Kernel Distance Metric Learning Using Pairwise Constraints for Person Re-Identification

Bac Nguyen and Bernard De Baets

***Abstract*— Person re-identification is a fundamental task in many computer vision and image understanding systems. Due to appearance variations from different camera views, person re-identification still poses an important challenge. In the literature, KISSME has already been introduced as an effective distance metric learning method using pairwise constraints to improve the re-identification performance. Computationally, it only requires two inverse covariance matrix estimations. However, the linear transformation induced by KISSME is not powerful enough for more complex problems. We show that KISSME can be kernelized, resulting in a nonlinear transformation, which is suitable for many real-world applications. Moreover, the proposed kernel method can be used for learning distance metrics from structured objects without having a vectorial representation. The effectiveness of our method is validated on five publicly available data sets. To further apply the proposed kernel method efficiently when data are collected sequentially, we introduce a fast incremental version that learns a dissimilarity function in the feature space without estimating the inverse covariance matrices. The experiments show that the latter variant can obtain competitive results in a computationally efficient manner.**

***Index Terms*— Distance metric learning, dissimilarity learning, person re-identification, kernel-based learning.**

I. Introduction

IN RECENT years, the deployment of camera networks has grown exponentially in wide-area public spaces, such as railway stations, airports, and office buildings. As a result, many applications in person re-identification demand fast and effective techniques that are capable of accurately searching images from video surveillance (see e.g. [1] and the references therein). Given an image of a person, the main task in person re-identification is to identify the person from images taken at a different location and/or from a different viewpoint across non-overlapping cameras. It is important to remark that when a person disappears from one camera, he/she can be recognized from other cameras. A good system should be able to keep track of a person throughout the network, i.e. the appearances of the same person from different cameras have to be matched.

Manuscript received November 20, 2017; revised May 18, 2018 and August 10, 2018; accepted September 13, 2018. Date of publication September 20, 2018; date of current version October 1, 2018. This work was supported by the Special Research Fund–Doctoral Scholarships, Ghent University, Belgium, under Grant BOF15/DOS/039. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Abd-Krim K. Seghouane. *(Corresponding author: Bac Nguyen.)*

The authors are with the Department of Data Analysis and Mathematical Modeling, Ghent University, 9000 Ghent, Belgium (e-mail: bac.nguyencong@ugent.be; bernard.debaets@ugent.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2870941

Fig. 1. An illustration of challenges in person re-identification (from left to right): different backgrounds, resolution, pose, view angle, lighting, partial occlusion, and similar clothings.

Person re-identification is a highly challenging problem, even for humans, due to the difficulty in characterizing the appearance and computing the similarity between images [1]–[3]. These difficulties are mainly caused by changing view angles, resolution, lighting, occlusions, and so on. See Fig. 1 for an illustration of challenges in person re-identification.

In order to find the correct match for a *probe* image from a set of *gallery* images captured by different cameras, two steps are employed. We first extract features from both probe and gallery images using a suitable feature extraction method. The identification results are then obtained by ranking the similarities between the probe and gallery images. Accordingly, the re-identification performance is measured by the top rank k matching rate, which is the percentage of probe images with correct matches found in the top-k ranked gallery images. That is why person re-identification can be formulated as a ranking problem [4]. Consequently, having an effective feature representation and a good distance metric can improve significantly the performance of re-identification [3].

Most of the existing studies focus on extracting more relevant or informative features that are able to discriminate different appearance patterns. A number of effective methods have been proposed to perform feature extraction for an image, including the scale invariant feature transform (SIFT) [5], the ensemble of local features (ELF) [6], local binary patterns (LBP) [7], Fisher vectors (LDFV) [8], and weighted histograms of overlapping stripes (WHOS) [9].

These handcrafted descriptors allow to significantly improve the performance of person re-identification. However, computing a set of representative and robust features is not always an easy task due to cross-view variations in appearance of images. Another interesting approach is to use a tensor representation rather than a vectorial representation for the input data [10], [11]. Under several realistic viewing changes, most visual features and their combinations are neither stable nor reliable. In contrast to using complex handcrafted features computed from the raw images, deep convolutional neural networks (DCNNs) have been exploited to learn a set of representative features that captures the variability of person appearance across views [12]–[14]. One of the major problems with DCNNs is that they often require the availability of a huge number of images to obtain a model that is generalizable to data beyond the training set.

A recent trend tries to learn a good distance metric by implicitly suppressing those cross-view variations between images [15], [16]. This is motivated by the fact that standard distance metrics, e.g. the Euclidean or Manhattan distance metric, are not reliable and flexible enough because they usually assume that all features are from the same domain with the same scale. Consequently, they become more sensitive to irrelevant features and fail to preserve the geometric characteristics of the data [17]. An ideal distance metric should accurately reflect the true underlying relationships between images, i.e. small distances for similar images and large distances for dissimilar or unrelated images. Previous studies [2], [15], [16], [18]–[20] have shown that optimizing a distance metric can significantly improve the performance of person re-identification.

The distance metric used may not fully reflect human judgments of dissimilarity without additional information from the users or from the training examples, such as class labels. One way to provide this information is through a set of constraints. As is common in person re-identification, we describe the information in the form of must-link and cannot-link pairwise constraints. Must-link constraints, e.g. images of the same person, are used to specify that the two examples should be in the same class. Cannot-link constraints, e.g. images of different persons, are used to specify that the two examples should be in different classes. These pairwise constraints have the following advantages that enable them to be applied in a wide range of application domains. First, collecting fully labeled training examples is a difficult task and also time-consuming. Particularly, annotating images with identity from every camera is prohibitively expensive in a large camera network. Second, it is often easier to collect pairwise relations, which are usually expressed in the form of pairwise constraints. The pairwise relations can be obtained, for instance, through interacting with the users by asking feedback whether two images are from the same person or not. Unlike the general procedure of asking feedback in the form of annotating images with exact labels, the users are not required to have experience or prior knowledge with the data set.

Given a set of constraints, distance metric learning is mostly cast as solving a convex optimization problem over the cone of positive semidefinite matrices. While many efforts [21], [22] have been devoted to reduce the computational complexity of semidefinite programming, they still require an expensive iterative optimization procedure. Based on a statistical inference perspective, Köstinger *et al.* [16] introduced a pairwise distance metric learning approach named KISSME to avoid this computational burden. KISSME has the advantage of being simple and obtains a good recognition rate in person re-identification [23]. One of the main problems is that KISSME may yield rather poor estimates of covariance matrices when the number of constraints is small, thus leading to a poor generalization ability. Several extensions [24], [25] have been proposed to address this problem, however, they are still limited to the use of a linear transformation and cannot capture the nonlinear structure of the input space. It is also important to note that KISSME can suffer from the curse of dimensionality in high-dimensional settings, just like other conventional distance metric learning methods that parameterize the distance metric by a matrix that scales quadratically with the dimensionality.

A common guiding principle for learning a distance metric from pairwise constraints is that the distances between examples in must-link constraints should be small, while the distances between those in cannot-link constraints should be large. Additionally, there are also several requirements for a good distance metric learning method: (1) it should reflect the true similarity relationships between examples in order to generalize well to unseen examples; (2) it should be easy to implement and to compute efficiently; (3) it should be flexible enough to handle different learning settings and data types. Based on these considerations, this paper presents the following two main contributions:

1) We propose the use of kernels for KISSME, named k-KISSME, which allows to capture the nonlinear structure in a data set. Our method operates in the kernel spaces, yielding a highly flexible distance metric. Compared to the original KISSME method, k-KISSME is not only more robust, but can also be used for naturally structured objects that have no vectorial representation.
2) Most of the kernel methods employ a "batch" setting, i.e. all examples need to be available during training. Unfortunately, in applications like video surveillance where images are collected sequentially, processing the whole data set upon the arrival of a new pairwise constraint can be computationally expensive. To alleviate this computational burden, we present an incremental update strategy for k-KISSME.

In the next section, we briefly review some of the most relevant works on person re-identification. Basic notations and definitions are introduced in Section III. Later on, in Section IV, we will revisit KISSME. Its kernel version, i.e. k-KISSME, is presented in Section V. Subsequently, we show that k-KISSME can be incrementally updated by relaxing the positive semidefiniteness constraint. Experiments on person re-identification benchmarks are conducted in Section VI, followed by some concluding remarks and future works in Section VII.

II. Related Work

In this section, we briefly review various relevant methods for learning an optimal distance metric in supervised settings that have been successfully applied to person re-identification tasks.

Typically, the supervision is induced in the form of pairwise constraints, i.e. must-link and cannot-link constraints. In the context of face identification, Guillaumin *et al.* [26] introduced logistic discriminant metric learning (LDML), which aims to make the distances between examples of similar pairs smaller than the distances between those of dissimilar pairs. Based on pairwise constraints, Davis *et al.* [27] formulated distance metric learning as a LogDet optimization problem, which can enforce the positive semidefiniteness constraint automatically to avoid the projection onto the positive semidefinite cone. Interestingly, Hirzer *et al.* [15] showed that relaxing the positive semidefiniteness constraint can dramatically simplify the problem of learning a Mahalanobis distance metric while still guaranteeing promising results. Recently, Sun *et al.* [20] presented a person re-identification framework based on distance metric learning with latent variables. Yang *et al.* [18] used only must-link constraints to learn an effective similarity function. The method most closely related to ours is the KISSME method proposed by Köstinger *et al.* [16], which will be discussed in Section IV. To perform KISSME in high-dimensional settings, Liao *et al.* [3] employed the generalized Rayleigh quotient to find a discriminant low-dimensional subspace in which to perform the KISSME method. Tao *et al.* [28] showed that the performance of the latter can be further improved when using deep learning features in conjunction with handcrafted features. Another extension of KISSME was proposed by Tao *et al.* [24], including a smoothing technique to improve the estimation of the covariance matrices. In [29], Zhao *et al.* considered a QR decomposition that maps the data into a low-dimensional space and subsequently perform KISSME to learn a robust Mahalanobis matrix in the projected space.

Triplet constraints are another common form of supervision, i.e. object $\mathbf{x}_i$ is more similar to object $\mathbf{x}_j$ than to object $\mathbf{x}_l$. In [22], Weinberger and Saul introduced the large-margin nearest neighbor (LMNN) method that aims to pull target neighbors (of the same class) close together while pushing impostor neighbors (of different classes) far apart. LMNN performs well for k-nearest-neighbor (k-NN) classification. In order to handle the rejection case for k-NN, which is quite common in person re-identification tasks, Dikmen *et al.* [30] proposed LMNN with rejection (LMNN-R). Similarly, Zheng *et al.* [31] proposed a probabilistic relative distance comparison (PRDC) method that maximizes the probability of a correct-match pair having a smaller distance than that of an incorrect-match pair.

Due to large variations in pose and illumination changes, it is unlikely that a linear transformation induced by the Mahalanobis distance metric can discriminate individuals satisfactorily. Instead of operating directly in the original input space, Xiong *et al.* [32] introduced the use of kernels in order to learn a distance metric in the feature space. In doing so, we obtain a more flexible linear transformation in the feature space, which can be applied inductively to new examples. Although kernelized versions of various distance metric learning methods exist [17], [27], [33], kernelizing a distance metric learning method is not always a trivial and straightforward task. In this paper, we show how to kernelize KISSME, making it more efficient and robust to person re-identification tasks.

III. Notations

The following notations will be used throughout this paper. We denote vectors by boldface lowercase letters and matrices by boldface uppercase letters. All scalars are denoted by lowercase or uppercase letters. Sets are denoted by calligraphic uppercase letters. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. The identity matrix is denoted by $\mathbf{I}$. The diagonal vector of a square matrix $\mathbf{M}$ is denoted by $\mathrm{diag}(\mathbf{M})$. We will use $\mathbf{K}_{i.}$ to refer to the i-th row vector and $\mathbf{K}_{.j}$ to refer to the j-th column vector of a matrix $\mathbf{K}$. A symmetric matrix $\mathbf{M} \in \mathbb{R}^{D\times D}$ is positive semidefinite (PSD), denoted by $\mathbf{M} \succcurlyeq 0$, if and only if, for any vector $\mathbf{x} \in \mathbb{R}^D$, the following condition holds: $\mathbf{x}^\top \mathbf{M}\mathbf{x} \geq 0$. We denote the projection of a matrix $\mathbf{M}$ onto the cone of PSD matrices $\mathbb{S}^+$ by $\Pi_{\mathbb{S}^+}(\mathbf{M}) = \sum_{i:\lambda_i>0} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$, where $(\lambda_i, \mathbf{u}_i)$ is the i-th pair of eigenvalue and eigenvector of $\mathbf{M}$.

Let $\mathcal{S}$ and $\mathcal{D}$ denote the set of must-link and cannot-link pairwise constraints obtained from a training set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^D$, respectively, i.e.

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ should be similar}\},$$
$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ should be dissimilar}\}.$$

Our main goal is to estimate a Mahalanobis distance metric,

$$d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j),$$

where $\mathbf{M} \succcurlyeq 0$, with the aim of satisfying as many as possible of the constraints in $\mathcal{S}$ and $\mathcal{D}$. We will denote by $\mathbf{X}$ the matrix whose columns contain the training examples.

IV. KISSME Revisited

To motivate our approach, we briefly review KISSME as introduced in [16]. Let us consider the difference $\mathbf{x}_i - \mathbf{x}_j$ between two examples $\mathbf{x}_i$ and $\mathbf{x}_j$. Consequently, two disjoint probability spaces of differences are defined, Ω_0 for differences of examples from different classes and Ω_1 for those from the same class. Let p_0 and p_1 denote the probability density functions of differences in Ω_0 and Ω_1, respectively. A possible way to verify whether or not $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class is through the use of a log-likelihood ratio statistic:

$$\sigma(\mathbf{x}_i, \mathbf{x}_j) = \log\left(\frac{p_0(\mathbf{x}_i - \mathbf{x}_j)}{p_1(\mathbf{x}_i - \mathbf{x}_j)}\right). \tag{1}$$

A high value of $\sigma(\mathbf{x}_i, \mathbf{x}_j)$ indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ likely belong to different classes. In contrast, a low value of $\sigma(\mathbf{x}_i, \mathbf{x}_j)$ indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ likely belong to the same class. Assuming that the differences in Ω_0 and Ω_1

are normally distributed with zero mean, Eq. (1) can be rewritten as

$$\begin{aligned}&\sigma(\mathbf{x}_i,\mathbf{x}_j)\\&=\log\left(\frac{\frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}_0|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_j)^\top\mathbf{\Sigma}_0^{-1}(\mathbf{x}_i-\mathbf{x}_j)\right)}{\frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}_1|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_j)^\top\mathbf{\Sigma}_1^{-1}(\mathbf{x}_i-\mathbf{x}_j)\right)}\right)\\&=\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_j)^\top(\mathbf{\Sigma}_1^{-1}-\mathbf{\Sigma}_0^{-1})(\mathbf{x}_i-\mathbf{x}_j)+\log\left(\frac{|\mathbf{\Sigma}_1|}{|\mathbf{\Sigma}_0|}\right),\end{aligned}$$

where $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$ denote the covariance matrices of p_0 and p_1, respectively. Note that the zero mean assumption was also argued by Moghaddam *et al.* [34] in a similar formulation as for each sample $\mathbf{x}_i-\mathbf{x}_j$ there always exists a sample $\mathbf{x}_j-\mathbf{x}_i$. Since the constant terms do not affect the log-likelihood ratio statistic for use in statistical hypothesis testing, we can simplify it to

$$\sigma(\mathbf{x}_i,\mathbf{x}_j)=(\mathbf{x}_i-\mathbf{x}_j)^\top(\mathbf{\Sigma}_1^{-1}-\mathbf{\Sigma}_0^{-1})(\mathbf{x}_i-\mathbf{x}_j)\,.$$

Finally, learning a Mahalanobis distance metric amounts to estimating two inverse covariance matrices, i.e. $\mathbf{M}=\mathbf{\Sigma}_0^{-1}-\mathbf{\Sigma}_1^{-1}$, as σ and $d_{\mathbf{M}}$ share very similar properties. To guarantee that $d_{\mathbf{M}}$ is a distance metric, we use instead the projection of $(\mathbf{\Sigma}_0^{-1}-\mathbf{\Sigma}_1^{-1})$ onto the cone of PSD matrices. Using maximum likelihood estimation, the covariance matrices $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$ are computed as follows

$$\mathbf{\Sigma}_0=\frac{1}{n_0}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top, \tag{2}$$

$$\mathbf{\Sigma}_1=\frac{1}{n_1}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top, \tag{3}$$

where $n_0=|\mathcal{D}|$ and $n_1=|\mathcal{S}|$.

As another alternative to the use of $\sigma(\mathbf{x}_i-\mathbf{x}_j)$, one may argue that a high value of $p_1(\mathbf{x}_i-\mathbf{x}_j)$ can indicate that $\mathbf{x}_i$ and $\mathbf{x}_j$ likely belong to the same class and a low value of $p_1(\mathbf{x}_i-\mathbf{x}_j)$ can indicate that $\mathbf{x}_i$ and $\mathbf{x}_j$ likely belong to different classes. Accordingly, the distance metric is only parameterized by the inverse of the covariance matrix $\mathbf{\Sigma}_1$, which is defined as the Mahalanobis distance between an example and a normal distribution. From this point of view, KISSME can be regarded as an extension of relevant component analysis (RCA) [35], a simple method for learning distance metrics using only must-link constraints.

Although KISSME is very effective on low-dimensional data sets, it quickly becomes intractable when increasing the number of features. This is due to the fact that KISSME has a high memory complexity $O(D^2)$, which is prohibitive for many applications that involve thousands of features. Besides, computing the inverse covariance matrices is expensive and tends to be an ill-posed inverse problem as the covariance matrices are likely to be singular in higher dimensions. Next, we consider the idea of using kernels to overcome these limitations.

V. Kernel Distance Metric Learning

In this section, we propose a nonlinear variant of KISSME. By introducing a regularizer into the covariance matrices, our method k-KISSME becomes more robust and stable. Moreover, to avoid recomputation of k-KISSME on the arrival of a new constraint, which is computationally expensive, an incremental version of k-KISSME is developed.

A. Kernel KISSME

The idea of kernel methods is to implicitly perform a nonlinear map ϕ from the input space $\mathcal{X}$ into a high-dimensional feature space $\mathcal{F}$, i.e. $\phi\colon\mathcal{X}\to\mathcal{F}$, by replacing the inner product with an appropriate positive semidefinite function. Formally, for any PSD kernel matrix $\mathbf{K}$, there exists a nonlinear map ϕ such that $K_{ij}=\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$. The matrix $\mathbf{K}$ can be computed efficiently using a kernel function $\mathcal{K}$ that computes the inner product between two examples in the feature space without carrying out the explicit map, i.e. $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)=\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$. Several kernel functions, such as polynomials, radial basis functions, and exponential χ^2 kernel functions, have been successfully used in the context of distance metric learning [17], [27], [32]. Motivated by the fact that kernel methods can overcome many limitations of its linear counterpart, in this subsection, we describe how to kernelize KISSME. Clearly, a direct computation of the inverse covariance matrices $\mathbf{\Sigma}_0^{-1}$ and $\mathbf{\Sigma}_1^{-1}$ is not feasible since the dimensionality of $\mathcal{F}$ is too high, or even infinite.

Assuming that the pairwise constraints in $\mathcal{S}$ and $\mathcal{D}$ are given, we start by introducing some notations. Let $\mathbf{1}_i$ be a column vector that has the value 1 at the i-th entry and 0 at the other entries. Let $\mathbf{B}_0$ (resp. $\mathbf{B}_1$) be an $n\times n$ diagonal matrix whose diagonal vector contains at the i-th entry the number of constraints in $\mathcal{D}$ (resp. $\mathcal{S}$) of which the first element is $\mathbf{x}_i$, i.e.

$$\begin{aligned}\operatorname{diag}(\mathbf{B}_0)_i&=|\{j\mid j\in\{1,\ldots,n\}\text{ and }(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}\}|,\\\operatorname{diag}(\mathbf{B}_1)_i&=|\{j\mid j\in\{1,\ldots,n\}\text{ and }(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}\}|.\end{aligned}$$

Let $\mathbf{E}_0$ (resp. $\mathbf{E}_1$) be an $n\times n$ diagonal matrix whose diagonal vector contains at the j-th entry the number of constraints in $\mathcal{D}$ (resp. $\mathcal{S}$) of which the second element is $\mathbf{x}_j$, i.e.

$$\begin{aligned}\operatorname{diag}(\mathbf{E}_0)_j&=|\{i\mid i\in\{1,\ldots,n\}\text{ and }(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}\}|,\\\operatorname{diag}(\mathbf{E}_1)_j&=|\{i\mid i\in\{1,\ldots,n\}\text{ and }(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}\}|.\end{aligned}$$

Let $\mathbf{W}_0$ (resp. $\mathbf{W}_1$) be an $n\times n$ matrix whose entry at the i-th row and j-th column is 1 if $(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}$ (resp. $(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}$), otherwise it takes value 0. Using the preceding notations, we can rewrite the matrix $\mathbf{\Sigma}_0$ in Eq. (2) as

$$\begin{aligned}\mathbf{\Sigma}_0&=\frac{1}{n_0}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}}\left(\mathbf{x}_i\mathbf{x}_i^\top-\mathbf{x}_j\mathbf{x}_i^\top-\mathbf{x}_i\mathbf{x}_j^\top+\mathbf{x}_j\mathbf{x}_j^\top\right)\\&=\frac{1}{n_0}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}}\Big(\mathbf{X}\mathbf{1}_i\mathbf{1}_i^\top\mathbf{X}^\top-\mathbf{X}\mathbf{1}_j\mathbf{1}_i^\top\mathbf{X}^\top\\&\qquad-\mathbf{X}\mathbf{1}_i\mathbf{1}_j^\top\mathbf{X}^\top+\mathbf{X}\mathbf{1}_j\mathbf{1}_j^\top\mathbf{X}^\top\Big)\\&=\frac{1}{n_0}\mathbf{X}\left(\mathbf{B}_0-\mathbf{W}_0^\top-\mathbf{W}_0+\mathbf{E}_0\right)\mathbf{X}^\top\\&=\frac{1}{n_0}\mathbf{X}\mathbf{H}_0\mathbf{X}^\top,\end{aligned}$$

where $\mathbf{H}_0 = \mathbf{B}_0 - \mathbf{W}_0^\top - \mathbf{W}_0 + \mathbf{E}_0$. Similarly, we can rewrite the covariance matrix $\boldsymbol{\Sigma}_1$ in Eq. (3) as

$$\boldsymbol{\Sigma}_1 = \frac{1}{n_1}\mathbf{X}\mathbf{H}_1\mathbf{X}^\top,$$

where $\mathbf{H}_1 = \mathbf{B}_1 - \mathbf{W}_1^\top - \mathbf{W}_1 + \mathbf{E}_1$. Note that $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ can be singular due to the lack of sufficient pairwise constraints. Therefore, to avoid the problem of inverting a singular matrix, we propose the use of a regularizing term by adding some small positive constant value ϵ to the diagonals of $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$, i.e.

$$\widehat{\boldsymbol{\Sigma}}_0 = \epsilon\mathbf{I} + \frac{1}{n_0}\mathbf{X}\mathbf{H}_0\mathbf{X}^\top, \quad \widehat{\boldsymbol{\Sigma}}_1 = \epsilon\mathbf{I} + \frac{1}{n_1}\mathbf{X}\mathbf{H}_1\mathbf{X}^\top. \tag{4}$$

According to Friedman [36], this method can obtain a more robust and stable estimation than using maximum likelihood estimation. To evaluate the inverses of these matrices, we consider the Kailath formula [37] given by

$$(\mathbf{A} + \mathbf{B}\mathbf{D})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}. \tag{5}$$

Applying Eq. (5) to the covariance matrices in (4), results in

$$\widehat{\boldsymbol{\Sigma}}_0^{-1} = \frac{1}{\epsilon}\mathbf{I} - \frac{1}{n_0\epsilon^2}\mathbf{X}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{n_0\epsilon}\mathbf{X}^\top\mathbf{X}\mathbf{H}_0\right)^{-1}\mathbf{X}^\top,$$
$$\widehat{\boldsymbol{\Sigma}}_1^{-1} = \frac{1}{\epsilon}\mathbf{I} - \frac{1}{n_1\epsilon^2}\mathbf{X}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{n_1\epsilon}\mathbf{X}^\top\mathbf{X}\mathbf{H}_1\right)^{-1}\mathbf{X}^\top.$$

Finally, the difference between these two inverse covariance matrices can be computed as

$$\begin{aligned}
\widehat{\boldsymbol{\Sigma}}_1^{-1} - \widehat{\boldsymbol{\Sigma}}_0^{-1} &= \mathbf{X}\left[\frac{1}{n_0\epsilon^2}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{n_0\epsilon}\mathbf{X}^\top\mathbf{X}\mathbf{H}_0\right)^{-1}\right. \\
&\qquad \left. - \frac{1}{n_1\epsilon^2}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{n_1\epsilon}\mathbf{X}^\top\mathbf{X}\mathbf{H}_1\right)^{-1}\right]\mathbf{X}^\top \\
&= \mathbf{X}\left[\frac{1}{n_0\epsilon^2}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{n_0\epsilon}\mathbf{K}\mathbf{H}_0\right)^{-1}\right. \\
&\qquad \left. - \frac{1}{n_1\epsilon^2}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{n_1\epsilon}\mathbf{K}\mathbf{H}_1\right)^{-1}\right]\mathbf{X}^\top \\
&= \mathbf{X}\mathbf{C}\mathbf{X}^\top,
\end{aligned}$$

where $\mathbf{K} = \mathbf{X}^\top\mathbf{X}$ denotes the $n \times n$ kernel matrix and

$$\mathbf{C} = \frac{1}{n_0\epsilon^2}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{n_0\epsilon}\mathbf{K}\mathbf{H}_0\right)^{-1} - \frac{1}{n_1\epsilon^2}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{n_1\epsilon}\mathbf{K}\mathbf{H}_1\right)^{-1}.$$

It is easy to see that if $\widehat{\boldsymbol{\Sigma}}_1^{-1} - \widehat{\boldsymbol{\Sigma}}_0^{-1}$ is a PSD matrix, then the matrix $\mathbf{C}$ needs to be PSD as well. Hence, we use the projection of $\mathbf{C}$ onto the cone of PSD matrices, i.e. $\widehat{\mathbf{C}} = \Pi_{\mathbb{S}^+}(\mathbf{C})$, to compute the squared distance between two examples $\mathbf{x}_i$ and $\mathbf{x}_j$ as follows

$$\begin{aligned}
d_{\mathbf{M}}^2(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i - \mathbf{x}_j)^\top\mathbf{X}\widehat{\mathbf{C}}\mathbf{X}^\top(\mathbf{x}_i - \mathbf{x}_j) \\
&= (\mathbf{k}_{\mathbf{x}_i} - \mathbf{k}_{\mathbf{x}_j})^\top\widehat{\mathbf{C}}(\mathbf{k}_{\mathbf{x}_i} - \mathbf{k}_{\mathbf{x}_j}),
\end{aligned}$$

where $\mathbf{k}_{\mathbf{x}} = \mathbf{X}^\top\mathbf{x}$. Clearly, the computations above only involve the inner products between examples. Therefore, we can easily replace the inner product by a kernel function to perform distance metric learning in the feature space $\mathcal{F}$. The great advantage is that the linear KISSME method is extended to nonlinear scenarios in a straightforward way through the use of kernel tricks. Although there exist general kernelization methods [38] for distance metric learning based on kernel principal component analysis (KPCA), by working directly in the feature space, our method leads to better and more robust results.

Another advantage of this kernelization is that it allows to apply KISSME on data sets containing structured objects on which kernel functions are defined. Since only a kernel function is required, many real-world data without an explicit vectorial representation (e.g., sequences, trees, and general graph-structured data) can be effectively dealt within our kernel-based framework. Several attempts have been made to design efficient kernel functions for such data. For instance, Leslie *et al.* [39] adopted the spectrum kernel on sequences for protein sequences. Collins and Duffy [40] showed how a kernel function can be applied to natural language structures. In [41], Gärtner *et al.* proposed kernels on labeled graphs with arbitrary structure. As the main focus of this paper is on person re-identification, interested readers may refer to the survey by Gärtner [42] for further details on defining kernel functions for structured data.

The overall computational complexity of k-KISSME mainly depends on the computation of the matrix $\mathbf{C}$. Due to the matrix multiplications and matrix inversions, this computation scales as $O(n^3)$. It is worth pointing out that k-KISSME has an advantage for problems where the number of features is significantly larger than the number of examples, i.e. $D \gg n$.

B. Incremental Settings

In person re-identification, a learning method should be less sensitive to appearance changes, such as varying lighting conditions, clothing, poses, and so on. It is desirable to formulate a computationally tractable distance metric learning framework in an incremental setting to address such dynamic behavior. However, to keep the Mahalanobis matrix being PSD, we always need to employ an eigenvalue decomposition, which is computationally expensive if this procedure has to be carried out upon the arrival of every new pairwise constraint. Therefore, instead of learning a Mahalanobis distance metric, we relax the positive semidefiniteness constraint and focus on learning a dissimilarity function. This relaxation strategy has already been adopted in various distance metric learning studies [15], [43]. We first define the dissimilarity function and then propose an efficient method for incrementally updating this dissimilarity function. As the new pairwise constraint can contain new examples, which are not observed in the training set, we also describe how to add these examples efficiently to the training set.

1) A Dissimilarity Function: In order to compute the dissimilarity of two examples in the feature space, it is necessary to redefine the covariance matrices. Since ϵ is a regularization constant, by redefining its value we can rewrite $\widehat{\boldsymbol{\Sigma}}_0$ and $\widehat{\boldsymbol{\Sigma}}_1$ in (4) as follows

$$\widehat{\boldsymbol{\Sigma}}_0 = \frac{1}{n_0}(\mathbf{X}\mathbf{H}_0\mathbf{X}^\top + \epsilon_0\mathbf{I}), \quad \widehat{\boldsymbol{\Sigma}}_1 = \frac{1}{n_1}(\mathbf{X}\mathbf{H}_1\mathbf{X}^\top + \epsilon_1\mathbf{I}),$$

where ϵ_0 and ϵ_1 are small positive constants. Applying Eq. (5) to compute the inverses of these covariance matrices, it yields

$$\widehat{\mathbf{\Sigma}}_0^{-1} = \frac{n_0}{\epsilon_0}\mathbf{I} - \frac{n_0}{\epsilon_0^2}\mathbf{X}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_0\right)^{-1}\mathbf{X}^\top,$$
$$\widehat{\mathbf{\Sigma}}_1^{-1} = \frac{n_1}{\epsilon_1}\mathbf{I} - \frac{n_1}{\epsilon_1^2}\mathbf{X}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{\epsilon_1}\mathbf{K}\mathbf{H}_1\right)^{-1}\mathbf{X}^\top.$$

Subsequently, the dissimilarity $\mathrm{dis}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ of two examples $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$\begin{aligned}
&\mathrm{dis}_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)\\
&= (\mathbf{x}_i - \mathbf{x}_j)^\top\left(\widehat{\mathbf{\Sigma}}_1^{-1} - \widehat{\mathbf{\Sigma}}_0^{-1}\right)(\mathbf{x}_i - \mathbf{x}_j)\\
&= (\mathbf{x}_i - \mathbf{x}_j)^\top\left[\frac{n_1}{\epsilon_1}\mathbf{I} - \frac{n_1}{\epsilon_1^2}\mathbf{X}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{\epsilon_1}\mathbf{K}\mathbf{H}_1\right)^{-1}\mathbf{X}^\top\right.\\
&\qquad\left. - \frac{n_0}{\epsilon_0}\mathbf{I} + \frac{n_0}{\epsilon_0^2}\mathbf{X}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_0\right)^{-1}\mathbf{X}^\top\right](\mathbf{x}_i - \mathbf{x}_j)\\
&= \left(\frac{n_1}{\epsilon_1} - \frac{n_0}{\epsilon_0}\right)(\mathbf{x}_i^\top\mathbf{x}_i - 2\mathbf{x}_i^\top\mathbf{x}_j + \mathbf{x}_j^\top\mathbf{x}_j)\\
&\quad + (\mathbf{k}_{\mathbf{x}_i} - \mathbf{k}_{\mathbf{x}_j})^\top\left[\frac{n_0}{\epsilon_0^2}\mathbf{H}_0\left(\mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_0\right)^{-1}\right.\\
&\qquad\left. - \frac{n_1}{\epsilon_1^2}\mathbf{H}_1\left(\mathbf{I} + \frac{1}{\epsilon_1}\mathbf{K}\mathbf{H}_1\right)^{-1}\right](\mathbf{k}_{\mathbf{x}_i} - \mathbf{k}_{\mathbf{x}_j}).
\end{aligned}$$

Note that $\mathrm{dis}_{\mathbf{M}}$ only depends on the inner products and, therefore, it can be learned in the feature space by applying the kernel trick. It is clear that the matrix $\widehat{\mathbf{\Sigma}}_1^{-1} - \widehat{\mathbf{\Sigma}}_0^{-1}$ obtained is not always PSD, consequently, the dissimilarity function $\mathrm{dis}_{\mathbf{M}}$ is not a pseudometric. However, our empirical experiments show that $\mathrm{dis}_{\mathbf{M}}$ obtains competitive results compared to $d_{\mathbf{M}}$, while being significantly faster to compute. Next, we show how to perform an efficient update for $\mathrm{dis}_{\mathbf{M}}$ upon the arrival of a new pairwise constraint.

2) Updating the Dissimilarity Function: Incremental learning usually arises in the case that images (examples) are sequentially collected, which is very common in a video surveillance system. An incremental learning system can be constructed, for instance, by adding additional cameras, or in a more general framework, by adding more knowledge from user interactions. It then follows that constraints are incrementally added using pairwise combinations of the new image and the images already in the training set. The following procedure only shows how the dissimilarity function is updated upon the arrival of a single constraint, but it is still possible to update efficiently given a set of constraints in a sequential manner. We consider the arrival of a new pairwise constraint $(\mathbf{x}_i, \mathbf{x}_j)$, which can be a must-link or a cannot-link constraint. Since $\mathrm{dis}_{\mathbf{M}}$ mainly depends on the inverses of the two covariance matrices $\widehat{\mathbf{\Sigma}}_0$ and $\widehat{\mathbf{\Sigma}}_1$, we need to perform an update for these inverses. We will assume that $(\mathbf{x}_i, \mathbf{x}_j)$ is a cannot-link constraint and discuss how to update the inverse of $\widehat{\mathbf{\Sigma}}_0$. The case of a must-link constraint $(\mathbf{x}_i, \mathbf{x}_j)$ can be treated in a similar way.

Let us assume that $\mathbf{x}_i$ and $\mathbf{x}_j$ are examples in the training set, hence, the input matrix $\mathbf{X}$ and the kernel matrix $\mathbf{K}$ remain the same, while the inverse of $\widehat{\mathbf{\Sigma}}_0$ becomes

$$\widehat{\mathbf{\Sigma}}_{\text{new}}^{-1} = \frac{n_0+1}{\epsilon_0}\mathbf{I} - \frac{n_0+1}{\epsilon_0^2}\mathbf{X}\mathbf{H}_{\text{new}}\left(\mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_{\text{new}}\right)^{-1}\mathbf{X}^\top, \tag{6}$$

where

$$\mathbf{H}_{\text{new}} = \mathbf{H}_0 + \mathbf{1}_i\mathbf{1}_i^\top - \mathbf{1}_i\mathbf{1}_j^\top - \mathbf{1}_j\mathbf{1}_i^\top + \mathbf{1}_j\mathbf{1}_j^\top. \tag{7}$$

One immediately observes that the inversion of $\widehat{\mathbf{\Sigma}}_{\text{new}}$ in Eq. (6) involves the computation of

$$\mathbf{T}_{\text{new}} = \mathbf{H}_{\text{new}}\left(\mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_{\text{new}}\right)^{-1}.$$

In order to compute $\mathbf{T}_{\text{new}}$ efficiently, we will perform the update for $\mathbf{H}_0$ in four steps instead of one as in Eq. (7). In each step, we add only a rank-one matrix to $\mathbf{H}_0$, while keeping track of the matrix $\mathbf{T}_{\text{new}}$. From Eq. (7), it is easy to see that $\mathbf{H}_0$ involves only four types of rank-one matrix update that are $\mathbf{1}_i\mathbf{1}_i^\top$, $-\mathbf{1}_i\mathbf{1}_j^\top$, $-\mathbf{1}_j\mathbf{1}_i^\top$, and $\mathbf{1}_j\mathbf{1}_j^\top$. By abuse of notation, we continue to write $\mathbf{H}_{\text{new}}$ to denote the matrix $\mathbf{H}_0$ after adding one of those rank-one matrices, i.e.

$$\mathbf{H}_{\text{new}} = \mathbf{H}_0 + \alpha\mathbf{1}_a\mathbf{1}_b^\top, \tag{8}$$

(one such update for all of the four steps)

where $\alpha \in \{-1, 1\}$ and $a, b \in \{i, j\}$. At each step, we also keep track of the matrices

$$\mathbf{Z}_0 = \mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_0, \quad \mathbf{T}_0 = \mathbf{H}_0\left(\mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_0\right)^{-1} = \mathbf{H}_0\mathbf{Z}_0^{-1},$$

and $\mathbf{Z}_0^{-1}$. The reason for doing so is to avoid extra computations by storing the previous computation results in each update. Next, we will show that

$$\mathbf{Z}_{\text{new}} = \mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_{\text{new}}, \quad \mathbf{T}_{\text{new}} = \mathbf{H}_{\text{new}}\mathbf{Z}_{\text{new}}^{-1}, \quad \text{and} \quad \mathbf{Z}_{\text{new}}^{-1}$$

can be computed with a complexity of $O(n^2)$ instead of $O(n^3)$ as the naive method. After each step, we set $\mathbf{Z}_0 = \mathbf{Z}_{\text{new}}$, $\mathbf{T}_0 = \mathbf{T}_{\text{new}}$, $\mathbf{H}_0 = \mathbf{H}_{\text{new}}$, and $\mathbf{Z}_0^{-1} = \mathbf{Z}_{\text{new}}^{-1}$ to perform the next step.

We now explain how to perform the update in one step. Substituting Eq. (8) into $\mathbf{Z}_{\text{new}}$ gives

$$\begin{aligned}
\mathbf{Z}_{\text{new}} &= \mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_{\text{new}} = \mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}(\mathbf{H}_0 + \alpha\mathbf{1}_a\mathbf{1}_b^\top)\\
&= \mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}\mathbf{H}_0 + \frac{\alpha}{\epsilon_0}\mathbf{K}_{.a}\mathbf{1}_b^\top = \mathbf{Z}_0 + \frac{\alpha}{\epsilon_0}\mathbf{K}_{.a}\mathbf{1}_b^\top.
\end{aligned}$$

The modification on $\mathbf{Z}_{\text{new}}$ involves only the computation of $\mathbf{K}_{.a}\mathbf{1}_b^\top$, which scales as $O(n^2)$. In order to compute $\mathbf{Z}_{\text{new}}^{-1}$, we consider the Sherman-Morrison formula [37] given by

$$(\mathbf{A} + \mathbf{c}\mathbf{d}^\top)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{c}\mathbf{d}^\top\mathbf{A}^{-1}}{1 + \mathbf{d}^\top\mathbf{A}^{-1}\mathbf{c}}.$$

(inverse)

Accordingly, it follows that

$$\mathbf{Z}_{\text{new}}^{-1} = \mathbf{Z}_0^{-1} - \frac{\alpha\mathbf{Z}_0^{-1}\mathbf{K}_{.a}\mathbf{1}_b^\top\mathbf{Z}_0^{-1}}{\epsilon_0 + \mathbf{1}_b^\top\mathbf{Z}_0^{-1}\alpha\mathbf{K}_{.a}} = \mathbf{Z}_0^{-1} - \mathbf{u}\mathbf{v}^\top,$$

where $\mathbf{u} = \alpha\mathbf{Z}_0^{-1}\mathbf{K}_{.a}/(\epsilon_0 + \mathbf{1}_b^\top\mathbf{Z}_0^{-1}\alpha\mathbf{K}_{.a})$ and $\mathbf{v}^\top = \mathbf{1}_b^\top\mathbf{Z}_0^{-1}$. Note that both vectors $\mathbf{u}$ and $\mathbf{v}$ are computed in $O(n^2)$,

therefore, the computation of $\mathbf{Z}_{\text{new}}^{-1}$ also scales as $O(n^2)$. Consequently, $\mathbf{T}_{\text{new}}$ can be computed in $O(n^2)$ as

$$\begin{aligned}\mathbf{T}_{\text{new}} &= \mathbf{H}_{\text{new}}\mathbf{Z}_{\text{new}}^{-1} = (\mathbf{H}_0 + \alpha\mathbf{1}_a\mathbf{1}_b^\top)(\mathbf{Z}_0^{-1} - \mathbf{u}\mathbf{v}^\top)\\ &= \mathbf{T}_0 - (\mathbf{H}_0\mathbf{u})\mathbf{v}^\top + \alpha(1 - \mathbf{1}_b^\top\mathbf{u})\mathbf{1}_a\mathbf{v}^\top.\end{aligned}$$

So far, we have assumed that $\mathbf{x}_i$ and $\mathbf{x}_j$ are examples in the training set. Of course, upon the arrival of a new pairwise constraint that is formed by new examples, we should also add these new examples to the training set before performing the above updates. The task now is to keep track of the matrices $\mathbf{Z}_0$, $\mathbf{T}_0$ and $\mathbf{Z}_0^{-1}$ efficiently. Next, we will explain how to perform this task after adding a new example to the training set in $O(n^2)$.

3) Adding a New Example: In the following, we will denote by $\mathbf{x}$ the newly arrived example. Without loss of generality, assuming that $\mathbf{x}$ will be added at the end of the training set, the input matrix $\mathbf{X}$ becomes $\mathbf{X}_{\text{new}} = (\mathbf{X}\ \mathbf{x})$. It follows that the kernel matrix $\mathbf{K}$ and the matrix $\mathbf{H}_0$ are changed to

$$\mathbf{K}_{\text{new}} = \begin{pmatrix}\mathbf{X}^\top\\ \mathbf{x}^\top\end{pmatrix}(\mathbf{X}\ \ \mathbf{x}) = \begin{pmatrix}\mathbf{K} & \mathbf{X}^\top\mathbf{x}\\ \mathbf{x}^\top\mathbf{X} & \mathbf{x}^\top\mathbf{x}\end{pmatrix},\quad \mathbf{H}_{\text{new}} = \begin{pmatrix}\mathbf{H}_0 & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix},$$

where $\mathbf{0}$ is a zero matrix with the appropriate dimensions. Note that $\mathbf{K}_{\text{new}}$ and $\mathbf{H}_{\text{new}}$ are computed in $O(n^2)$. Consequently, we should update the matrices $\mathbf{Z}_0$, $\mathbf{T}_0$, and $\mathbf{Z}_0^{-1}$ to make the update procedure in the previous subsection feasible.

We start by computing $\mathbf{Z}_{\text{new}}$ as follows

$$\begin{aligned}\mathbf{Z}_{\text{new}} &= \mathbf{I} + \frac{1}{\epsilon_0}\mathbf{K}_{\text{new}}\mathbf{H}_{\text{new}} = \mathbf{I} + \frac{1}{\epsilon_0}\begin{pmatrix}\mathbf{K} & \mathbf{X}^\top\mathbf{x}\\ \mathbf{x}^\top\mathbf{X} & \mathbf{x}^\top\mathbf{x}\end{pmatrix}\begin{pmatrix}\mathbf{H}_0 & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix}\\ &= \mathbf{I} + \frac{1}{\epsilon_0}\begin{pmatrix}\mathbf{K}\mathbf{H}_0 & \mathbf{0}\\ \mathbf{x}^\top\mathbf{X}\mathbf{H}_0 & 0\end{pmatrix} = \begin{pmatrix}\mathbf{Z}_0 & \mathbf{0}\\ \frac{1}{\epsilon_0}\mathbf{x}^\top\mathbf{X}\mathbf{H}_0 & 1\end{pmatrix}.\end{aligned}$$

The modification of $\mathbf{Z}_{\text{new}}$ depends only on the computation of $\mathbf{x}^\top\mathbf{X}\mathbf{H}_0$, which scales as $O(n^2)$. Applying block matrix inversion [37] given by

$$\begin{pmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{D} & \mathbf{E}\end{pmatrix}^{-1} = \begin{pmatrix}\mathbf{F}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{G}^{-1}\\ -\mathbf{G}^{-1}\mathbf{D}\mathbf{A}^{-1} & \mathbf{G}^{-1}\end{pmatrix},$$

where $\mathbf{F} = \mathbf{A} - \mathbf{B}\mathbf{E}^{-1}\mathbf{D}$ and $\mathbf{G} = \mathbf{E} - \mathbf{D}\mathbf{A}^{-1}\mathbf{B}$, we can compute $\mathbf{Z}_{\text{new}}^{-1}$ as follows

$$\mathbf{Z}_{\text{new}}^{-1} = \begin{pmatrix}\mathbf{Z}_0 & \mathbf{0}\\ \frac{1}{\epsilon_0}\mathbf{x}^\top\mathbf{X}\mathbf{H}_0 & 1\end{pmatrix}^{-1} = \begin{pmatrix}\mathbf{Z}_0^{-1} & \mathbf{0}\\ -\frac{1}{\epsilon_0}\mathbf{x}^\top\mathbf{X}\mathbf{H}_0\mathbf{Z}_0^{-1} & 1\end{pmatrix}.$$

This computation also scales as $O(n^2)$. Finally, $\mathbf{T}_{\text{new}}$ can be decomposed as

$$\begin{aligned}\mathbf{T}_{\text{new}} &= \mathbf{H}_{\text{new}}\mathbf{Z}_{\text{new}}^{-1} = \begin{pmatrix}\mathbf{H}_0 & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix}\begin{pmatrix}\mathbf{Z}_0^{-1} & \mathbf{0}\\ -\frac{1}{\epsilon_0}\mathbf{x}^\top\mathbf{X}\mathbf{H}_0\mathbf{Z}_0^{-1} & 1\end{pmatrix}\\ &= \begin{pmatrix}\mathbf{H}_0\mathbf{Z}_0^{-1} & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix} = \begin{pmatrix}\mathbf{T}_0 & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix}.\end{aligned}$$

Finally, we set $\mathbf{Z}_0 = \mathbf{Z}_{\text{new}}$, $\mathbf{T}_0 = \mathbf{T}_{\text{new}}$, $\mathbf{H}_0 = \mathbf{H}_{\text{new}}$, $\mathbf{Z}_0^{-1} = \mathbf{Z}_{\text{new}}^{-1}$, $\mathbf{X} = \mathbf{X}_{\text{new}}$, and $\mathbf{K} = \mathbf{K}_{\text{new}}$ to perform the next step.

4) Pseudocode: To summarize the whole procedure of incorporating a new pairwise constraint $(\mathbf{x}_i, \mathbf{x}_j)$, a pseudocode is given in Algorithm 1. We use $t \in \{0, 1\}$ to denote the type of the constraint $(\mathbf{x}_i, \mathbf{x}_j)$, i.e. $t = 0$ for a cannot-link constraint and $t = 1$ for a must-link constraint.

Algorithm 1 Incremental Update for k-KISSME

Input: A pairwise constraint $(\mathbf{x}_i, \mathbf{x}_j)$ of type $t \in \{0, 1\}$;
Output: The updated matrices $\mathbf{Z}_t$, $\mathbf{T}_t$, $\mathbf{Z}_t^{-1}$, $\mathbf{H}_t$, $\mathbf{X}$ and $\mathbf{K}$;
Begin
1) **For** $\mathbf{x} \leftarrow \{\mathbf{x}_i, \mathbf{x}_j\}$,
If $\mathbf{x} \notin \mathbf{X}$, ▷ adding a new example
- Insert $\mathbf{x}$ into $\mathbf{X}$, then update $\mathbf{K}$ and $\mathbf{H}$;
- Update $\mathbf{Z}_0$, $\mathbf{T}_0$, and $\mathbf{Z}_0^{-1}$ as in Subsection V-B.3;
- Update $\mathbf{Z}_1$, $\mathbf{T}_1$, and $\mathbf{Z}_1^{-1}$ as in Subsection V-B.3;

2) **For** $(\alpha, a, b) \leftarrow \{(1, i, i), (1, j, j), (-1, i, j), (-1, j, i)\}$
- Update $\mathbf{Z}_t$, $\mathbf{T}_t$, and $\mathbf{Z}_t^{-1}$ as in Subsection V-B.2;
- Update $\mathbf{H} \leftarrow \mathbf{H} + \alpha\mathbf{1}_a\mathbf{1}_b^\top$;

End

VI. Experiments

In this section, we evaluate the performance of our method on the task of identifying people for five publicly available data sets from real-world surveillance video. First, we describe the experimental settings. Then, we provide experimental results along with discussions.

A. Experimental Settings

1) Competing Distance Metric Learning Methods: We have implemented **k-KISSME**[1] in Matlab in order to compare its performance with other distance metric learning methods, including the information-theoretic metric learning (**ITML**) [27], the large-margin nearest neighbor (**LMNN**) [22], the original **KISSME** [16], and the cross-view quadratic discriminant analysis (**XQDA**) [3]. For k-KISSME, we apply the χ^2 kernel [44], given by

$$\mathcal{K}(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{D}\frac{2u_iv_i}{u_i + v_i}.$$

Following [3], the regularization parameter $\epsilon = 0.001$ is chosen. The constraints are extracted by forming all pairwise combinations of the training examples. As the number of cannot-link constraints can be significantly larger than that of must-link constraints, we use random subsampling to set the number of cannot-link constraints to ten times the number of must-link constraints to prevent very unbalanced problems.

2) Evaluation Protocol: We adopt a *single-shot* experimental setting as evaluation protocol. More specifically, we randomly select all images of p persons to form the test set and the rest to form the training set. Following the same experimental settings as used in [3], [16], and [32], we split each data set into two equal parts, one half for training and the other half for testing. Each test set contains a gallery set and a probe set. We randomly select an image for each person to form the gallery set and use the rest to form the probe set. In order to facilitate the comparison with previously published results, the average cumulative matching accuracies at rank 1, 5, 10 and 20 are reported over ten runs to evaluate the performance of a distance metric learning method.

[1] Source codes are available at http://users.ugent.be/~bacnguye/k-KISSME.v1.0.zip

TABLE I
A BRIEF DESCRIPTION OF THE DATA SETS USED IN OUR EXPERIMENTS

Data set	# individuals	# images	p
iLIDS	119	476	60
CAVIAR4REID	72	1,220	36
3DPeS	192	1,011	95
PRID450S	450	900	225
CUHK01	971	3,884	486

p: number persons for test set

3) Feature Representation: We use the local maximal occurrence representation (LOMO) recently proposed by Liao *et al.* [3] to employ feature extraction for all the distance metric learning methods. First, a multiscale Retinex transformation [45] is applied for image processing, resulting in a good representation of color and lightness. Then, LOMO applies the scale-invariant local ternary pattern method (SILTP) [46] to avoid intensity scale changes. Specifically, it locally constructs two scales of SILTP histograms and one HSV histogram of pixel features in a sliding window of size 10×10 to address viewpoint variations while maintaining local characteristics of a person. Finally, LOMO applies a log transform to normalize both HSV and SILTP features to unit length and obtains a 26,960-dimensional descriptor for each image. Due to the very high dimensionality, we project the extracted features into a 100-dimensional subspace using principal component analysis (PCA). In addition, we also report the performance of k-KISSME and XQDA based on the raw LOMO features because both of them can operate in a high-dimensional input space without reducing the dimensionality. Empirically, we have found that the results based on the raw LOMO features and its PCA subspace can be significantly different.

B. Experiments With Re-Identification Benchmark Data Sets

We conduct extensive experiments on five data sets, including **iLIDS** [47], **CAVIAR4REID** [48], **3DPeS** [49], **PRID450S** [50], and **CUHK01** [51], to validate the effectiveness of the proposed k-KISSME method. A brief description of these data sets is given in Table I. These data sets are widely used and provide many challenges in person re-identification, such as pose, viewpoint, background, resolution, and so on. We report the experimental results in two groups: (1) the performance comparison between k-KISSME and other distance metric learning methods using the low-dimensional features, (2) the performance of k-KISSME compared to other state-of-the-art methods. For the first group, we report the cumulative matching rates of all the competing distance metric learning methods based on the same 100-dimensional features using PCA. Additionally, the explicit feature map $\phi(\mathbf{u}) = \hat{\mathbf{u}}$, where $\hat{u}_i = \text{sign}(u_i)\sqrt{|u_i|}$, which resembles the Hellinger kernel embedding [44], is employed to turn KISSME into a baseline nonlinear method (PCA+Helli.+KISSME) in the feature space mapped by ϕ. For the second group, we report the performance of k-KISSME using the raw LOMO features against previously published results. A more detailed description and evaluation for each data set are described next.

TABLE II
THE TOP MATCHING RATES (%) ON THE iLIDS DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	Rank 1	Rank 5	Rank 10	Rank 20	Ref.
PCA+ITML	45.2	69.2	80.5	90.4	–
PCA+LMNN	43.5	66.6	77.8	88.1	–
PCA+KISSME	40.7	64.6	75.1	86.0	–
PCA+XQDA	42.2	65.7	77.2	89.2	–
PCA+Helli.+KISSME	40.8	64.7	75.5	86.2	–
PCA+k-KISSME	**48.3**	**70.3**	**80.9**	**90.7**	–
PCCA	23.0	51.1	67.0	83.3	[32]
LFDA	32.2	56.0	68.7	81.6	[32]
SVMML	20.8	49.1	65.4	81.7	[32]
rPCCA	26.6	54.3	69.7	84.5	[32]
kLFDA	36.5	64.1	76.5	88.5	[32]
MFA	32.6	58.5	71.5	84.4	[32]
DCNNs	**52.1**	68.2	78.0	88.8	[14]
LATENT-re-id	46.2	**70.2**	80.7	91.3	[20]
XQDA	43.5	69.9	81.8	93.3	–
k-KISSME	44.0	70.0	**82.6**	**93.4**	–

The iLIDS data set[2] contains 476 images of 119 pedestrians taken from two non-overlapping cameras at an airport. For each individual, the number of images varies from 2 to 8. All images are normalized to the same size of 128×48 pixels. Most of them contain several occlusions caused by luggage and people. We randomly choose images of 60 persons to form the test set, i.e. $p = 60$. The performances of k-KISSME and XQDA using the raw LOMO features against several state-of-the-art methods, including LATENT-re-id [20], PCCA [52], LFDA [53], SVMML [54], rPCCA [32], kLFDA [32], MFA [55], and DCNNs [14], are reported in Table II. As can be seen from the table, k-KISSME consistently outperforms the recent XQDA and other state-of-the-art methods. Even compared to the deep net proposed in [14], k-KISSME obtains a higher matching rate for rank 5, 10, and 20. Interestingly, k-KISSME achieves a significantly higher performance at rank 1 on PCA features.

The CAVIAR4REID data set[3] contains 1,220 images of 72 persons taken from two cameras at a shopping center in Lisbon. This data set is particularly designed with the aim of maximizing appearance variations in resolution changes, lighting conditions, and pose changes. There are 50 persons with both camera views and the remaining 22 persons with one camera view. The number of images for each individual varies from 10 to 20. Since the image sizes vary from 39×17 to 144×72 pixels, we normalize all images to the same size of 128×48 pixels in order to extract the same set of features as is done in [32]. Table III shows the cumulative matching accuracy with $p = 36$ for k-KISSME and XQDA using the raw LOMO features against several state-of-the-art methods, including PCCA [52], LFDA [53], SVMML [54], rPCCA [32], kLFDA [32], MFA [55], and RMLLC [56]. We can observe that k-KISSME obtains a competitive result compared to XQDA and outperforms other state-of-the-art methods.

[2]https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems

[3]http://www.lorisbazzani.info/caviar4reid.html

TABLE III

THE TOP MATCHING RATES (%) ON THE CAVIAR4REID DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	Rank 1	Rank 5	Rank 10	Rank 20	Ref.
PCA+ITML	25.3	53.6	71.7	89.6	–
PCA+LMNN	38.2	59.4	71.7	86.4	–
PCA+KISSME	44.3	72.6	85.7	96.6	–
PCA+XQDA	41.8	71.0	84.8	96.2	–
PCA+Helli.+KISSME	44.9	72.8	85.9	96.6	–
PCA+k-KISSME	**46.0**	**74.6**	**86.8**	**96.9**	**–**
PCCA	29.1	62.5	79.7	94.2	[32]
LFDA	31.7	56.1	70.4	86.9	[32]
SVMML	25.8	61.4	78.6	93.6	[32]
rPCCA	30.4	63.6	80.4	94.5	[32]
kLFDA	36.2	64.0	78.7	92.2	[32]
MFA	37.7	67.2	82.1	94.6	[32]
RMLLC	41.2	**73.5**	85.0	94.4	[56]
XQDA	**42.3**	71.8	**86.0**	96.0	–
k-KISSME	41.9	71.5	85.5	**96.1**	**–**

TABLE IV

THE TOP MATCHING RATES (%) ON THE 3DPeS DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	Rank 1	Rank 5	Rank 10	Rank 20	Ref.
PCA+ITML	26.8	51.3	64.9	79.7	–
PCA+LMNN	39.5	62.3	74.6	85.4	–
PCA+KISSME	**45.7**	69.7	79.1	88.2	–
PCA+XQDA	44.4	69.7	**80.0**	**89.4**	**–**
PCA+Helli.+KISSME	45.3	68.9	78.1	87.9	–
PCA+k-KISSME	**45.7**	**69.8**	79.3	88.2	–
PCCA	36.4	66.3	78.1	88.6	[32]
LFDA	39.1	61.7	71.8	82.6	[32]
SVMML	27.7	58.5	72.1	84.1	[32]
rPCCA	40.4	69.5	80.5	90.0	[32]
kLFDA	48.4	72.5	82.1	89.9	[32]
MFA	42.3	65.3	75.2	84.8	[32]
XQDA	46.7	70.7	81.2	91.0	–
k-KISSME	**48.7**	**72.6**	**83.9**	**92.1**	**–**

The 3DPeS data set[4] contains 1,011 images of 192 individuals captured from 8 different surveillance cameras. This data set is particularly designed for people tracking and person re-identification. The number of images for each individual varies from 2 to 26. Since the image sizes vary from 100×31 to 267×176 pixels, we normalize all images to the same size of 128×48 pixels. Table IV reports the cumulative matching accuracy with $p = 95$ for k-KISSME and XQDA using the raw LOMO features against other state-of-the-art methods, including PCCA [52], LFDA [53], SVMML [54], rPCCA [32], kLFDA [32], and MFA [55]. The results show that k-KISSME achieves the best overall performance. In particular, it achieves a recognition rate of 48.7% at rank 1.

The PRID450S data set[5] contains 900 images from 450 single-shot image pairs captured by two different surveillance cameras. It is a very challenging data set due to different viewpoint changes, background interference, and partial occlusion. In our experiment, each image is normalized to 128×48 pixels. Since the PRID450S is a newly constructed data set, there are only a few results reported in the literature. We show the cumulative matching rate with $p = 225$ of k-KISSME and XQDA using the raw LOMO features compared to some state-of-the-art methods, including EIML [57], SCNCN [23], ECM [58], and QRKISS [29], in Table V. Clearly, k-KISSME obtains the best performance on most of the reported ranks.

TABLE V

THE TOP MATCHING RATES (%) ON THE PRID450S DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	Rank 1	Rank 5	Rank 10	Rank 20	Ref.
PCA+ITML	30.6	60.4	73.2	85.3	–
PCA+LMNN	45.7	74.9	84.7	91.2	–
PCA+KISSME	41.6	71.3	81.1	89.4	–
PCA+XQDA	**48.7**	**77.7**	**86.1**	**93.2**	**–**
PCA+Helli.+KISSME	41.7	71.9	80.2	89.8	–
PCA+k-KISSME	47.4	76.4	85.0	91.9	–
EIML	35.0	–	68.0	77.0	[23]
SCNCD	41.6	68.9	79.4	87.8	[23]
ECM	41.9	66.3	76.9	84.2	[58]
QRKISS	**57.1**	80.7	88.0	–	[29]
XQDA	49.6	77.6	86.3	92.4	–
k-KISSME	53.9	**81.0**	**88.8**	**94.5**	**–**

TABLE VI

THE TOP MATCHING RATES (%) ON THE CUHK01 DATA SET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	Rank 1	Rank 5	Rank 10	Rank 20	Ref.
PCA+ITML	22.6	40.6	50.4	61.5	–
PCA+LMNN	42.3	61.5	70.5	79.2	–
PCA+KISSME	52.8	73.6	81.2	87.1	–
PCA+XQDA	50.6	72.4	80.5	87.3	–
PCA+Helli.+KISSME	51.9	73.8	80.9	87.6	
PCA+k-KISSME	**54.6**	**75.4**	**82.5**	**88.6**	**–**
kLFDA	32.8	59.0	69.6	79.2	[60]
IDLA	47.5	71.5	80.0	–	[12]
Ensembles	53.4	76.3	84.4	90.5	[2]
DeepRanking	50.4	75.9	84.1	91.3	[60]
ImpTrpLoss	53.7	**84.3**	**91.0**	**96.3**	[59]
XQDA	53.5	75.8	83.9	89.9	–
k-KISSME	**54.3**	74.2	80.5	87.0	–

The CUHK01 data set[6] contains 3,884 images of 971 pedestrians captured from two disjoint cameras on a college campus. Each camera has taken two images of every individual. In our experiment, all images are downsized to a resolution of 128×48 pixels to reduce the computation time. We set $p = 486$ in order to facilitate the comparison with other methods. Table VI shows the cumulative matching accuracy of k-KISSME using the raw LOMO features against some state-of-the-art methods, including kLFDA [32], Ensembles [2], IDLA [12], ImpTrpLoss [59], and DeepRanking [60]. Clearly, k-KISSME achieves the best matching performance among the competing distance metric learning methods on the PCA features. Despite its simplicity, k-KISSME obtains a better performance than deep learning methods at rank 1, while being less accurate at rank 5, 10, and 20. This result is not surprising

[4]http://imagelab.ing.unimore.it/visor/3dpes.asp

[5]https://www.tugraz.at/institute/icg/research/team-bischof/lrs/downloads/prid450s/

[6]http://www.ee.cuhk.edu.hk/~xgwang/CUHK_identification.html

TABLE VII

AVERAGE TRAINING TIME (IN SECONDS) OF THE COMPETING DISTANCE METRIC LEARNING METHODS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLDFACE

Method	iLIDS	CAVIAR4REID	3DPeS	PRID450S	CUHK01
ITML	78.96	76.03	57.63	63.33	38.68
LMNN	61.23	209.54	231.41	267.51	2,908.68
KISSME	0.09	0.54	0.24	0.06	1.03
XQDA	**0.03**	**0.02**	**0.08**	**0.05**	**0.33**
k-KISSME	1.49	22.65	7.23	1.82	51.17

since deep learning methods use advanced techniques such as data augmentation [59] and additional training data [60] to improve the matching rate as well as to avoid overfitting.

According to the overall results, we can see that the kernelized version of KISSME leads to significant improvements over the original KISSME. Our method k-KISSME yields the best performance on most data sets, demonstrating a great flexibility and accuracy for matching compared to other competing methods. It is also interesting to note that k-KISSME consistently obtains high rank r matching rates with small values of r. This provides important information for a person re-identification system because the top matched images are usually verified by a human operator [61]. We also note that k-KISSME outperforms most of the linear methods on the PCA features. The reason for this may lie in the fact that the projection made by PCA may intertwine the useful features and the noisy features. Consequently, the data set may be transformed into a nonlinearly separable problem, thus making it difficult for linear methods. As commonly used in computer vision, an explicit feature map, such as the signed square root, can approximate nonlinear distance metric learning methods in the feature space by linear ones. Although this approach is scalable for large data sets, the improvement is still very limited (see the results of PCA+KISSME and PCA+Helli.+KISSME). In contrast, k-KISSME employs the kernel trick to find a good solution in the implicit feature space, making it more robust for complex tasks.

C. Running Time

The average training times of the competing distance metric learning methods on the low-dimensional as well as the raw LOMO features are shown in Table VII. The running time is computed on a laptop with 4 Intel Core i5-5200U CPUs (2.20GHz) and 8GB RAM. Note that the results include the time for computing the kernel matrix. As can be seen from the table, XQDA is the least time consuming on all these data sets, followed by KISSME. It should be noticed that k-KISSME is significantly faster than other iterative methods such as ITML and LMNN on small-sized data sets. The slower speed on large-sized data sets of k-KISSME is a result of computing the kernel matrix. Further running time improvements can be anticipated by using advanced techniques to speed up the calculation of the kernel matrix. Although our method has mainly been implemented in MATLAB, a careful implementation can significantly improve the real computation time. More importantly, k-KISSME can perform efficiently on very high-dimensional data sets, which could be computationally challenging for those methods that directly learn a distance metric from the input space. It requires significantly less memory and a lower training time compared to the deep neural networks. Moreover, k-KISSME is very simple to implement, computationally efficient, and serves our main goal, which is to develop an efficient system for person re-identification.

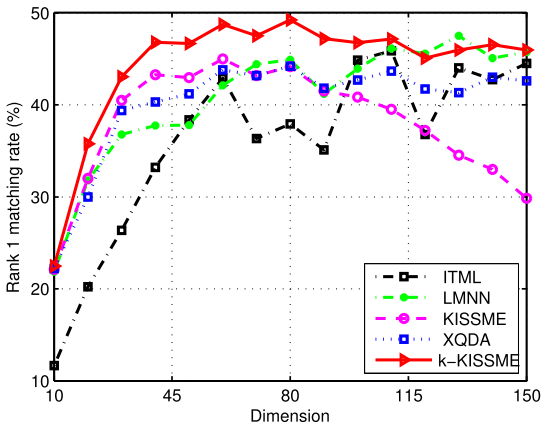

Fig. 2. Illustration of rank 1 matching rate vs. number of dimensions on the iLIDS data set.

D. Experiments With Dimensionality

In this subsection, we investigate how the performance of distance metric learning methods varies with different subspace dimensions. For this purpose, we report the matching rate at rank 1 for ITML, LMNN, KISSME, XQDA, and k-KISSME on the iLIDS data set with different dimensions extracted by PCA (see Fig. 2). We keep the same experimental settings for all the competing methods. As can be seen from this figure, k-KISSME consistently outperforms other methods over all the reported dimensions. We found that KISSME is very sensitive to the choice of the number of PCA dimensions, yielding a relatively high variance over different dimensions. This behavior was also noted by Xiong *et al.* [32]. Nevertheless, we observe that XQDA and k-KISSME tend to have a more stable performance over the different dimensions. The latter can be easily explained by the fact that both XQDA and k-KISSME add a small regularizer to the diagonal elements of the covariance matrices, making the estimation more smooth and robust, especially when the dimensionality is increased.

E. Experiments With Incremental Learning

We further verify the efficiency and effectiveness of using the incremental update procedure described in Subsection V-B for k-KISSME. As an illustration, we compare k-KISSME and the method that learns a dissimilarity function (denoted by **k-KISSME (inc)**) in terms of training time and rank 1 matching rate on the CAVIAR4REID and 3DPeS data sets (see Fig. 3). The same experimental settings are used. We keep on randomly adding a pairwise constraint on each update. As we pointed out in Subsection V-B, ϵ_0 and ϵ_1 act as hyperparameters that can be used to adjust the regularizing

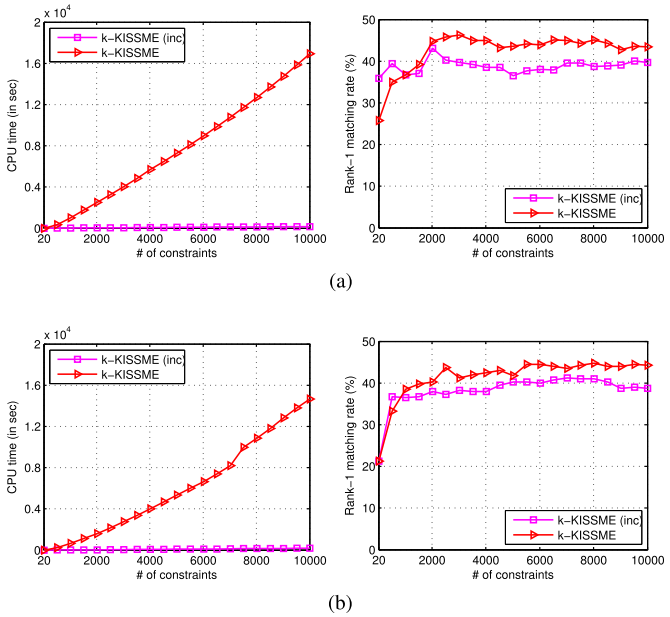

Fig. 3. Illustration of the incremental update procedure on the CAVIAR4REID and 3DPeS data sets (left) training time (in seconds) vs. number of constraints, (right) rank 1 matching rate vs. number of constraints. (a) CAVIAR4REID. (b) 3DPeS.

terms of the covariance matrices. For the CAVIAR4REID and 3DPeS data sets, we set the values of ϵ_0 and ϵ_1 to 1 and 0.05, respectively, which yield the best results in most of our experiments. Clearly, these hyperparameters should be determined by the characteristics of the data sets as well as the pairwise constraints.

As expected, using the incremental update procedure yields a significant speedup, while obtaining a competitive performance. Like many online learning algorithms, fluctuations in performance are mainly due to the randomness of adding constraints. However, this incremental technique ensures that the similarity function is immediately trained and will become more accurate over time as more new points and pairwise constraints are added. The advantage of incremental k-KISSME is particularly apparent when there is an unbounded stream of possible constraints to learn from.

VII. Conclusion and Future Work

Person re-identification is a challenging problem in video surveillance due to the large variations in appearance by using different cameras. To deal with this challenge, we have proposed a distance metric learning method, named k-KISSME, by incorporating kernels into the KISSME method. This allows k-KISSME to operate in a nonlinear feature space induced by a kernel function. As a result, k-KISSME improves the recognition rate and could be applied in learning a distance metric from structural objects without having a vectorial representation. Moreover, we have also introduced a fast version for k-KISSME avoiding expensive recomputations in an incremental setting. Experiments on five real-world data sets have demonstrated the effectiveness of k-KISSME compared to other distance metric learning methods for person re-identification tasks.

Despite the promising results, there are still some aspects of k-KISSME and its incremental version that require further efforts. For instance, the computational bottleneck of k-KISSME becomes impractical on large-scale data sets. The latter is endemic to most kernel-based methods and reducing the training set size may be useful in this case. While our incremental update strategy for k-KISSME may be initially sufficient, it would be more interesting to be able to keep the Mahalanobis matrix within the cone of PSD matrices and to perform an update on the arrival of multiple constraints at the same time.

References

[1] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image Vis. Comput.*, vol. 32, no. 4, pp. 270–286, 2014.
[2] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1846–1855.
[3] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2197–2206.
[4] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang, "Person re-identification by support vector ranking," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 1–11.
[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[6] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 262–275.
[7] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[8] B. Ma, Y. Su, and F. Jurie, "Local descriptors encoded by fisher vectors for person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 413–422.
[9] G. Lisanti, I. Masi, A. D. Bagdanov, and A. Del Bimbo, "Person re-identification by iterative re-weighted sparse ranking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1629–1642, Aug. 2015.
[10] D. Tao, Y. Guo, Y. Li, and X. Gao, "Tensor rank preserving discriminant analysis for facial recognition," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 325–334, Jan. 2018.
[11] D. Tao, J. Cheng, M. Song, and X. Lin, "Manifold ranking-based matrix factorization for saliency detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1122–1134, Jun. 2016.
[12] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3908–3916.
[13] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1249–1258.
[14] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, Oct. 2015.
[15] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 780–793.
[16] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2288–2295.
[17] B. Nguyen, C. Morell, and B. De Baets, "Supervised distance metric learning through maximization of the Jeffrey divergence," *Pattern Recognit.*, vol. 64, pp. 215–225, Apr. 2017.
[18] Y. Yang, S. Liao, Z. Lei, and S. Z. Li, "Large scale similarity learning using similar pairs for person verification," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3655–3661.
[19] R. Zhao, W. Ouyang, and X. Wang, "Person re-identification by saliency learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 356–370, Feb. 2017.
[20] C. Sun, D. Wang, and H. Lu, "Person re-identification via distance metric learning with latent variables," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 23–34, Jan. 2017.

[21] C. Shen, J. Kim, L. Wang, and A. van den Hengel, "Positive semidefinite metric learning using boosting-like algorithms," *J. Mach. Learn. Res.*, vol. 13, pp. 1007–1036, Apr. 2012.
[22] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
[23] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient color names for person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 536–551.
[24] D. Tao, Y. Guo, M. Song, Y. Li, Z. Yu, and Y. Y. Tang, "Person re-identification by dual-regularized kiss metric learning," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2726–2738, Jun. 2016.
[25] D. Tao, L. Jin, Y. Wang, and X. Li, "Person reidentification by minimum classification error-based KISS metric learning," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 242–252, Feb. 2015.
[26] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *Proc. 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 498–505.
[27] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 209–216.
[28] D. Tao, Y. Guo, B. Yu, J. Pang, and Z. Yu, "Deep multi-view feature learning for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[29] C. Zhao, Y. Chen, Z. Wei, D. Miao, and X. Gu, "QRKISS: A two-stage metric learning via QR-decomposition and KISS for person re-identification," *Neural Process. Lett.*, pp. 1–24, Mar. 2018.
[30] M. Dikmen, E. Akbas, T. S. Huang, and N. Ahuja, "Pedestrian recognition with a learned metric," in *Proc. Asian Conf. Comput. Vis.*, 2011, pp. 501–512.
[31] W.-S. Zheng, S. Gong, and T. Xiang, "Reidentification by relative distance comparison," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 653–668, Mar. 2013.
[32] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 1–16.
[33] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 519–547, Jan. 2012.
[34] B. Moghaddam, T. Jebara, and A. Petland, "Bayesian face recognition," *Pattern Recognit.*, vol. 33, no. 11, pp. 1771–1782, 2000.
[35] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 937–965, 2006.
[36] J. H. Friedman, "Regularized discriminant analysis," *J. Amer. Statist. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.
[37] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep., 2012. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274
[38] C. Zhang, F. Nie, and S. Xiang, "A general kernelization framework for learning algorithms based on kernel PCA," *Neurocomputing*, vol. 73, nos. 4–6, pp. 959–967, 2010.
[39] C. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification," in *Proc. Pacific Symp. Biocomput.*, 2002, pp. 564–575.
[40] M. Collins and N. Duffy, "Convolution kernels for natural language," in *Proc. 14th Adv. Neural Inf. Process. Syst.*, 2002, pp. 625–632.
[41] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Proc. 16th Annu. Conf. Comput. Learn. Theory 7th Kernel Workshop*, 2003, pp. 129–143.
[42] T. Gärtner, "A survey of kernels for structured data," *ACM SIGKDD Explorations Newslett.*, vol. 5, no. 1, pp. 49–58, 2003.
[43] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, Jan. 2010.
[44] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, Mar. 2012.
[45] D. J. Jobson, Z.-U. Rahman, and G. A. Woodell, "Properties and performance of a center/surround Retinex," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 451–462, Mar. 1997.
[46] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1301–1306.
[47] W.-S. Zheng, S. Gong, and T. Xiang, "Associating groups of people," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1, 2009, pp. 1–11.
[48] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 68.1–68.11.
[49] D. Baltieri, R. Vezzani, and R. Cucchiara, "3DPeS: 3D people dataset for surveillance and forensics," in *Proc. Joint ACM Workshop Human Gesture Behav. Understand.*, 2011, pp. 59–64.
[50] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis distance learning for person re-identification," in *Proc. Adv. Comput. Vis. Pattern Recognit.*, 2014, pp. 247–267.
[51] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Proc. Asian Conf. Comput. Vis.*, 2013, pp. 31–44.
[52] A. Mignon and F. Jurie, "PCCA: A new approach for distance learning from sparse pairwise constraints," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2666–2672.
[53] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local Fisher discriminant analysis for pedestrian re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3318–3325.
[54] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith, "Learning locally-adaptive decision functions for person verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3610–3617.
[55] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.
[56] J. Chen, Z. Zhang, and Y. Wang, "Relevance metric learning for person re-identification by exploiting listwise similarities," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4741–4755, Dec. 2015.
[57] M. Hirzer, P. M. Roth, and H. Bischof, "Person re-identification by efficient impostor-based metric learning," in *Proc. 9th Int. Conf. Adv. Video Signal-Based Surveill.*, Sep. 2012, pp. 203–208.
[58] X. Liu, H. Wang, Y. Wu, J. Yang, and M.-H. Yang, "An ensemble color model for human re-identification," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 868–875.
[59] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1335–1344.
[60] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2353–2367, May 2016.
[61] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill.*, Oct. 2007, pp. 41–48.

Bac Nguyen received the B.Sc. and M.Sc. *(summa cum laude)* degrees in computer science from the Universidad Central de Las Villas in 2014 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Department of Data Analysis and Mathematical Modeling, Ghent University, Belgium. His research interests are in the areas of data mining, machine learning, and their applications.

Bernard De Baets received the M.Sc. degree *(summa cum laude)* in maths, the master's degree *(summa cum laude)* in knowledge technology, and the Ph.D. degree *(summa cum laude)* in maths from Ghent University, Belgium, in 1988, 1991, and 1995, respectively. Since 1999, he has been a Senior Full Professor in applied maths with Ghent University, where he is leading KERMIT, the research unit Knowledge-Based Systems. He was an Honorary Professor of Budapest Tech in 2006. He was an Honorary Doctor of the University of Turku in 2017 and a Profesor Invitado of the Universidad Central "Marta Abreu" de Las Villas, Cuba. His publications comprise over 450 papers in international journals and about 60 book chapters. In 2011, he was an IFSA Fellow. He was a recipient of the Government of Canada Award (1988). He serves on the Editorial Boards of various international journals, in particular as the Co-Editor-in-Chief of *Fuzzy Sets and Systems*.