feature.

In addition, we present the empirical analysis result to compare our method with its variants which are similar to the variants for the re-identification task in the Table 2. The results given in Table 7, not only demonstrate the effectiveness of the proposed similarity function from the comparison among F1, F2, F3, single and ours but also show the significance of the regularization from the comparison among NSS, SP, SD and ours.

7. Conclusion

We present a novel similarity learning approach to person re-identification. The success of our approach stems from the robust pairwise feature - explicit polynomial kernel feature map, which leaves the data to determine the importance of the patch match degree instead of keeping the best matched patch, a mixture of similarity functions, which is able to discover different similarity patterns, as well as two regularization terms that increase the generalization ability of our approach.

Acknowledgement. This work was supported by the National Basic Research Program of China (Grant No.2015CB351703), the State Key Program of National Natural Science Foundation of China (Grant No.61231018), the 111 Project (Grant No.B13043), and the Fundamental Research Funds for the Central Universities. It was also supported by the National Institute Of Nursing Research of the National Institutes of Health(R01NR015371), US National Science Foundation (Grant IIS 1350763) and GH's start-up funds form Stevens Institute of Technology.

References

[1] S. Bak, E. Corvée, F. Brémond, and M. Thonnat. Person reidentification using spatial covariance regions of human body parts. In *International Conference on Advanced Video and Signal Based Surveillance*, 2010. 2

[2] L. Bazzani, M. Cristani, A. Perina, and V. Murino. Multiple-shot person re-identification by chromatic and epitomic analyses. *Pattern Recognition Letters*, 33(7):898–903, 2012. 1, 5

[3] S. Chang, G. Qi, C. Aggarwal, J. Zhou, M. Wang, and T. Huang. Factorized similarity learning in networks. In *ICDM*, 2014. 1

[4] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision*, 2012. 5

[5] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *British Machine Vision Conference*, 2011. 1, 2

[6] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja. Pedestrian recognition with a learned metric. In *Asian Conference on Computer Vision*, 2010. 2

[7] M. Dikmen, D. Hoiem, and T. S. Huang. A data driven method for feature transformation. In *Computer Vision and Pattern Recognition*, 2012. 5

[8] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *International Conference on Knowledge Discovery and Data Mining*, 2004. 5

[9] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition*, 2010. 1, 2

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell*, 32(9):1627–1645, 2010. 5, 7

[11] M. Fornoni, B. Caputo, and F. Orabona. Multiclass latent locally linear support vector machines. In *Asian Conference on Machine Learning, ACML 2013, Canberra, ACT, Australia, November 13-15, 2013*, 2013. 5

[12] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *International Workshop on PETS, Rio de Janeiro*, 2007. 1, 2, 5, 6

[13] Q. Gu and J. Han. Clustered support vector machines. In *AISTATS*, 2013. 5

[14] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof. Relaxed pairwise learned metric for person re-identification. In *European Conference on Computer Vision*, 2012. 1, 2, 7

[15] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 5, 8

[16] H. Jégou and O. Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *European Conference on Computer Vision*, 2012. 5

[17] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition*, 2012. 1, 2

[18] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic matching for pose variant face verification. In *Computer Vision and Pattern Recognition*, 2013. 8

[19] H. Li, G. Hua, X. Shen, L. Zhe, and J. Brandt. Eigen-pep for video face recognition. In *ACCV*, 2014. 8

[20] W. Li and X. Wang. Locally aligned feature transforms across views. In *Computer Vision and Pattern Recognition*, 2013. 1, 2, 7

[21] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *Computer Vision and Pattern Recognition*, 2013. 1, 2, 5, 7, 8

[22] S. Liao, Y. Hu, and S. Z. Li. Joint dimension reduction and metric learning for person re-identification. *CoRR*, abs/1406.4216, 2014. 7

[23] C. C. Loy, C. Liu, and S. Gong. Person re-identification by manifold ranking. In *ICIP*, 2013. 5, 7

[24] B. Ma, Y. Su, and F. Jurie. Bicov: a novel image representation for person re-identification and face verification. In *British Machine Vision Conference*, 2012. 1, 2

[25] B. Ma, Y. Su, and F. Jurie. Local descriptors encoded by fisher vectors for person re-identification. In *ECCV Workshops and Demonstrations*, 2012. 2

[26] L. Ma, X. Yang, and D. Tao. Person re-identification over camera networks using multi-task distance metric learning. *IEEE Transactions on Image Processing*, 23(8):3656–3670, 2014. 1, 2

[27] S. Maji and A. C. Berg. Max-margin additive classifiers for detection. 2

[28] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition*, 2012. 1, 2, 6, 7

[29] S. Pedagadi, J. Orwell, S. A. Velastin, and B. A. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *Computer Vision and Pattern Recognition*, 2013. 1, 2, 5, 7

[30] N. Pinto, J. J. DiCarlo, and D. D. Cox. How far can you get with a modern face recognition test set using only simple features? In *Computer Vision and Pattern Recognition*, 2009. 8

[31] B. Prosser, W. Zheng, S. Gong, and T. Xiang. Person re-identification by support vector ranking. In *British Machine Vision Conference*, 2010. 1, 2, 7

[32] K. Simonyan, A. Parkhi, Omkarand Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference*, 2013. 8

[33] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE TPAMI.*, 34(3):480–492, 2012. 2

[34] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. H. Tu. Shape and appearance context modeling. In *International Conference on Computer Vision*, 2007. 1

[35] L. Wolf, T. Hassner, Y. Taigman, et al. Descriptor based methods in the wild. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008. 8

[36] F. Xiong, M. Gou, O. I. Camps, and M. Sznaier. Person reidentification using kernel-based metric learning methods. In *European Conference on Computer Vision*, 2014. 2, 6, 7, 8

[37] R. Zhao, W. Ouyang, and X. Wang. Person re-identification by salience matching. In *International Conference on Computer Vision*, 2013. 1, 2

[38] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *Computer Vision and Pattern Recognition*, 2013. 1, 2

[39] R. Zhao, W. Ouyang, and X. Wang. Learning mid-level filters for person re-identification. In *Conference on Computer Vision and Pattern Recognition*, 2014. 7

[40] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian. Query-adaptive late fusion for image search and person re-identification. In *Computer Vision and Pattern Recognition*, 2015. 1

[41] W. Zheng, S. Gong, and T. Xiang. Reidentification by relative distance comparison. *IEEE Trans. Pattern Anal. Mach. Intell*, 35(3):653–668, 2013. 2, 5, 7