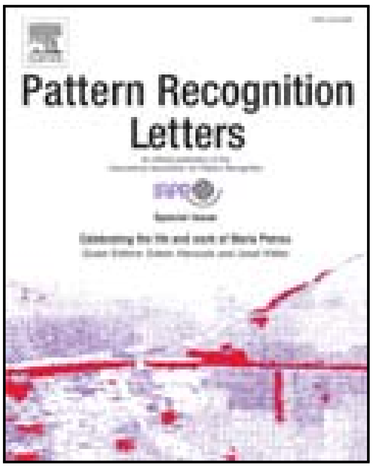
Pattern Recognition Letters

Highlights

- Kernel versions of the keep it simple and straightforward metric learning method
- Mathematical formulation based on infinite dimensional covariance matrices for the kernel methods
- A closed-form solution to project on the positive cone in a reproducing kernel Hilbert space
- Accurate Riemannian optimization method for the projection
- Nystrom method to approximate a reproducing kernel Hilbert space before learning a metric

Pattern Recognition Letters
journal homepage: www.elsevier.com

No Fuss Metric Learning, a Hilbert Space Scenario

Masoud Faraki[a,b,**], Mehrtash T. Harandi[a,b], Fatih Porikli[a]

[a]*Research School of Engineering, Australian National University, Canberra, Australia*
[b]*Data61-CSIRO, Acton building, Canberra, Australia*

ABSTRACT

In this paper, we devise a kernel version of the recently introduced keep it simple and straightforward metric learning method, hence adding a novel dimension to its applicability in scenarios where input data is non-linearly distributed. To this end, we make use of the infinite dimensional covariance matrices and show how a matrix in a reproducing kernel Hilbert space can be projected onto the positive cone efficiently. In particular, we propose two techniques towards projecting on the positive cone in a reproducing kernel Hilbert space. The first method, though approximating the solution, enjoys a closed-form and analytic formulation. The second solution is more accurate and requires Riemannian optimization techniques. Nevertheless, both solutions can scale up very well as our empirical evaluations suggest. For the sake of completeness, we also employ the Nyström method to approximate a reproducing kernel Hilbert space before learning a metric. Our experiments evidence that, compared to the state-of-the-art metric learning algorithms, working directly in reproducing kernel Hilbert space, leads to more robust and better performances.

1. Introduction

Unlike many other metric learning techniques such as Large Margin Nearest Neighbor (LMNN) Weinberger and Saul (2009) and Information-Theoretic Metric Learning (ITML) Davis et al. (2007), the recently introduced method of "Keep It Simple and Straightforward MEtric" (KISSME) Koestinger et al. (2012) avoids hefty optimization routines, which makes it very attractive, if not the first or only choice, in many cases. On the downside, the KISSME algorithm is designed to work with explicit and vectorized data. As such, the algorithm is unable to learn efficiently from non-linear data or if data is not in vector form (e.g., manifold-value data). In this paper, we provide solutions to both limitations in a principal way and present techniques to kernelize KISSME, making it applicable to a wider set of problems.

The commonly used Euclidean distance assumes that all features are of equal importance, which is almost never the case in practice. In computer vision, determining a suitable metric plays a pivotal role in various applications such as person reidentification Xiong et al. (2014); Chen et al. (2015); Zheng et al. (2015); Cheng et al. (2011), face and kinship verification Koestinger et al. (2012); Li et al. (2013); Lu et al. (2014); Guillaumin et al. (2009); Wolf et al. (2011), and image retrieval Song et al. (2016); Hoi et al. (2006), to name a few. In the literature, the most common practice is to learn a Mahalanobis distance which ultimately boils down to learning a Symmetric Positive Definite (SPD) matrix from the given data Harandi et al. (2017). While significant progress has been made over the years, optimization techniques involving SPD matrices are notoriously slow and do not scale well if the dimensionality of the data increases. The beauty of the KISSME algorithm comes from the fact that the Mahalanobis distance is learned by one sweep over the data with the dominant computation being an eigenvalue decomposition. However, and as evidenced by some recent studies (e.g., Xiong et al. (2014)), non-linearity associated with high-dimensional data cannot be captured by the KISSME algorithm, making the algorithm fall short compared to the methods that are efficiently benefiting from such information.

Contributions: To kernelize KISSME algorithm while preserving its unique features, we make use of the recently introduced infinite dimensional covariance matrices Harandi et al. (2014); Quang et al. (2014); Faraki et al. (2015) and show how

[**] Corresponding author: Tel.: +61-2-6267-6200;
e-mail: masoud.faraki@data61.csiro.au (Masoud Faraki)

a matrix in a Reproducing Kernel Hilbert Space (RKHS) can be projected onto the positive cone efficiently. In particular, we propose two techniques towards projecting onto the positive cone in an RKHS. The first method, albeit approximating the solution, enjoys a closed-form and analytic formulation. The second solution is more accurate and requires Riemannian optimization techniques. Nevertheless, both solutions can scale up very well as our empirical evaluations suggest. Furthermore, to have the full package, we employ the Nyström method Baker (1977) to approximate an RKHS and formulate the Nyström KISSME accordingly.

In our experiments, we demonstrate the benefits of the presented kernelized KISSME approach over existing metric learning schemes on the task of person reidentification using the iLIDS Zheng et al. (2009) and the CAVIAR Cheng et al. (2011) datasets and kinship verification from unconstrained face images using the KinFace-I and the KinFace-II datasets Lu et al. (2014).

Before concluding this part, we emphasize that our method learns a metric purely from the equivalence constraints (similar/dissimilar pairs) and does not use class-labels as required by some other learning techniques (e.g., Song et al. (2016); Ding et al. (2015); Xiong et al. (2014)).

2. Related Work

Very relevant to our work is the "Keep It Simple and Straightforward MEtric" (KISSME) Koestinger et al. (2012) algorithm that addresses large-scale problems. We will discuss KISSME in detail in §3 but before that we review some notable examples of metric learning techniques below.

A goal common to the state-of-the-art metric learning techniques is to make use of discriminative information existing in training data. Neighborhood Component Analysis (NCA) Goldberger et al. (2004) learns a Mahalanobis distance to improve k-Nearest Neighbor (kNN) classification score in a supervised manner. To this end, NCA minimizes the expected value of a stochastic variant of the kNN error. The classification model is parameter free, without any assumptions about the shape of the class distributions or the boundaries between them, which makes NCA attractive and easy to use.

Large Margin Nearest Neighbor (LMNN), learns a global linear transformation of labeled input data to improve the kNN classification accuracy Weinberger and Saul (2009). In doing so, the learned transformation (or equivalently the metric) is deemed to unite the k-nearest neighbors of each point sharing the same label while separating instances from different classes by a margin. Learning the linear transformation is formulated as a semi-definite programming problem and solved by iterating between a gradient descent step followed by projecting the solution onto the positive semi-definite cone.

Davis et al., leverage on the connection between the multivariate Gaussian distributions and the Mahalanobis metrics in their Information-Theoretic Metric Learning (ITML) method Davis et al. (2007). The method seeks a metric to enforce the distance between similar pairs to be below the threshold δ_l while making the distance between dissimilar pairs exceeding the threshold δ_u with $\delta_l < \delta_u$. In ITML, the proximity between two Mahalanobis metrics is measured by the Kullback-Leibler divergence of their corresponding distributions.

Guillaumin et al. Guillaumin et al. (2009) propose Logistic Discriminant Metric Learning (LDML) to tackle the problem of face verification. The key idea is to find a metric to make the distances between similar pairs smaller than the distances between dissimilar pairs. Thereby, a probabilistic estimate depicting whether a pair of face images belong to the same person or not is obtained using the Mahalanobis distance along a linear logistic discriminant model. The Mahalanobis metric is obtained by maximizing the log-likelihood of the logistic model.

In recent years, deep metric learning has received growing attention, following the trend of deep Convolutional Neural Networks (CNN) in solving large-scale classification problems Krizhevsky et al. (2012).

Metric Learning and Deep Nets

Similarity and metric learning using deep nets can be traced back to the advent of Siamese networks Chopra et al. (2005). Exploiting the objective of successful metric learning algorithms in deep nets is a major trend nowadays Wolf et al. (2011); Sun et al. (2014). For example, in the spirit of LDML, a two layer discriminative network for face verification is proposed in Hu et al. (2014). Mimicking the learning strategy of LMNN is studied in Ding et al. (2015) where triplets are also considered during training. Song et al. Song et al. (2016) discuss drawbacks of pairwise constraints and triplets when combined with stochastic gradient descent updates in deep nets. In short, given the small size of batches, the full potential of pairwise or triplet information cannot be exploited in deep nets. As suggested in Song et al. (2016), careful construction of batches by the concept of lifted structured feature embedding, i.e., including hard triplets during training, leads to significant improvement in accuracy.

3. Background

Throughout the paper, we use bold lower-case letters (e.g., $\boldsymbol{x}$) to denote vectors and bold upper-case letters (e.g., $\boldsymbol{X}$) to show matrices. $\mathbf{I}_n$ is the $n \times n$ identity matrix. The Frobenius norm of a matrix is $\|\boldsymbol{X}\|_F = \sqrt{\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X})}$, where $\mathrm{Tr}(\cdot)$ indicates the matrix trace. $\mathcal{S}^n_{++}$ is the space of $n \times n$ Symmetric Positive Definite (SPD) matrices.

Let $\mathcal{X}$ be a set. A distance or metric over $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ that satisfies the following axioms $\forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$

1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity),
2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$ (distinguishability),
3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry),
4. $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ (triangle inequality).

Choosing $\mathcal{X}$ to be the d-dimensional Euclidean space, the class of Mahalanobis distances can be defined as

$$d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y})}\,, \tag{1}$$

with $\boldsymbol{M} \in \mathcal{S}^d_{++}$.

The goal of Mahalanobis Metric Learning (MML) is to determine $\boldsymbol{M}$ such that $d_{\boldsymbol{M}}(\cdot,\cdot)$ endows certain useful properties. For this purpose, the MML algorithm accepts a set of training data in the form $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, l_i)\}_{i=1}^{n}$ with $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathbb{R}^d$ and $l_i \in \{0, 1\}$ to determine $\boldsymbol{M}$. Here, l_i indicates the similarity label of the pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, i.e., $l_i = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ come from the same class and $l_i = 0$ otherwise[1].

In KISSME algorithm, which our work is built upon, a dissimilarity hypothesis is defined as

$$\epsilon(\boldsymbol{x}_i, \boldsymbol{y}_i) = log\left(\frac{\frac{1}{\sqrt{2\pi|\Sigma_d|}}\exp\Big(-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{y}_i)^T\Sigma_d^{-1}(\boldsymbol{x}_i-\boldsymbol{y}_i)\Big)}{\frac{1}{\sqrt{2\pi|\Sigma_s|}}\exp\Big(-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{y}_i)^T\Sigma_s^{-1}(\boldsymbol{x}_i-\boldsymbol{y}_i)\Big)}\right), \quad (2)$$

where

$$\Sigma_d = \frac{1}{\#(l_i=0)}\sum_{i,l_i=0}(\boldsymbol{x}_i-\boldsymbol{y}_i)(\boldsymbol{x}_i-\boldsymbol{y}_i)^T,$$
$$\Sigma_s = \frac{1}{\#(l_i=1)}\sum_{i,l_i=1}(\boldsymbol{x}_i-\boldsymbol{y}_i)(\boldsymbol{x}_i-\boldsymbol{y}_i)^T\,. \quad (3)$$

where # denotes the number of samples.

Having a large $\epsilon(\boldsymbol{x}_i, \boldsymbol{y}_i)$ indicates that $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are dissimilar, and vice-versa. With this hypothesis, the Mahalanobis matrix is obtained as $\boldsymbol{M} = \mathrm{Proj}(\Sigma_s^{-1} - \Sigma_d^{-1})$ with $\mathrm{Proj}(\cdot)$ denoting projection to the cone of positive definite matrices. Such a projection is required to have a valid distance. In KISSME, the projection is obtained by clipping the spectrum of $\Sigma_s^{-1} - \Sigma_d^{-1}$. That is given the eigen-decomposition of $\Sigma_s^{-1} - \Sigma_d^{-1}$ as $\boldsymbol{UDU}^T$ then $\boldsymbol{M} = \boldsymbol{UD}_+\boldsymbol{U}^T$ where $\boldsymbol{D}_+ = \mathrm{diag}(\max(d_i, \varepsilon))$ with $\boldsymbol{D} = \mathrm{diag}(d_i)$ and ε being a very small positive number.

4. Our Approach

Let $\mathcal{X}$ and $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be a set and a positive definite (pd) kernel defined on $\mathcal{X}$, respectively. According to the Mercer theorem, a mapping $\phi : \mathcal{X} \to \mathcal{H}$ to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ exists for any pd kernel. Our aim in this section is to derive a Mahalanobis distance $d_{\mathcal{H}} : \mathcal{H}\times\mathcal{H} \to \mathbb{R}_+$ in the feature space $\mathcal{H}$ with certain properties.

Suppose $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, l_i)\}_{i=1}^{n}$ with $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathcal{X}$ and $l_i \in \{0, 1\}$ be a set of n training samples. Given a pd kernel $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ the Mahalanobis distance in $\mathcal{H}$ can be written as

$$d_{\mathcal{H}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sqrt{(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i))^T \boldsymbol{M}_{\mathcal{H}}(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i))}\,. \quad (4)$$

To learn $\boldsymbol{M}_{\mathcal{H}}$, we define the likelihood ratio test of the pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ as

$$\epsilon_{\mathcal{H}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \quad (5)$$
$$log\left(\frac{\frac{1}{\sqrt{2\pi|\Sigma_{\mathcal{H},d}|}}\exp\Big(-\frac{1}{2}(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))^T\Sigma_{\mathcal{H},d}^{-1}(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))\Big)}{\frac{1}{\sqrt{2\pi|\Sigma_{\mathcal{H},s}|}}\exp\Big(-\frac{1}{2}(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))^T\Sigma_{\mathcal{H},s}^{-1}(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))\Big)}\right).$$

Here, the covariance matrices are

$$\Sigma_{\mathcal{H},d} = \frac{1}{\#(l_i=0)}\sum_{i,l_i=0}(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))^T,$$
$$\Sigma_{\mathcal{H},s} = \frac{1}{\#(l_i=1)}\sum_{i,l_i=1}(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))(\phi(\boldsymbol{x}_i)-\phi(\boldsymbol{y}_i))^T\,. \quad (6)$$

With the same line of reasoning as Koestinger et al. (2012), the Mahalanobis form that maximizes $\epsilon_{\mathcal{H}}(\cdot,\cdot)$ over the training samples is obtained by choosing $\boldsymbol{M}_{\mathcal{H}} = \mathrm{Proj}_{\mathcal{H}}(\Sigma_{\mathcal{H},s}^{-1} - \Sigma_{\mathcal{H},d}^{-1})$. As such, we need to answer the following questions to extend the KISSME algorithm to work in $\mathcal{H}$:

1. *How $\Sigma_{\mathcal{H},s}^{-1}$ and $\Sigma_{\mathcal{H},d}^{-1}$ can be obtained in $\mathcal{H}$?*
2. *How the projection* $\mathrm{Proj}_{\mathcal{H}}(\cdot)$ *can be defined efficiently in $\mathcal{H}$?*
3. *Having answers to the previous questions at our disposal, how $d_{\mathcal{H}}(\cdot,\cdot)$ can be obtained efficiently $\mathcal{H}$?*

Below, we address these questions one-by-one.

4.1. Obtaining $\Sigma_{\mathcal{H},s}^{-1}$ and $\Sigma_{\mathcal{H},d}^{-1}$

In essence, obtaining $\Sigma_{\mathcal{H},s}^{-1}$ and $\Sigma_{\mathcal{H},d}^{-1}$ follow the same procedure. For the sake of simplicity, we describe how in general the inverse of a covariance matrix, namely $\Sigma_{\mathcal{H}}^{-1}$ in the RKHS $\mathcal{H}$, can be obtained. In doing so, we start with the familiar Euclidean space. Given a set of pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$, we have

$$\Sigma = \frac{1}{n}\sum_i(\boldsymbol{x}_i-\boldsymbol{y}_i)(\boldsymbol{x}_i-\boldsymbol{y}_i)^T = \boldsymbol{ZJJ}^T\boldsymbol{Z}^T\,, \quad (7)$$

with $\boldsymbol{Z} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n]$ and

$$\boldsymbol{JJ}^T = \frac{1}{n}\begin{bmatrix}\mathbf{I}_n & -\mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{I}_n\end{bmatrix}.$$

Accordingly, the covariance matrix $\Sigma_{\mathcal{H}}$ in the RKHS $\mathcal{H}$ with dimensionality $|\mathcal{H}|$ can be written as

$$\Sigma_{\mathcal{H}} = \Phi_{\boldsymbol{Z}}\boldsymbol{JJ}^T\Phi_{\boldsymbol{Z}}^T, \quad (8)$$

with $\Phi_{\boldsymbol{Z}} = \big[\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \cdots, \phi(\boldsymbol{x}_n), \phi(\boldsymbol{y}_1), \phi(\boldsymbol{y}_2), \cdots, \phi(\boldsymbol{y}_n)\big]$.

The difficulty in obtaining $\Sigma_{\mathcal{H}}^{-1}$ lies in the fact that for universal kernels (e.g., Gaussian kernel) the dimensionality of $\mathcal{H} \to \infty$. With limited data, $\Sigma_{\mathcal{H}}$ is positive semi-definite and hence $\Sigma_{\mathcal{H}}^{-1}$ does not theoretically exist. As such, we need to preserve the positive eigenvalues and the associated eigenvectors of $\Sigma_{\mathcal{H}}$ and regularize the zero ones. This can be understood as the best approximation to $\Sigma_{\mathcal{H}}$ given the set $\boldsymbol{Z}$. In doing so, we make use of the relationship between the eigenvalues and eigenvectors of the product $\boldsymbol{AA}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$.

In particular, let $\mathbb{K}_{\boldsymbol{Z}} \in \mathbb{R}^{2n\times 2n}$ be the kernel matrix of $\boldsymbol{Z}$, i.e.,

$$[\mathbb{K}_{\boldsymbol{Z}}]_{i,j} = \begin{cases} k(\boldsymbol{x}_i, \boldsymbol{x}_j), & i, j \le n \\ k(\boldsymbol{y}_i, \boldsymbol{y}_j), & i, j > n \\ k(\boldsymbol{x}_i, \boldsymbol{y}_j), & \text{otherwise} \end{cases}$$

[1] This is called the restricted metric learning and is more challenging than the unrestricted scenario where the learning algorithm has access to the class labels of the samples $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$.

Let the SVD decomposition of $\boldsymbol{J}^T\Phi_{\mathbf{Z}}^T\Phi_{\mathbf{Z}}\boldsymbol{J} = \boldsymbol{J}^T\mathbb{K}_{\mathbf{Z}}\boldsymbol{J}$ be $\boldsymbol{V}_{\mathbf{Z}}\Lambda_{\mathbf{Z}}\boldsymbol{V}_{\mathbf{Z}}^T$. The regularized estimate of $\Sigma_{\mathcal{H}}$ then can be written Harandi et al. (2014)

$$\hat{\Sigma}_{\mathcal{H}} = \Phi_{\mathbf{Z}}\boldsymbol{W}_{\mathbf{Z}}\boldsymbol{W}_{\mathbf{Z}}^T\Phi_{\mathbf{Z}}^T + \rho\mathbf{I}_{\mathcal{H}} \ , \tag{9}$$

where $\boldsymbol{W}_{\mathbf{Z}} = \boldsymbol{J}\boldsymbol{V}_{\mathbf{Z}}(\mathbf{I}_{2n} - \rho\Lambda_{\mathbf{Z}}^{-1})^{0.5}$ with ρ being a positive regularizor.

To obtain $\hat{\Sigma}_{\mathcal{H}}^{-1}$, we make use of the Woodbury matrix identity Golub and Van Loan (2012) to arrive at

$$\hat{\Sigma}_{\mathcal{H}}^{-1} = (\Phi_{\mathbf{Z}}\boldsymbol{W}_{\mathbf{Z}}\boldsymbol{W}_{\mathbf{Z}}^T\Phi_{\mathbf{Z}}^T + \rho\mathbf{I}_{\mathcal{H}})^{-1} = \frac{1}{\rho}\mathbf{I}_{\mathcal{H}} - \frac{1}{\rho}\Phi_{\mathbf{Z}}\boldsymbol{W}_{\mathbf{Z}}\Lambda_{\mathbf{Z}}^{-1}\boldsymbol{W}_{\mathbf{Z}}^T\Phi_{\mathbf{Z}}^T \ . \tag{10}$$

This lets us answer the first question, i.e., obtaining $\Sigma_{\mathcal{H},s}^{-1} - \Sigma_{\mathcal{H},d}^{-1}$ as

$$\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1} = \frac{1}{\rho}\Phi_{\mathbf{Z}_d}\boldsymbol{W}_{\mathbf{Z}_d}\Lambda_{\mathbf{Z}_d}^{-1}\boldsymbol{W}_{\mathbf{Z}_d}^T\Phi_{\mathbf{Z}_d}^T - \frac{1}{\rho}\Phi_{\mathbf{Z}_s}\boldsymbol{W}_{\mathbf{Z}_s}\Lambda_{\mathbf{Z}_s}^{-1}\boldsymbol{W}_{\mathbf{Z}_s}^T\Phi_{\mathbf{Z}_s}^T \ . \tag{11}$$

4.2. Projection onto the Positive Cone in $\mathcal{H}$

We note that the form of $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$ cannot be directly used to define a Mahalanobis distance in $\mathcal{H}$. This is because the difference of two positive definite matrices is not necessarily positive definite, violating the very basic definition of a metric given in §3.

In this part, we propose two methods to project $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$ onto the positive cone in $\mathcal{H}$. In the first method, though being an approximation, the projection can be obtained in closed-form. The second method relies on Riemannian optimization techniques and is an iterative scheme. Our experiments suggest that the solution obtained by the second method is more reliable. As such, we recommend to use the first solution only if the burden of Riemannian optimization techniques is a concern.

Our main idea here is to define an implicit form of a positive definite matrix and then minimize a measure of similarity between the implicit form and $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$. More specifically, with $\boldsymbol{C} \in \mathcal{S}_{++}^n$ and trn denoting a set of n training vectors, we propose to solve the following problem as a means of projection onto the cone of positive definite matrices in $\mathcal{H}$

$$\arg\min_{\boldsymbol{C}\succ 0} \mathcal{L}(\boldsymbol{C}) \triangleq \left\|\Phi_{trn}\boldsymbol{C}\Phi_{trn}^T + \Phi_{\mathbf{Z}_s}\boldsymbol{A}_s\Phi_{\mathbf{Z}_s}^T - \Phi_{\mathbf{Z}_d}\boldsymbol{A}_d\Phi_{\mathbf{Z}_d}^T\right\|_F^2 \ , \tag{12}$$

where $\boldsymbol{A}_s = \boldsymbol{W}_{\mathbf{Z}_s}\Lambda_{\mathbf{Z}_s}^{-1}\boldsymbol{W}_{\mathbf{Z}_s}^T$ and $\boldsymbol{A}_d = \boldsymbol{W}_{\mathbf{Z}_d}\Lambda_{\mathbf{Z}_d}^{-1}\boldsymbol{W}_{\mathbf{Z}_d}^T$.

Expanding the Frobenious norm and considering only the terms that include $\boldsymbol{C}$, we get

$$\begin{aligned}\mathcal{L}(\boldsymbol{C}) = &\operatorname{Tr}(\mathbb{K}_{trn}\boldsymbol{C}\mathbb{K}_{trn}\boldsymbol{C}) + 2\operatorname{Tr}(\boldsymbol{K}_{\mathbf{Z}_s,trn}\boldsymbol{C}\boldsymbol{K}_{\mathbf{Z}_s,trn}^T\boldsymbol{A}_s) \\ &- 2\operatorname{Tr}(\boldsymbol{K}_{\mathbf{Z}_d,trn}\boldsymbol{C}\boldsymbol{K}_{\mathbf{Z}_d,trn}^T\boldsymbol{A}_d) + const \ .\end{aligned} \tag{13}$$

First Solution (The Approximation).
Without considering the constraint $\boldsymbol{C} \succ 0$, a closed-form solution can be obtained by setting as

$$\begin{aligned}&\nabla_{\boldsymbol{C}}(\mathcal{L}(\boldsymbol{C})) = 0 \\ &\Rightarrow 2\mathbb{K}_{trn}\boldsymbol{C}\mathbb{K}_{trn} + 2\boldsymbol{K}_{\mathbf{Z}_s,trn}^T\boldsymbol{A}_s\boldsymbol{K}_{\mathbf{Z}_s,trn} - 2\boldsymbol{K}_{\mathbf{Z}_d,trn}^T\boldsymbol{A}_d\boldsymbol{K}_{\mathbf{Z}_d,trn} = 0 \\ &\Rightarrow \boldsymbol{C}^* = \mathbb{K}_{trn}^{-1}\Big(\boldsymbol{K}_{\mathbf{Z}_d,trn}^T\boldsymbol{A}_d\boldsymbol{K}_{\mathbf{Z}_d,trn} - \boldsymbol{K}_{\mathbf{Z}_s,trn}^T\boldsymbol{A}_s\boldsymbol{K}_{\mathbf{Z}_s,trn}\Big)\mathbb{K}_{trn}^{-1} \ .\end{aligned} \tag{14}$$

Unlike $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$, which is implicit, $\boldsymbol{C}^*$ has an explicit form. As such, projecting onto the set of positive definite matrices can be attained by simply applying the Proj(·) operator (see §3). We note that the proposed two step approach (minimizing followed by projection) does not necessarily provide the closest point inside the positive cone to $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$, hence the name approximation. In our experiments, we refer to this method as CF-K^{2}ISSME .

Second Solution (The Riemannian Approach).
Classical optimization methods generally turn a constrained optimization problem into a sequence of unconstrained problems for which unconstrained techniques can be applied. In contrast, recent advances in optimization on Riemannian manifolds offer an alternative if the constraints can be modeled by a Riemannian structure. This is indeed the case here.

Consider a constrained optimization problem in the form of minimizing $f(\boldsymbol{x})$ with the constraint that $\boldsymbol{x}$ should lie on a Riemannian manifold $\mathcal{M}$ (think of a Riemannian manifold as a smooth surface embedded in some Euclidean space). This problem can be understood as an unconstrained problem in the form $f : \mathcal{M} \to \mathbb{R}$. Optimization techniques on Riemannian manifolds (e.g., Riemannian Gradient Descent (RGD)) enjoy several unique properties (e.g., convergence, smooth behavior) that make them competent alternatives to classical techniques.

To apply RGD on $f : \mathcal{M} \to \mathbb{R}$, one ultimately needs to have the gradient of f at $\boldsymbol{x}$, i.e., $\operatorname{grad}_{\boldsymbol{x}} f \in T_{\boldsymbol{x}}\mathcal{M}$ with $T_{\boldsymbol{x}}\mathcal{M}$ denoting the tangent space of $\mathcal{M}$ at $\boldsymbol{x}$. For the problem of our interest, i.e., minimizing $\mathcal{L}(\boldsymbol{C})$ while satisfying $\boldsymbol{C} \succ 0$, the Riemannian structure that describes the constraint is $\mathcal{S}_{++}^n$, e.g. the manifold of SPD matrices. For a smooth function $f : \mathcal{S}_{++}^n \to \mathbb{R}$, the gradient $grad_{\boldsymbol{C}}\ f \in T_{\boldsymbol{C}}\mathcal{S}_{++}^n$ is given by

$$grad_{\boldsymbol{C}}\ f = \boldsymbol{C} sym(\nabla_{\boldsymbol{C}}(f))\boldsymbol{C} \ , \tag{15}$$

where $\nabla_{\boldsymbol{C}}(\cdot)$ is the Euclidean gradient w.r.t $\boldsymbol{C}$ and

$$sym(\boldsymbol{X}) = \frac{\boldsymbol{X} + \boldsymbol{X}^T}{2} \ .$$

We have already computed $\nabla_{\boldsymbol{C}}(\cdot)$ in the previous section, hence applying RGD is straightforward. In our experiments, we refer to this method as R-K^{2}ISSME . We use the implementation provided by the Manopt toolbox Boumal et al. (2014) to determine $\boldsymbol{C}$.

Figure 1 illustrates the convergence behavior of our R-K^{2}ISSME algorithm using the iLIDS dataset Zheng et al. (2009). In all our experiments, we observed that the algorithm typically converges in less than 30 iterations, thus making it scalable to learning large metrics. To have a complete picture,

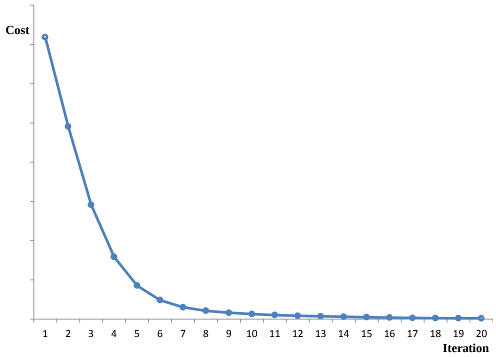

Fig. 1: Convergence behavior of our R-K^{2}ISSME algorithm.

we report the computational load of our proposal for our last experiment in §5. Averaging over 10 splits on a quad-core machine using Matlab, computing the kernel matrix for all samples takes about 110 seconds. Computing the metric matrix in the CF-K^{2}ISSME takes 0.7 seconds, making it the preferred technique when computational cost is important. Finally, performing 30 iterations in the R-K^{2}ISSME takes near 45 seconds.

The complexity of our methods is mostly dictated by the computational cost of computing the objective function and the gradient (or equivalently obtaining $\boldsymbol{C}^*$). These steps take $O(n^3+n^2d+nd^2+n^2s+ns^2)$ flops where n, s, d denote the number of training, similar and dissimilar samples, respectively.

4.3. *Efficient Computation of the Mahalanobis Distances in $\mathcal{H}$*

Once $\boldsymbol{C}$ is obtained either by the first method or the second solution, the Mahalanobis distance in $\mathcal{H}$ can be obtained as

$$\begin{aligned} d_{\mathcal{H}}(\boldsymbol{p},\boldsymbol{q}) &= \sqrt{(\phi(\boldsymbol{p})-\phi(\boldsymbol{q}))^T\Phi_{trn}\boldsymbol{C}\Phi_{trn}^T(\phi(\boldsymbol{p})-\phi(\boldsymbol{q}))} \qquad (16)\\ &= \sqrt{\boldsymbol{k}_{p,trn}\boldsymbol{C}\boldsymbol{k}_{p,trn}^T - 2\boldsymbol{k}_{p,trn}\boldsymbol{C}\boldsymbol{k}_{q,trn}^T + \boldsymbol{k}_{q,trn}\boldsymbol{C}\boldsymbol{k}_{q,trn}^T}\,. \end{aligned}$$

which answers our third question.

4.4. *The Nyström Solution*

In the previous parts, we showed how the KISSME algorithm can be kernelized. Very related to our goal in this paper is the concept of approximating the feature map ϕ of a pd kernel. For specific kernels (e.g., the Gaussian kernel), such approximations are known Vedaldi and Zisserman (2012). Hence, one can obtain a vectorized representation of the kernel space towards kernelizing the KISSME algorithm.

For more complicated kernel functions, one can employ the Nyström method to kernelize KISSME. The Nyström method is a data-driven approach to estimate the RKHS induced by a kernel. Briefly, let $\mathcal{D} = \{\boldsymbol{t}_i\}_{i=1}^M$ be a collection of M training samples. A rank D approximation to $\boldsymbol{K} = [k(\boldsymbol{t}_i,\boldsymbol{t}_j)]_{M\times M}$ can be obtained using SVD as $\boldsymbol{K} \simeq \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T$. Here, $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ is a diagonal matrix keeping the top D eigenvalues of $\boldsymbol{K}$ and $\boldsymbol{V} \in \mathbb{R}^{M\times D}$ is a column matrix storing the associated top eigenvectors. Having the low-rank representation at our disposal, a D-dimensional approximation to $\phi(\boldsymbol{x})$ is given by

$$\hat{\phi}(\boldsymbol{x}) = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{V}\Big(k(\boldsymbol{x},\boldsymbol{t}_1),\cdots,k(\boldsymbol{x},\boldsymbol{t}_M)\Big)^T. \qquad (17)$$

Fig. 2: From left to right four sample images of the iLIDS Zheng et al. (2009) and the CAVIAR Cheng et al. (2011) datasets are shown, respectively.

Table 1: CMC at rank r on the iLIDS dataset with $p = 60$ test individuals.

Method	**r = 1**	**r = 5**	**r = 10**	**r = 20**
kLFDA-χ^2	36.5%	64.1%	76.5%	88.5%
MFA-χ^2	32.6%	58.5%	71.5%	84.5%
LMNN	32.6%	56.2%	68.9%	83.0%
ITML	29.5%	50.3%	62.6%	76.4%
LDML	27.8%	53.2%	67.0%	82.5%
KISSME	30.3%	54.8%	68.3%	83.6%
Nyström-KISSME	33.1%	60.6%	73.2%	86.2%
CF-K^{2}ISSME	37.8%	64.3%	76.5%	88.7%
R-K^{2}ISSME	**38.1%**	**65.0%**	**78.2%**	**89.4%**

We will call this solution, i.e., obtaining $\hat{\phi}(\cdot)$ followed by applying the original KISSME algorithm, the Nyström-KISSME method.

5. Experiments

In this section, we compare our proposed methods with several state-of-the-art metric learning techniques. In particular, we evaluate the performance of our R-K^{2}ISSME , CF-K^{2}ISSME , and Nyström-KISSME against LMNN Weinberger and Saul (2009), ITML Davis et al. (2007), LDML Guillaumin et al. (2009), and KISSME Koestinger et al. (2012). As another indicator, we also measure our performance to dataset-specific baselines. For all the baselines, we carefully tune their parameters and report their maximum accuracies here.

In all the experiments, we follow the so-called restricted protocol, where only the set of similar/dissimilar pairs is available during training. Furthermore, we utilize the parameter-free Chi-squared kernel depicted below in R-K^{2}ISSME , CF-K^{2}ISSME and Nyström-KISSME ;

$$k_{\chi^2}(\boldsymbol{x},\boldsymbol{y}) = \sum_i \frac{2\boldsymbol{x}_i\boldsymbol{y}_i}{\boldsymbol{x}_i+\boldsymbol{y}_i}\,. \qquad (18)$$

5.1. *Person Reidentification*

As our first experiment, we tackled the task of person reidentification using two widely used datasets, namely iLIDS Zheng et al. (2009) and CAVIAR Cheng et al. (2011). The iLIDS dataset contains images of 119 pedestrians captured by 8 cameras with different view points in an airport. Each individual has 2 to 8 images, and the dataset exhibits severe occlusions caused by people and their luggage. The CAVIAR4REID (CAVIAR) dataset includes 1220 images of 72 different persons captured from two different cameras in an indoor shopping mall. The

Table 2: CMC at rank r on the CAVIAR dataset with $p = 36$ test individuals.

Method	r = 1	r = 5	r = 10	r = 20
kLFDA-χ^2	36.2%	64.0%	78.7%	92.2%
MFA-χ^2	37.7%	67.2%	82.1%	94.6%
LMNN	33.8%	61.9%	78.6%	92.0%
ITML	29.1%	61.4%	75.8%	92.0%
LDML	30.4%	62.5%	77.8%	91.2%
KISSME	31.4%	61.9%	77.8%	92.5%
Nyström-KISSME	37.5%	67.5%	82.5%	95.0%
CF-K^{2}ISSME	**38.7%**	**68.2%**	**82.9%**	**95.4%**
R-K^{2}ISSME	**38.7%**	67.1%	80.9%	95.0%

Table 3: Classification accuracies on various subsets of the KinFace-I dataset.

Method	F-D	F-S	M-D	M-S	Mean
NRML	65.2%	64.7%	65.4%	59.4%	63.7%
LMNN	63.2%	62.7%	63.4%	57.4%	61.7%
ITML	55.2%	58.3%	56.7%	55.6%	56.5%
LDML	57.1%	60.5%	57.4%	57.4%	58.1%
KISSME	65.4%	72.8%	66.7%	65.5%	67.6%
Nyström-KISSME	69.8%	**79.8%**	70.1%	68.5%	72.1%
CF-K^{2}ISSME	70.9%	**79.8%**	69.4%	66.0%	71.5%
R-K^{2}ISSME	**71.3%**	79.5%	**73.7%**	**69.4%**	**73.5%**

number of images per individual varies from 10 to 20. Sample images of both datasets are shown in Fig. 2.

In our experiments, we followed the standard single-shot protocol. That is, the dataset was randomly partitioned into two exclusive subset of individuals, with p individuals constituting the test set and the remaining ones forming the training data. The random partitioning was repeated 10 times. In each partition, one image from each individual in the test set was randomly selected as the reference image and the rest of the images were used as query images. This process was repeated 20 times.

As for features, we used the histogram based descriptors provided by Xiong et al. (2014) for fair comparisons[2]. More specifically, each image in the dataset is described by 16-bin histogram of RGB, YUV and HSV color channels, as well as texture histograms based on the Local Binary Patterns (LBP) Ojala et al. (2002) extracted from 6 non-overlapping horizontal bands. This leads to a 2580 dimensional descriptor for each image.

Aside from the aforementioned MML baselines, we compare our proposed algorithms with the state-of-the-art kernel Local Fisher Discriminant Analysis (kLFDA) Xiong et al. (2014) and Marginal Fisher Analysis (MFA) Xiong et al. (2014). Assuming Gaussian distribution for each class and using the Fisher discriminant objective, kLFDA finds a projection matrix to maximize the between-class scatters while minimizing the within-class scatters. MFA is a graph embedding dimensionality reduction method which allows to maximize the marginal discriminant even when the class distributions are not Gaussian.

We report performances in terms of the Cumulative Match Characteristic (CMC) curves for different rank values in Tables 1 and 2. To obtain CMC curves, a hit for rank k is considered if the correct class is identified among the k-nearest points of a query. From Table 1, we observe that our R-K^{2}ISSME achieves the highest scores for all the studied ranks. On the CAVIAR dataset, the best reported performance was achieved using the CF-K^{2}ISSME , while R-K^{2}ISSME works on par with that. It is worth mentioning that both kLFDA and MFA require the subject identities during training (i.e., they are unrestricted approaches) while our proposals do not require such additional information.

A parameter to take care of in R-K^{2}ISSME and CF-K^{2}ISSME is the number of eigenvalues and eigenvectors used to establish $\boldsymbol{W}_Z$ (see Eq. (10)). A similar parameter in conventional KISSME and Nyström-KISSME is the dimensionality of PCA (required as a preprocessing step) and rank of Nyström approximation, respectively. In Fig. 3, we analyze the sensitivity of R-K^{2}ISSME , CF-K^{2}ISSME , Nyström-KISSME and KISSME over the aforementioned parameters on the iLIDS dataset. Both R-K^{2}ISSME and CF-K^{2}ISSME generate demonstrate a robust and increasing performances when most of the energy is preserved. In contrast, the performance of Nyström-KISSME and KISSME may drop if more than 90% of energy is preserved.

As an indicator, on the iLIDS dataset, the deep net proposed in Ding et al. (2015) achieves 52.1%, 68.2%, 78.0%, and 88.8% at rank 1, 5, 10, and 20, respectively. Interestingly, our method performs on par or better than the deep solution for rank 5, 10, and 20 while underperforming at rank 1. This shows a potential research direction by incorporating the proposed technique in a deep net to benefit from deep architectures.

5.2. *Kinship Verification*

We performed another experiment to verify kinship relations from facial images. To this end, we made use of the KinFace-I dataset Lu et al. (2014) (see Fig. 4). The dataset contains images of four kin types: Father-Son (F-S), Father-Daughter (F-D), Mother-Son (M-S), and Mother-Daughter (M-D).

The coordinates of eyes in each face image are manually labelled, and facial regions are cropped and aligned into 64 × 64 templates. Then, histogram equalization is applied to mitigate the illumination variation. We have used the provided Local Binary Patterns (LBP) Ojala et al. (2002) features in our experiments. More specifically, each face image is divided into blocks of size 16 × 16 and for each block a 256 dimensional LBP histogram is extracted. The extracted histograms are finally concatenated to form a 4096 dimensional descriptor.

In Table 3, we compare our proposed algorithms against the baselines and the state-of-the-art NRML Lu et al. (2014) on the KinFace-I dataset Lu et al. (2014). R-K^{2}ISSME , CF-K^{2}ISSME and Nyström-KISSME outperform the state-of-the-art NRML by a large margin. For example, the gap between R-K^{2}ISSME and NRML is near 10%. We also note that R-K^{2}ISSME , CF-K^{2}ISSME and Nyström-KISSME are superior to the other metric learning baselines.

In Table 4, we provide the results on the KinFace-II dataset Lu et al. (2014). Here, our R-K^{2}ISSME again achieves the highest accuracy with CF-K^{2}ISSME being the second best.

[2] https://github.com/NEU-Gou/kernel-metric-learning-reid

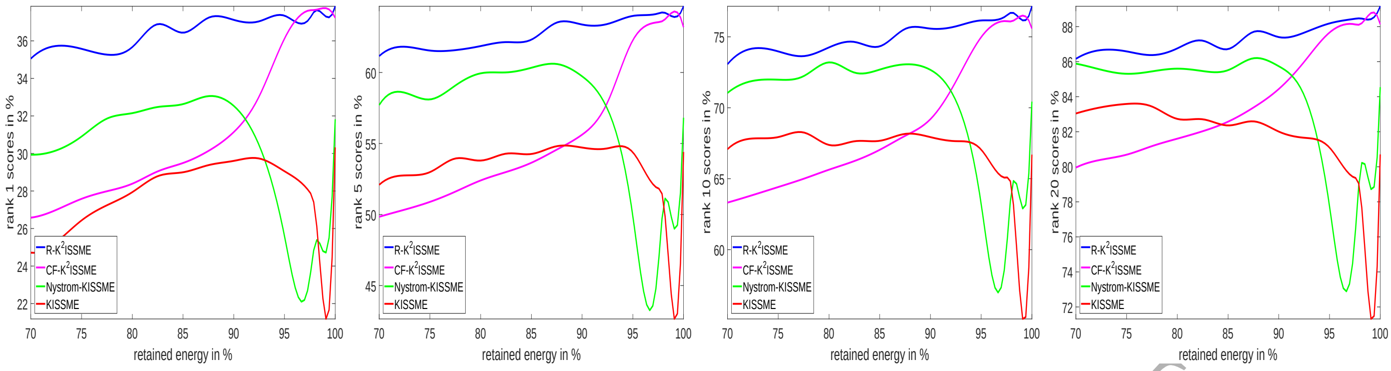

Fig. 3: Rank x scores vs retained variance of the data on the iLIDS dataset Zheng et al. (2009) where x is 1, 5, 10, 20.

Fig. 4: Examples of the KinFace-I and KinFace-II datasets Lu et al. (2014). From left to right two examples are shown in each column for kinship relations: F-D, F-S, M-D, and M-S, respectively.

Table 4: Classification accuracies on various subsets of the KinFace-II dataset.

Method	F-D	F-S	M-D	M-S	Mean
NRML	69.5%	69.0%	69.0%	69.8%	69.5%
LMNN	68.5%	68.0%	67.0%	68.8%	68.2%
ITML	63.6%	69.2%	63.4%	64.2%	65.1%
LDML	65.6%	68.0%	66.0%	65.8%	66.4%
KISSME	72.0%	68.6%	68.6%	68.6%	70.4%
Nyström-KISSME	62.6%	64.1%	**72.6%**	70.2%	67.4%
CF-K^{2}ISSME	73.0%	77.5%	69.2%	70.2%	72.5%
R-K^{2}ISSME	**75.6%**	**78.4%**	68.6%	**73.2%**	**74.0%**

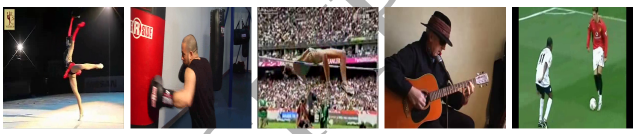

Fig. 5: Examples of the ASLAN dataset Kliper-Gross et al. (2012).

Table 5: Matching accuracies on various descriptors of the ASLAN dataset Kliper-Gross et al. (2012).

Method	HoG	HoF	Hnf
Baseline Kliper-Gross et al. (2012)	54.2%	54.0%	54.5%
LMNN	55.9%	53.5%	56.0%
ITML	55.6%	53.9%	55.9%
LDML	57.3%	56.5%	58.0%
KISSME	55.2%	52.8%	55.7%
Nyström-KISSME	55.6%	53.3%	56.0%
CF-K^{2}ISSME	57.3%	57.8%	57.5%
R-K^{2}ISSME	**57.9%**	**58.3%**	**58.2%**

Both R-K^{2}ISSME and CF-K^{2}ISSME comfortably outperform the state-of-the-art NRML Lu et al. (2014) method[3].

5.3. Action Similarity Matching

As our last experiment, we considered the task of action similarity recognition using the ASLAN dataset Kliper-Gross et al. (2012). The dataset contains 3,697 unique human action clips collected from YouTube, spanning 432 categories (see Fig. 5 for example frames). The benchmark protocol is a binary pair matching and the goal is to decide whether two videos present the same action or not. The sample distribution across the categories in the benchmark is quite unbalanced, with 116 categories possessing only one video clip. Furthermore, categories included in the test sets are not available during training.

An action is represented by spatio-temporal bag-of-words descriptor Laptev et al. (2008) with a codebook of size 5,000 evaluated individually on three different types of descriptors, namely Histogram of Oriented Gradients (HoG), Histogram of Optical Flow (HoF) and a combination of both (HnF). We followed the standard matching protocol on this dataset which makes use of 10 predefined splits of data. There are 12,000 samples including 5,400 training and 600 testing pairs of action videos in each split.

In Table 5, we compare our proposed algorithms against the baselines on the ASLAN dataset. Here, our R-K^{2}ISSME again achieves the highest accuracies, while the closed-form solution works on par with it. Compared to the conventional KISSME, the Nyström-KISSME offers a better recognition rate, demonstrating benefits of analysis in the estimated RKHS in this method. Lastly, for this larger-scale problem the LDML baseline works very competitively to our R-K^{2}ISSME .

6. Conclusions

In this paper, we kernelized the recently introduced "Keep It Simple and Straightforward MEtric" (KISSME) algorithm. This not only enables us to deal with non-linearity in data but also provides a principal way to employ KISSME on non-vectorized data (e.g., manifold-value data). Along the way, we developed two methods (a closed-form and a Riemannian

[3] We note that a recent study by López et al. discusses the bias in the KinFace dataset. Since our main goal here is to compare our proposal with other metric learning techniques, the bias does not harm the conclusions made here.

optimization based method) to project a matrix into the positive cone in an RKHS. As our last experiment suggests, when the computational load is important and larger scale problem is considered, the closed-form method can be the method of choice to obtain a more efficient (though approximated) solution. We also developed an approximated solution based on the Nyström method towards kernelizing KISSME. Our experiments demonstrate consistent improvements of the kernelized solutions over the original KISSME and other baselines. Given the importance of dimensionality reduction in the original KISSME algorithm, in the future we plan to combine dimensionality reduction and metric learning in an RKHS to achieve a more robust solution.

References

Baker, C.T., 1977. The numerical treatment of integral equations. Clarendon press.

Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R., 2014. Manopt, a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research 15, 1455–1459. URL: http://www.manopt.org.

Chen, J., Zhang, Z., Wang, Y., 2015. Relevance metric learning for person re-identification by exploiting listwise similarities. IEEE Transactions on Image Processing 24, 4741–4755. doi:10.1109/TIP.2015.2466117.

Cheng, D.S., Cristani, M., Stoppa, M., Bazzani, L., Murino, V., 2011. Custom pictorial structures for re-identification., in: Proc. British Machine Vision Conference, p. 6.

Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 539–546.

Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S., 2007. Information-theoretic metric learning, in: Proc. Int. Conference on Machine Learning, ACM. pp. 209–216.

Ding, S., Lin, L., Wang, G., Chao, H., 2015. Deep feature learning with relative distance comparison for person re-identification. Pattern Recognition 48, 2993–3003.

Faraki, M., Harandi, M.T., Porikli, F., 2015. Approximate infinite-dimensional region covariance descriptors for image classification, in: Acoustics, Speech and Signal Processing, 2015 IEEE International Conference on, IEEE. pp. 1364–1368.

Goldberger, J., Hinton, G.E., Roweis, S.T., Salakhutdinov, R., 2004. Neighbourhood components analysis, in: Proc. Advances in Neural Information Processing Systems, pp. 513–520.

Golub, G.H., Van Loan, C.F., 2012. Matrix computations. volume 3. JHU Press.

Guillaumin, M., Verbeek, J., Schmid, C., 2009. Is that you? metric learning approaches for face identification, in: Proc. Int. Conference on Computer Vision, IEEE. pp. 498–505.

Harandi, M., Salzmann, M., Hartley, R., 2017. Joint dimensionality reduction and metric learning: A geometric take, in: International Conference on Machine Learning.

Harandi, M., Salzmann, M., Porikli, F., 2014. Bregman divergences for infinite dimensional covariance matrices, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1003–1010.

Hoi, S.C., Liu, W., Lyu, M.R., Ma, W.Y., 2006. Learning distance metrics with contextual constraints for image retrieval, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 2072–2078.

Hu, J., Lu, J., Tan, Y.P., 2014. Discriminative deep metric learning for face verification in the wild, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1875–1882.

Kliper-Gross, O., Hassner, T., Wolf, L., 2012. The action similarity labeling challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 615–621.

Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P.M., Bischof, H., 2012. Large scale metric learning from equivalence constraints, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 2288–2295.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Proc. Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 1097–1105.

Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B., 2008. Learning realistic human actions from movies, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 1–8.

Li, Z., Chang, S., Liang, F., Huang, T., Cao, L., Smith, J., 2013. Learning locally-adaptive decision functions for person verification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3610–3617.

López, M.B., Boutellaa, E., Hadid, A., . Comments on the" kinship face in the wild" data sets .

Lu, J., Zhou, X., Tan, Y.P., Shang, Y., Zhou, J., 2014. Neighborhood repulsed metric learning for kinship verification. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 331–345.

Ojala, T., Pietikäinen, M., Mäenpää, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987.

Quang, M.H., San Biagio, M., Murino, V., 2014. Log-hilbert-schmidt metric between positive definite operators on hilbert spaces, in: Proc. Advances in Neural Information Processing Systems, pp. 388–396.

Song, H.O., Xiang, Y., Jegelka, S., Savarese, S., 2016. Deep metric learning via lifted structured feature embedding .

Sun, Y., Wang, X., Tang, X., 2014. Deep learning face representation from predicting 10,000 classes, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1891–1898.

Vedaldi, A., Zisserman, A., 2012. Efficient additive kernels via explicit feature maps. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 480–492.

Weinberger, K.Q., Saul, L.K., 2009. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244.

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

Xiong, F., Gou, M., Camps, O., Sznaier, M., 2014. Person re-identification using kernel-based metric learning methods, in: Proc. European Conference on Computer Vision. Springer, pp. 1–16.

Zheng, W.S., Gong, S., Xiang, T., 2009. Associating groups of people., in: Proc. British Machine Vision Conference, p. 6.

Zheng, W.S., Gong, S., Xiang, T., 2015. Towards open-world person re-identification by one-shot group-based verification. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 591–606.