Joint Dimensionality Reduction and Metric Learning: A Geometric Take

Mehrtash Harandi [1 2] **Mathieu Salzmann** [3] **Richard Hartley** [2 1]

Abstract

To be tractable and robust to data noise, existing metric learning algorithms commonly rely on PCA as a pre-processing step. How can we know, however, that PCA, or any other specific dimensionality reduction technique, is the method of choice for the problem at hand? The answer is simple: We cannot! To address this issue, in this paper, we develop a Riemannian framework to jointly learn a mapping performing dimensionality reduction and a metric in the induced space. Our experiments evidence that, while we directly work on high-dimensional features, our approach yields competitive runtimes with and higher accuracy than state-of-the-art metric learning algorithms.

1. Introduction

"*To make it tractable for the distance metric learning algorithms we perform dimensionality reduction by PCA to a 100 dimensional subspace*" (Koestinger et al., 2012). "*Like most of the metric learning methods we first center the dataset and reduce the dimensionality to a n-dimensional space by PCA*" (Bohné et al., 2014). "*ITML and LDML are intractable when using 600 PCA dimensions"* (Guillaumin et al., 2009).

These quotations, extracted from the metric learning literature, give rise to a simple question: ***Is PCA, or, more generally, dimensionality reduction, a must to make metric learning work on high-dimensional data, such as that in computer vision problems?***

To quantify this, in the top portion of Table 1, we provide the area under the ROC curve of state-of-the-art metric learning techniques applied to the ASLAN dataset (Kliper-Gross et al., 2012) using various PCA dimensions (denoted by p). These results suggest that state-of-the-art methods either scale poorly with the dimensionality of the input and thus require PCA to remain tractable (*e.g.*, LDML), or require PCA to achieve an accuracy comparable to the other baselines (*e.g.*, KISSME).

In essence, this observation indicates that dimensionality reduction is beneficial to (i) reduce the computational burden of the algorithms; and (ii) extract the relevant information from the original noisy data. However, it also raises an additional question: ***Is PCA, or any other specific dimensionality reduction technique, really the best method for the problem at hand?*** In other words, ***Shouldn't we rather learn the low-dimensional representation and the metric jointly?***

Motivated by these questions, in this paper, we introduce a unified formulation for dimensionality reduction and metric learning. As suggested by our results on the ASLAN dataset in the bottom row of Table 1, our method outperforms the state-of-the-art metric learning techniques. Furthermore, despite the fact that we directly use high-dimensional features as input, our method has comparable runtimes to that of the fastest algorithms working on PCA-based low-dimensional representations.

In the context of Mahalanobis metric learning, several methods have proposed to allow the metric $\boldsymbol{M}$ to have low rank, thus inherently performing dimensionality reduction. Most methods, however, achieve this implicitly, by letting $\boldsymbol{M}$ be positive *semi*-definite (Weinberger & Saul, 2009; Davis et al., 2007), which, as opposed to explicit dimensionality reduction, does not reduce the computational cost of these algorithms. As a consequence, they still need to rely on PCA as a pre-processing step in practice. While (Lu et al., 2014) explicitly decomposes $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$, it enforces orthogonality constraints on $\boldsymbol{L}$ to disambiguate the solutions, thus effectively only performing dimensionality reduction, and not metric learning. By contrast, our approach lets us learn a complete Mahalanobis metric jointly with a low-dimensional projection.

At the heart of our joint dimensionality reduction and metric learning formulation lie notions of Riemannian geometry and quotient spaces. More specifically, we model the projection to a low-dimensional space as a point on a Stiefel manifold, and the metric in this space as a Symmetric Pos-

[1]Data61, CSIRO, Canberra, Australia [2]Australian National University, Canberra, Australia [3]CVLab, EPFL, Switzerland. Correspondence to: Mehrtash Harandi <mehrtash.harandi@anu.edu.au>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

Method	p = 25	p = 150	p = 500	p = 1000
NCA (Goldberger et al., 2004)	0.594 (610s)	0.586 (635s)	0.584 (720s)	0.584 (850s)
ITML (Davis et al., 2007)	0.575 (13s)	0.579 (100s)	0.569 (1360s)	0.571 (10050s)
LDML (Guillaumin et al., 2009)	0.602 (220s)	0.598 (900s)	0.611 (3000s)	0.609 (6900s)
LMNN (Weinberger & Saul, 2009)	0.587 (37s)	0.591 (42s)	0.585 (1130s)	0.583 (2325s)
KISSME (Koestinger et al., 2012)	0.574 (8s)	0.522 (15s)	0.504 (**35**s)	0.501 (**100**s)
GMML (Zadeh et al., 2016)	0.570 (9s)	0.554 (34s)	0.543 (130s)	0.539 (370s)
DRML (Ours)	**0.630** (**7**s)	**0.627** (**12**s)	**0.621** (105s)	**0.617** (360s)

Table 1: AUCs and training times for the state-of-the-art metric learning techniques and for our approach (DRML) on the ASLAN dataset (Kliper-Gross et al., 2012). The baseline algorithms were applied after projecting the features to a p-dimensional space by PCA, whereas our method learns the p-dimensional representation and the metric jointly.

itive Definite (SPD) matrix. We then show that our search space reduces to a quotient of the product space of the Stiefel and SPD manifolds with the orthogonal group. By building upon recent advances in optimization on Riemannian matrix manifolds (Absil et al., 2009), we therefore develop a mathematical framework that effectively and efficiently lets us find a solution in this space. Furthermore, we show that our formulation can be kernelized. This not only lets us handle non-linearity in the data, but also makes our approach applicable to non-vectorial input data, such as linear subspaces (Harandi et al., 2014), which have proven beneficial for many recognition tasks.

We demonstrate the benefits of our joint dimensionality reduction and metric learning approach over existing metric learning schemes on several tasks, including action similarity matching, face verification and person re-identification.

2. Mathematical Background

In this work, as most metric learning algorithms, we are interested in learning a Mahalanobis distance defined below.

Definition 1 (The Mahalanobis distance)**.** *The Mahalanobis distance between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ in $\mathbb{R}^n$ is defined as*

$$d^2_{\boldsymbol{M}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2_{\boldsymbol{M}} = (\boldsymbol{x} - \tilde{\boldsymbol{x}})^T \boldsymbol{M} (\boldsymbol{x} - \tilde{\boldsymbol{x}}) \ . \quad (1)$$

To have a valid metric, the Mahalanobis matrix $\boldsymbol{M}$ must be positive definite.

As will be shown in Section 3, our approach to learning a Mahalanobis metric can be formulated as a non-convex optimization problem on a Riemannian manifold. This type of problems can be expressed with the general form

$$\begin{aligned} &\text{minimize } f(\boldsymbol{z}) \\ &\text{s.t.} \quad \boldsymbol{z} \in \mathcal{M} \ , \end{aligned} \quad (2)$$

where $\mathcal{M}$ is a Riemannian manifold, *i.e.*, informally, a smooth surface that locally resembles a Euclidean space.

While Riemannian manifolds can often be explicitly encoded in terms of constraints on $\boldsymbol{z}$, the recent advances in Riemannian optimization techniques (Absil et al., 2009) have shown the benefits of truly exploiting the geometry of the manifold over standard constrained optimization. As a consequence, these techniques have become increasingly popular in diverse application domains (Mishra et al., 2014; Harandi et al., 2017; Cunningham & Ghahramani, 2015). A detailed discussion of Riemannian optimization goes beyond the scope of this paper, and we refer the interested reader to (Absil et al., 2009).

As will be discussed in details in Section 3, we formulate metric learning in the quotient space of the product space of two Riemannian manifolds with the orthogonal group. The two Riemannian manifolds at the heart of this formulation are the Stiefel manifold and the manifold of Symmetric Positive Definite (SPD) matrices defined below.

Definition 2 (The Stiefel Manifold)**.** *The set of $(n \times p)$-dimensional matrices, $p \leq n$, with orthonormal columns endowed with the Frobenius inner product*[1] *forms a compact Riemannian manifold called the Stiefel manifold* $\mathrm{St}(p, n)$ *(Boothby, 2003).*

$$\mathrm{St}(p, n) \triangleq \{\boldsymbol{W} \in \mathbb{R}^{n \times p} : \boldsymbol{W}^T \boldsymbol{W} = \mathbf{I}_p\} \ . \quad (3)$$

Definition 3 (The SPD Manifold)**.** *The set of $(p \times p)$ dimensional real, SPD matrices endowed with the Affine Invariant Riemannian Metric (AIRM) (Pennec et al., 2006) forms the SPD manifold $\mathcal{S}^p_{++}$.*

$$\mathcal{S}^p_{++} \triangleq \{\boldsymbol{M} \in \mathbb{R}^{p \times p} : \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v} > 0, \ \forall \boldsymbol{v} \in \mathbb{R}^p - \{\mathbf{0}_p\}\} \ . \quad (4)$$

The dimensionality of $\mathrm{St}(p, n)$ and $\mathcal{S}^p_{++}$ are $np - \frac{1}{2}p(p+1)$ and $p(p+1)/2$, respectively.

3. Our Approach

Our goal, as that of many other metric learning algorithms, is to learn a Mahalanobis distance between the input measurements. Ideally, this distance should reflect the class

[1] Note that the literature is divided between this choice and another form of Riemannian metric. See (Edelman et al., 1998) for details.

labels of the samples. Furthermore, motivated by our analysis of existing methods, which all benefit from a PCA preprocessing step, we also seek to reduce the dimensionality of the data. However, in contrast to existing methods, we propose to learn the lower-dimensional representation and the Mahalanobis distance in that space *jointly*.

More specifically, we want to learn a projection $\boldsymbol{W} : \mathbb{R}^n \to \mathbb{R}^p$ and a Mahalanobis matrix $\boldsymbol{M} \in \mathcal{S}^p_{++}$, such that the induced distance in $\mathbb{R}^p$ is more discriminative. To this end, let $\mathbb{X} = \{(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i, y_i)\}_{i=1}^m$ be a set of triplets, where $\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i \in \mathbb{R}^n$ are the feature vectors of two training samples, and the label $y_i \in \{0, 1\}$ determines whether $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$ are similar ($y_i = 1$) or not ($y_i = 0$). The Mahalanobis distance between $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$ in the low-dimensional space can thus be written as

$$\begin{aligned} d^2_{\boldsymbol{M},\boldsymbol{W}}(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) &= (\boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{W}^T\tilde{\boldsymbol{x}}_i)^T\boldsymbol{M}(\boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{W}^T\tilde{\boldsymbol{x}}_i) \\ &= (\boldsymbol{x} - \tilde{\boldsymbol{x}}_i)^T\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T(\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i) \ . \end{aligned} \tag{5}$$

To learn a latent space whose Mahalanobis distance reflects class similarity, we make use of the logistic loss. More precisely, for each pair of samples $(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i)$ sharing the same label, *i.e.*, $y_i = 1$, we define the loss

$$\ell(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i | y_i = 1) = \log(1 + p_i) \ , \tag{6}$$

with

$$p_i = \exp\left(\beta(\boldsymbol{x} - \tilde{\boldsymbol{x}})^T\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T(\boldsymbol{x} - \tilde{\boldsymbol{x}})\right) \ , \beta > 0. \tag{7}$$

Conversely, for a pair of samples $(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}_j)$ whose labels differ, *i.e.*, $y_j = 0$, we define the loss

$$\ell(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}_j | y_j = 0) = \log(1 + p_j^{-1}) \ . \tag{8}$$

Intuitively, the loss of Eq. 6 is minimized when $d^2_{\boldsymbol{M},\boldsymbol{W}}(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) \to 0$, whereas the loss of Eq. 8 is minimized when $d^2_{\boldsymbol{M},\boldsymbol{W}}(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}_j) \to \infty$.

The losses for all training triplets can be grouped into a cost function of the form

$$\begin{aligned} \mathcal{L}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X}) \triangleq &\sum_{i|y_i=1} \log(1 + p_i) \\ &+ \sum_{i|y_i=0} \log(1 + p_i^{-1}) + \lambda r(\boldsymbol{M}, \boldsymbol{M}_0), \end{aligned} \tag{9}$$

which further encodes a regularizer on $\boldsymbol{M}$. This regularizer, $r : \mathcal{S}^p_{++} \times \mathcal{S}^p_{++} \to \mathbb{R}_+$, allows us to exploit prior knowledge on the Mahalanobis matrix, encoded by a reference matrix $\boldsymbol{M}_0$. Following common practice (Davis et al., 2007; Hoffman et al., 2014), we make use of the asymmetric Burg divergence, which yields

$$r(\boldsymbol{M}, \boldsymbol{M}_0) = \operatorname{Tr}(\boldsymbol{M}\boldsymbol{M}_0^{-1}) - \log\det(\boldsymbol{M}\boldsymbol{M}_0^{-1}) - p \ . \tag{10}$$

In our experiments, since typically no strong prior is available, we simply use $\boldsymbol{M}_0 = \mathbf{I}_p$, *i.e.*, the identity matrix. Joint dimensionality reduction and metric learning can then be achieved by minimizing the cost function of Eq. 9 w.r.t. $\boldsymbol{W}$ and $\boldsymbol{M}$. To avoid degeneracies, and following common practice in dimensionality reduction, we constrain $\boldsymbol{W}$ to be a matrix with orthonormal columns. That is,

$$\boldsymbol{W}^T\boldsymbol{W} = \mathbf{I}_p. \tag{11}$$

With this constraint, $\boldsymbol{W}$ is in fact a point on the Stiefel manifold $\operatorname{St}(p, n)$. Since both $\boldsymbol{M}$ and $\boldsymbol{W}$ lie on Riemannian manifolds, albeit different ones, we propose to make use of Riemannian optimization to solve our problem, as described below.

3.1. Manifold-based Optimization

To determine $\boldsymbol{W}$ and $\boldsymbol{M}$, we need to solve the optimization problem

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{M}} \quad & \mathcal{L}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X}) \\ \text{s.t.} \quad & \boldsymbol{W}^T\boldsymbol{W} = \mathbf{I}_p, \ \boldsymbol{M} \succ 0 \ . \end{aligned} \tag{12}$$

Jointly minimizing with respect to $\boldsymbol{W}$ and $\boldsymbol{M}$ can be achieved by making use of the product space of the Stiefel and SPD manifolds, $\mathcal{M}_p = \operatorname{St}(p, n) \times \mathcal{S}^p_{++}$. Both $\operatorname{St}(p, n)$ and $\mathcal{S}^p_{++}$ are smooth homogeneous spaces and their product preserves smoothness and differentiability (Absil et al., 2009). Thus, $\mathcal{M}_p$ can be given a Riemannian structure. However, in our case, a closer look at $\mathcal{L}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X})$ reveals that

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X}) = \mathcal{L}(\boldsymbol{W}\boldsymbol{R}, \boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}|\mathbb{X}) \ , \forall \boldsymbol{R} \in \mathcal{O}_p \ , \tag{13}$$

where $\mathcal{O}_p$ is the orthogonal group. This implies that

$$\pi : \mathcal{M}_p \times \mathcal{O}_p \to \mathcal{M}_p : \big((\boldsymbol{W}, \boldsymbol{M}), \boldsymbol{R}\big) \to \big(\boldsymbol{W}\boldsymbol{R}, \boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}\big) \tag{14}$$

is a right group action on $\mathcal{M}_p$. The theorem below establishes an important property about the action of $\mathcal{O}_p$ on $\mathcal{M}_p$, which will prove crucial to our develop our approach.

Theorem 1. *The set* $\mathcal{M} \triangleq \big(\operatorname{St}(p, n) \times \mathcal{S}^p_{++}\big)\backslash\mathcal{O}(p)$ *with the equivalence relation*

$$[(\boldsymbol{W}, \boldsymbol{M})] \sim \Big\{\big(\boldsymbol{W}\boldsymbol{R}, \boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}\big); \ \forall \boldsymbol{R} \in \mathcal{O}(p)\Big\} \tag{15}$$

and Riemannian metric

$$\begin{aligned} g_{(\boldsymbol{W},\boldsymbol{M})}\big((\xi_{\boldsymbol{W}}, \xi_{\boldsymbol{M}}), (\varsigma_{\boldsymbol{W}}, \varsigma_{\boldsymbol{M}})\big) = &\ 2\operatorname{Tr}(\xi^T_{\boldsymbol{W}}\varsigma_{\boldsymbol{W}}) \\ &+ \operatorname{Tr}(\boldsymbol{M}^{-1}\xi_{\boldsymbol{M}}\boldsymbol{M}^{-1}\varsigma_{\boldsymbol{M}}) \end{aligned} \tag{16}$$

forms a Riemannian quotient manifold.

Proof. See the supplementary material. □

The search space of our problem therefore truly is this Riemannian manifold $\mathcal{M}$. To be able to perform Riemannian optimization on $\mathcal{M}$, below, we derive the required entities.

The Geometry of $\mathcal{M}$

The general theory of quotient manifolds (Lee, 2003; Absil et al., 2009) tells us that the equivalence relation splits the tangent space of $\mathcal{M}_p$ at $\Omega = (\boldsymbol{W}, \boldsymbol{M})$ into two complementary parts: the horizontal space $\mathcal{H}_\Omega\mathcal{M}_p$ and the vertical space $\mathcal{V}_\Omega\mathcal{M}_p$. These two spaces are such that

$$g_p(\boldsymbol{h}_\Omega, \boldsymbol{v}_\Omega) = 0,\ \forall \boldsymbol{h}_\Omega \in \mathcal{H}_\Omega\mathcal{M}_p \text{ and } \forall \boldsymbol{v}_\Omega \in \mathcal{V}_\Omega\mathcal{M}_\mathrm{p}\ , \tag{17}$$

where g_p is the Riemannian metric of the product manifold $\mathcal{M}_p$. The vertical space $\mathcal{V}_\Omega\mathcal{M}_p$ has the property that projecting any of its vectors to $\mathcal{M}_p$ via the exponential map yields a point in the equivalence class of Ω. Therefore, the tangent space of $\mathcal{M}$ can be identified with the horizontal space, *i.e.*, $T_{[\Omega]}\mathcal{M} \triangleq \mathcal{H}_\Omega\mathcal{M}_p$.

A tangent vector $\xi^{\uparrow}_\Omega \in T_{[\Omega]}\mathcal{M}$ can be obtained from a tangent vector $\xi_\Omega \in T_\Omega\mathcal{M}_p$ by projection. It can be shown that the horizontal space at $\mathcal{M}_p \ni (\boldsymbol{W}, \boldsymbol{M}) = (\boldsymbol{U}[\mathbf{I}_p,\ \mathbf{0}_{p,n-p}]^T, \boldsymbol{M})$ with $\boldsymbol{U} \in \mathcal{O}_n$ is the set (details in the supplementary material)

$$\left\{ \left(\boldsymbol{U} \begin{bmatrix} \boldsymbol{V}\boldsymbol{M}^{-1} - \boldsymbol{M}^{-1}\boldsymbol{V} \\ \boldsymbol{B} \end{bmatrix}, \boldsymbol{V} \right) \right\},$$

with $\boldsymbol{V} \in \mathrm{Sym}(p),\ \boldsymbol{B} \in \mathbb{R}^{(n-p)\times p}$. Furthermore, we have the following theorem to obtain the tangent vectors in $\mathcal{M}$.

Theorem 2 (Projecting on the Horizontal Space)**.** *For $(\xi_{\boldsymbol{W}}, \xi_{\boldsymbol{M}}) \in T_{(\boldsymbol{W},\boldsymbol{M})}\mathcal{M}_p$, the horizontal vector (*i.e.*, the associated tangent vector in $T_{[(\boldsymbol{W},\boldsymbol{M})]}\mathcal{M}$) is identified as*

$$\left(\xi_{\boldsymbol{W}} - \boldsymbol{W}\Theta, \xi_{\boldsymbol{M}} - \boldsymbol{M}\Theta + \Theta\boldsymbol{M}\right)\ , \tag{18}$$

with Θ the solution of the following Sylvester equation:

$$\Theta\boldsymbol{M}^2 + \boldsymbol{M}^2\Theta = \boldsymbol{M}\big(\xi_{\boldsymbol{W}}^T\boldsymbol{W} - \boldsymbol{W}^T\xi_{\boldsymbol{W}} + \boldsymbol{M}^{-1}\xi_{\boldsymbol{M}} - \xi_{\boldsymbol{M}}\boldsymbol{M}^{-1}\big)\boldsymbol{M}\ . \tag{19}$$

Proof. See the supplementary material. □

To perform Newton-type optimization on $\mathcal{M}$, we also need the form of the retraction $R_{[(\boldsymbol{W},\boldsymbol{M})]} : T_{[(\boldsymbol{W},\boldsymbol{M})]}\mathcal{M} \to \mathcal{M}$, which follows from the retraction on $\mathcal{M}_p$. In particular, we suggest the following retraction:

$$R_{[(\boldsymbol{W},\boldsymbol{M})]}(\xi_{\boldsymbol{W}}, \xi_{\boldsymbol{M}}) \triangleq \big(\mathrm{uf}(\boldsymbol{W} + \xi_{\boldsymbol{W}}), \boldsymbol{M}^{1/2}\,\mathrm{expm}(\boldsymbol{M}^{-1/2}\xi_{\boldsymbol{M}}\boldsymbol{M}^{-1/2})\boldsymbol{M}^{1/2}\big)\ . \tag{20}$$

Here $\mathrm{uf}(\boldsymbol{A}) = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1/2}$, which yields an orthogonal matrix and $\mathrm{expm}(\cdot)$ denotes the matrix exponential. Altogether, this provides us with the tools required to perform Riemannian optimization to solve our problem. The only missing mathematical entity is the Euclidean gradient of our loss function w.r.t. $\boldsymbol{W}$ and $\boldsymbol{M}$, which we provide in the supplementary material.

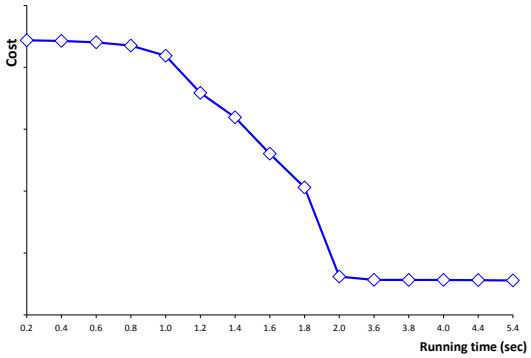

Figure 1: Convergence behavior of our algorithm.

In our experiments, we employed Conjugate Gradient descent on $\mathcal{M}$ to solve (12). In particular, we implemented the operations required for our manifold within the *manopt* Riemannian optimization toolbox (Boumal et al., 2014). The code is available at `https://sites.google.com/site/mehrtashharandi/`.

In Fig. 1, we illustrate the typical convergence behavior of our algorithm using the ASLAN dataset (Kliper-Gross et al., 2012). In our experiments, we have observed that the algorithm converges quite fast (typically in less than 25 iterations), thus making it scalable to learning large metrics.

3.2. Computational Complexity

The complexity of each iteration of our algorithm to solve (12) depends on the computational cost of the following major steps:

- Objective function evaluation. Computing $\mathcal{L}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X})$ takes $O(mnp + mp^2 + p^3 + np^2)$.
- Euclidean gradient evaluation: Computing $\nabla_{\boldsymbol{W}}$ takes $O(mn^2 + pn^2 + np^2)$, and computing $\nabla_{\boldsymbol{M}}$ takes $O(mn^2 + pn^2 + np^2 + p^3)$. Note that some computations are common to both $\nabla_{\boldsymbol{W}}$ and $\nabla_{\boldsymbol{M}}$. Hence the total flops for this step is less than the addition of the Stiefel and SPD parts.
- Projecting $(\nabla_{\boldsymbol{W}}, \nabla_{\boldsymbol{M}})$ to the tangent space of $\mathcal{M}_p$ takes $O(2p^2(n+p))$.
- Projecting a tangent vector in $\mathcal{M}_p$ costs $O(2np^2)$ to form the Sylvester equation and $O(p^3)$ to solve it using the Bartels - Stewart algorithm (Bartels & Stewart, 1972).
- Retraction: For the Stiefel part, the retraction τ_{St} takes $O(4np^2 + 11p^3)$. For the SPD part, τ_{SPD} takes $O(3p^3)$.

These steps are either linear or quadratic in n. Therefore, and as evidenced by our experiments, our approach can effectively and efficiently handle high-dimensional input features without any PCA pre-processing.

Remark 1. *The number of unknowns determined by our algorithm corresponds to the dimensionality of $\mathcal{M}_p$, that is, $np - \frac{1}{2}p(p+1) + \frac{1}{2}p(p+1) - \frac{1}{2}p(p-1) = \frac{1}{2}p(2n - p + 1)$. By contrast, the metric learning techniques that utilize PCA as a pre-processing step only determine $\frac{1}{2}p(p+1)$ unknowns, which is typically much smaller. As such, our method can potentially better leverage large amounts of training triplets. In our Big Data era, we believe this to be an important strength of our approach.*

3.3. Discussion

An SPD matrix $\boldsymbol{M} \in \mathcal{S}_{++}^p$ can be decomposed as $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ with $\boldsymbol{U} \in \mathcal{O}_p$ and $\boldsymbol{D}$ a diagonal matrix with positive elements. As such, the term $\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T$ appearing in our loss $\mathcal{L}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X})$ can be written as

$$\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T = \boldsymbol{W}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{W}^T = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T \ ,$$

with $\mathrm{St}(p,n) \ni \boldsymbol{V} = \boldsymbol{W}\boldsymbol{U}$. Thus, our optimization problem can be expressed by a loss $\mathcal{L}(\boldsymbol{V}, \boldsymbol{D}|\mathbb{X})$ with a search space defined as $\mathrm{St}(p,n) \times \mathbb{R}_+^p$. Theoretically, this representation has the same expressive power as our formulation if we ignore the invariance of $\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$ to permutations. However, in the context of fixed-rank matrix factorization (see (Mishra et al., 2014), Section 3.2), it has been shown that, for a parametrization of the form $\boldsymbol{W}\boldsymbol{M}\boldsymbol{V}^T$, where $\boldsymbol{W}, \boldsymbol{V} \in \mathrm{St}(p,n)$, modeling $\boldsymbol{M}$ as an SPD matrix is typically more effective than as a diagonal matrix with positive elements. The argument there is that it "gives more flexibility to optimization algorithms" (Mishra et al., 2014).

One can also factorize $\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T$ as $\boldsymbol{L}\boldsymbol{L}^T$ with $\boldsymbol{L} \in \mathbb{R}^{n\times p}$. This factorization, though being widely used, is not invariant to the action of $\mathcal{O}_p$, meaning that replacing $\boldsymbol{L} \rightarrow \boldsymbol{L}\boldsymbol{R}, \boldsymbol{R} \in \mathcal{O}_p$ will not change the loss. Such an invariance hinders gradient descent algorithms, as shown for example in (Journée et al., 2010; Mishra et al., 2014). In Section 6, we empirically show that this is indeed the case for the problem of interest here, *i.e.*, metric learning.

In (Journée et al., 2010), the invariance induced by the action of $\mathcal{O}_p$ in a factorization of the form $\boldsymbol{L}\boldsymbol{L}^T$ is taken into account. In particular, the authors make use of a quotient geometry to overcome the undesirable effects of the invariance in gradient descent optimization. There is a subtle, yet important difference between our formulation and that of (Journée et al., 2010): Our approach can benefit from a factorization with redundancy, which is effective in practice. Furthermore, note that the geometry developed in our paper can also handle the case where a Mahalanobis metric is searched for (*i.e.*, without recasting the problem as a factorization problem), which is the case in techniques such as (Globerson & Roweis, 2005; Koestinger et al., 2012; Zadeh et al., 2016).

Method	p = 25	p = 150	p = 500	p = 1000
Euc-$\boldsymbol{L}\boldsymbol{L}^T$	0.597	0.600	0.599	0.601
Rim-$\boldsymbol{L}\boldsymbol{L}^T$	0.625	0.601	0.602	0.608
Rim-$\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$	0.622	0.624	0.615	0.610
$\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T$ (Ours)	**0.630**	**0.627**	**0.621**	**0.617**

Table 2: AUC for various geometries on ASLAN.

Before concluding this part, we contrast the aforementioned factorization for the experiment reported in Table 1. To this end, we replace the term $\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T$ in our loss with

1. $\boldsymbol{L}\boldsymbol{L}^T$, $\boldsymbol{L} \in \mathbb{R}^{n\times p}$, and optimize using Euclidean geometry. We call this solution *Euc-*$\boldsymbol{L}\boldsymbol{L}^T$.

2. $\boldsymbol{L}\boldsymbol{L}^T$, $\boldsymbol{L} \in \mathbb{R}^{n\times p}$, and optimize using the geometry developed in (Journée et al., 2010). We call this solution *Rim-*$\boldsymbol{L}\boldsymbol{L}^T$.

3. $\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$, $\boldsymbol{V} \in \mathrm{St}(p,n)$ and $\boldsymbol{D}$ a diagonal and positive matrix. We optimize using the geometry of the product manifold $\mathrm{St}(p,n) \times \mathbb{R}_+^p$. We call this solution *Rim-*$\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$.

Following the experiment shown in Table 1, we evaluate the AUC for various dimensionalities using the aforementioned geometries. The results are provided in Table 2. First, we note that the general practice, *i.e.*, using Euclidean geometry, is significantly outperformed by its Riemannian counterparts. The quotient geometry developed in (Journée et al., 2010) performs on par with our approach for low dimensionalities (*e.g.*, $p = 25$). However, for larger dimensionalities, our technique yields more accurate solutions, suggesting that the redundancy in the formulation plays an important role. The importance of the redundancy can also be noticed by comparing Rim-$\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$ against our solution. In terms of computation time, the diagonal form, *i.e.*, Rim-$\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$ yields only slightly faster runtimes. In the particular case of ASLAN, the training time for $p = 1000$ was reduced to 150s.

4. Kernelizing the Solution

We now show how our approach can handle nonlinearity in the data, as well as generalize to non-vectorial input data, such as linear subspaces, which have proven effective for video recognition (Turaga et al., 2011; Harandi et al., 2014; Jayasumana et al., 2015). Following common practice when converting a linear algorithm to a nonlinear one (*e.g.*, from PCA to kernel PCA), we make use of a mapping of the input data to a Reproducing Kernel Hilbert Space (RKHS). As shown below, the resulting algorithm then only depends on kernel values (*i.e.*, it does not explicitly depend on the mapping to RKHS). Since much progress has recently been made in developing positive def-

inite kernels for non-vectorial data (Harandi et al., 2014; Jayasumana et al., 2015; Vishwanathan et al., 2010), this makes our approach applicable to a much broader variety of input types.

Specifically, let $\phi : \mathcal{X} \rightarrow \mathcal{H}$ be a mapping from the input space $\mathcal{X}$ to an RKHS $\mathcal{H}$ with corresponding kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$. Following the same formalism as before, we can define a cost function of the form

$$\mathcal{L}_{\mathcal{H}}(\boldsymbol{W}, \boldsymbol{M}|\mathbb{X}) \triangleq \sum_{i,y_i=1} \log(1+\tilde{p}_i) + \sum_{i,y_i=0} \log(1+\tilde{p}_i^{-1}) + \lambda r(\boldsymbol{M}, \boldsymbol{M}_0), \tag{21}$$

with $\tilde{p}_i = \exp\left(\beta d^2_{\mathcal{H}}(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i)\right)$ and $d^2_{\mathcal{H}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\big)^T \boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T\big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)\big)$, with $\boldsymbol{M} \in \mathcal{S}^p_{++}$ and $\boldsymbol{W} \in \mathrm{St}(p, \dim(\mathcal{H}))$. Note that for universal kernel functions, such as the Gaussian kernel, $\dim(\mathcal{H}) \rightarrow \infty$. We therefore need a formulation where only the kernel function appears, and not ϕ explicitly. To this end, we exploit the representer theorem (Schölkopf et al., 2001), which states that the mapping $\boldsymbol{W}$ lies in the span of the training data, and can thus be expressed as $\boldsymbol{W} = \Phi(\boldsymbol{D})\boldsymbol{A}$. Here, $\Phi(\boldsymbol{D}) = (\phi(\boldsymbol{d}_1), \cdots, \phi(\boldsymbol{d}_l)) \in \mathbb{R}^{\dim(\mathcal{H}) \times l}$ is a matrix that stacks the representation of the l training samples in the feature space. In this formalism, the orthogonality constraint on $\boldsymbol{W}$ can be written as

$$\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{A}^T\Phi(\boldsymbol{D})^T\Phi(\boldsymbol{D})\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{D})\boldsymbol{A} = \mathbf{I}_p \ ,$$

where $\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{D}) \in \mathcal{S}^l_{++}$ is the kernel matrix with elements $[\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{D})]_{i,j} = k(\boldsymbol{d}_i, \boldsymbol{d}_j)$. Let us define $\mathrm{St}(p, l) \ni \boldsymbol{B} = \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{D})^{1/2}\boldsymbol{A}$, such that the orthogonality constraint becomes $\boldsymbol{B}^T\boldsymbol{B} = \mathbf{I}_p$. This lets us write

$$\begin{aligned} d^2_{\mathcal{H}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) &= \big(\phi(\boldsymbol{x}) - \phi(\tilde{\boldsymbol{x}})\big)^T \boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T\big(\phi(\boldsymbol{x}) - \phi(\tilde{\boldsymbol{x}})\big) \\ &= \big(\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{D}) - \boldsymbol{k}(\tilde{\boldsymbol{x}}, \boldsymbol{D})\big)^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{D})^{-\frac{1}{2}} \boldsymbol{B}\boldsymbol{M}\boldsymbol{B}^T \\ &\quad \times \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{D})^{-\frac{1}{2}} \big(\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{D}) - \boldsymbol{k}(\tilde{\boldsymbol{x}}, \boldsymbol{D})\big), \end{aligned} \tag{22}$$

where $\mathbb{R}^l \ni \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{D}) = \big(k(\boldsymbol{x}, \boldsymbol{d}_1), \cdots, k(\boldsymbol{x}, \boldsymbol{d}_l)\big)^T$. Thus, the cost defined in Eq. 21 can be rewritten as a function of $\boldsymbol{B}$, *i.e.*, $\mathcal{L}_{\mathcal{H}}(\boldsymbol{B}, \boldsymbol{M}|\mathbb{X})$, which, by taking $d^2_{\mathcal{H}}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ from Eq. 22, only depends on kernel values. Since this cost function has essentially the same form as the one derived in Section 3, and the variables $\boldsymbol{M}$ and $\boldsymbol{B}$ lie on the same types of manifold as those of Section 3, we can use the same optimization strategy as before.

5. Related Work

Metric learning is a well-studied problem whose origins can be traced back to the early eighties (*e.g.*, (Short & Fukunaga, 1981)). Here, we focus on the prime representatives that will be used as baselines in our experiments. For a more thorough study, we refer the reader to the recent book by (Bellet et al., 2015).

The idea of Neighborhood Component Analysis (NCA) (Goldberger et al., 2004) is to optimize the error of a stochastic nearest neighbor classifier in the space induced by the Mahalanobis metric. The Information-Theoretic Metric Learning (ITML) algorithm, proposed by (Davis et al., 2007), learns a Mahalanobis metric by exploiting a notion of margin between pairs of samples. More precisely, the algorithm searches for a Mahalanobis matrix satisfying two types of constraints: (i) an upper bound u on the distance between pairs of samples from the same class, *i.e.*, in our formalism, $d^2_M(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) \leq u$, $\forall i \mid y_i = 1$; (ii) a lower bound l on the distance between pairs of dissimilar samples, *i.e.*, $d^2_M(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i) \geq l$, $\forall i \mid y_i = 0$.

The Large Margin Nearest Neighbors (LMNN) of (Weinberger & Saul, 2009) introduces the notion of local margins for metric learning. In LMNN, learning the Mahalanobis metric is expressed as a convex optimization problem that encourages the k nearest neighbors of any training instance $\boldsymbol{x}_i$ to belong to the same class as $\boldsymbol{x}_i$, while keeping away instances of other classes.

Logistic Discriminant based Metric Learning (LDML) (Guillaumin et al., 2009) relies on a Mahalanobis distance-based sigmoid function to encode the likelihood that two samples belong to the same class. The metric is then learned by maximizing the likelihood of the sample pairs $(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i)$ that truly belong to the same class, *i.e.*, $y_i = 1$, while minimizing that of the sample pairs that do not, *i.e.*, $y_i = 0$.

While effective, all the above-mentioned techniques rely on PCA as a pre-processing step to remain tractable. By contrast, the efficient "Keep It Simple and Straightforward Metric" (KISSME) algorithm of (Koestinger et al., 2012) focuses on addressing large-scale problems. KISSME assumes that the similar and dissimilar pairs are generated from two independent Gaussian distributions. Computing the Mahalanobis metric then translates to maximizing a log-likelihood, which can be achieved in closed-form. As illustrated in Table 1, however, this algorithm requires PCA pre-processing to achieve accuracies comparable to the ones produced by the other algorithms. In the spirit of KISSME, Geometric Mean Metric Learning (GMML) (Zadeh et al., 2016) relies on the geodesic connecting two covariance matrices to identify the Mahalanobis metric.

While effective and quite efficient, the above-mentioned techniques usually rely on PCA as a pre-processing step to reduce the dimensionality of the data. As evidenced by our experiments, this pre-processing step is sub-optimal.

Figure 2: Examples from the ASLAN dataset.

Method	HoG		HoF		HnF	
	CRR	AUC	CRR	AUC	CRR	AUC
baseline (Kliper-Gross et al., 2012)	54.2%	0.56	54.0%	0.57	54.5%	0.58
NCA (Goldberger et al., 2004)	56.6%	0.59	57.1%	0.60	56.7%	0.60
ITML (Davis et al., 2007)	55.6%	0.58	53.9%	0.55	55.9%	0.58
LDML (Guillaumin et al., 2009)	57.3%	0.61	56.5%	0.60	58.0%	0.61
LMNN (Weinberger & Saul, 2009)	55.9%	0.59	53.5%	0.56	56.0%	0.59
KISSME (Koestinger et al., 2012)	55.2%	0.58	52.8%	0.54	55.7%	0.60
GMML (Zadeh et al., 2016)	55.6%	0.57	52.8%	0.53	55.8%	0.58
DRML	58.3%	0.63	59.1%	**0.64**	58.6%	**0.63**
kDRML	**59.7%**	**0.64**	**60.7%**	**0.64**	**59.2%**	**0.63**

Table 3: Average accuracy and AUC for the ASLAN dataset.

6. Experimental Evaluation

We now evaluate our algorithms (DRML and kDRML) and compare them with the representative baseline metric learning methods discussed above, *i.e.*, NCA (Goldberger et al., 2004) LMNN (Weinberger & Saul, 2009), ITML (Davis et al., 2007), LDML (Guillaumin et al., 2009), KISSME (Koestinger et al., 2012) and GMML (Zadeh et al., 2016), as well as with dataset-specific baselines mentioned below. Our experiments consist of two parts. First, we make use of benchmark datasets where the data can be represented in vector (Euclidean) form, and thus both DRML and kDRML are applicable. Second, we consider manifold-valued data where only kDRML applies.

In all our experiments, we followed the so-called *restricted protocol*. That is, the only information accessible to the algorithms is the similarity/dissimilarity labels of pairs of samples; the class labels of the samples are unknown. For all the methods, we report the results obtained with the best subspace dimension. Note that this means that not all methods use the same subspace dimension. However, it makes the comparison more fair, since it truly shows the full potential of the algorithms.

6.1. Experiments with Euclidean Data

ACTION SIMILARITY MATCHING.

As a first experiment, we considered the task of action similarity recognition using the ASLAN dataset (Kliper-Gross et al., 2012). The ASLAN dataset contains 3,697 human action clips collected from YouTube, spanning over 432 unique action categories (see Fig. 2). The sample distribution across the categories is highly uneven, with 116 classes possessing only one video clip. The benchmark protocol focuses on action similarity (same/not-same), rather than action classification, and testing is performed on previously-unseen actions.

The dataset comes with 10 predefined splits of the data, where each split consists of 5,400 training and 600 testing pairs of action videos. The ASLAN dataset also provides three different types of descriptors: Histogram of Oriented Gradients (HoG), Histogram of Optical Flow (HoF), and a composition of both (referred to as HnF). The videos are represented by spatiotemporal bags of features (Laptev et al., 2008) with a codebook of size 5,000. For kDRML, we used an RBF Gaussian kernel whose bandwidth was set using Jaakkola's heuristic (Jaakkola et al., 1999).

In Table 3, we report the classification accuracy and the Area Under the ROC Curve (AUC) of our algorithms and of the baselines. Here, we also include the results of the benchmark (Kliper-Gross et al., 2012), which provides us with a direct comparison of previously published results. Note that DRML and kDRML outperform all the other algorithms. In general, kDRML performs better than DRML.

To further evidence the benefits of jointly learning the low-dimensional projection and the metric, we performed the following experiment, using the HoG features. We fixed the matrix $\boldsymbol{W}$ to the subspace obtained by PCA, and learned the metric using our loss function. This resulted in a drop in accuracy of roughly 1%, *i.e.*, a CRR of 57.4%. This confirms our intuition that we can achieve better than PCA by jointly learning the subspace and the metric.

Remark 2. *In (Kliper-Gross et al., 2012), it was shown that other metrics (*e.g.*, the cosine similarity) could outperform the Euclidean distance (used here as a baseline). In principle, our framework can also be used to learn cosine similarities by generalizing the inner product $\langle a, b\rangle$ as $a^T\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T b$. Doing so, however, goes beyond the scope of this paper.*

PERSON RE-IDENTIFICATION.

For the task of person re-identification, we used the iLIDS dataset (Zheng et al., 2009). The dataset consists of 476 images of 119 pedestrians and was captured in an airport. The number of images for each person varies from 2 to 8. The dataset contains severe occlusions caused by people and baggage.

In our experiments, we adopted the single-shot protocol. That is, the dataset was randomly divided into two subsets, training and test, with 59 and 60 exclusive individuals, respectively. The random splitting was repeated 10 times. In each partition, one image from each individual

Figure 3: Examples from the iLIDS dataset (Zheng et al., 2009).

Method	r = 1	r = 5	r = 10	r = 20
NCA (Goldberger et al., 2004)	27.9%	52.0%	65.5%	80.7%
kLFDA-lin (Xiong et al., 2014)	32.3%	57.2%	70.0%	83.9%
kLFDA-Chi2 (Xiong et al., 2014)	36.5%	64.1%	76.5%	88.5%
ITML (Davis et al., 2007)	29.5%	50.3%	62.6%	76.4%
LDML (Guillaumin et al., 2009)	27.8%	53.2%	67.0%	82.5%
LMNN (Weinberger & Saul, 2009)	32.6%	56.2%	68.9%	83.0%
KISSME (Koestinger et al., 2012)	28.0%	54.2%	67.9%	81.6%
GMML (Zadeh et al., 2016)	31.1%	55.6%	68.5%	82.9%
DRML	32.0%	57.6%	71.6%	85.7%
kDRML	**39.2%**	**65.5%**	**77.6%**	**89.5%**

Table 4: CMC at rank 1, 5, 10 and 20 on the iLIDS dataset.

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. We used the features provided by the authors of (Xiong et al., 2014). These features describe each image using 16-bin histograms from the RGB, YUV and HSV color channels, as well as texture histograms based on Local Binary Patterns (Ojala et al., 2002) extracted from 6 non-overlapping horizontal bands. For the kernel-based solutions, *i.e.*, kDRML and kLFDA, we used the Chi-square kernel.

We report performance in terms of the Cumulative Match Characteristic (CMC) curves for different rank values r indicating that we search for a correct match among the r nearest neighbors. Table 4 compares our results with those of the baseline metric learning algorithms, as well as with kernel Local Fisher Discriminant Analysis (kLFDA) (Xiong et al., 2014), which represents the state-of-the-art on this dataset. Our kDRML method achieves the highest scores for all ranks. Note that kLFDA requires the subject identities during training, while the other methods, including ours, don't. Despite this, kDRML outperforms the state-of-the-art results of kLFDA-Chi2.

6.2. Experiments with Manifold-Valued Data

To illustrate the fact that our algorithm generalizes to non-vectorial input data, we utilized the Youtube Faces (YTF) dataset (Wolf et al., 2011) and represented each video as a point on a Grassmann manifold. The YTF dataset contains 3,425 videos of 1,595 subjects collected from the YouTube website. These videos depict large variations in pose, illumination and expression. To evaluate the performance of the algorithms, we followed the protocol suggested in (Wolf et al., 2011). Specifically, we used the 5,000 video pairs officially provided with the dataset, which are equally divided into 10 folds. Each fold contains 250 'same' and 250 'not-same' pairs. We used the provided LBP features

Method	Acc.	AUC	EER
baseline (Wolf et al., 2011)	65.4%	0.698	0.360
NCA (Goldberger et al., 2004)	63.3%	0.794	0.284
ITML (Davis et al., 2007)	62.8%	0.715	0.346
LDML (Guillaumin et al., 2009)	59.7%	0.631	0.411
LMNN (Weinberger & Saul, 2009)	57.8%	0.763	0.306
KISSME (Koestinger et al., 2012)	67.5%	0.778	0.301
GMML (Zadeh et al., 2016)	67.7%	0.738	0.328
kDRML	**71.9%**	**0.798**	**0.277**

Table 5: Average accuracy, AUC and EER on the YTF dataset (Wolf et al., 2011).

and modeled each video by a subspace of dimensionality 10 as described in (Wolf et al., 2011). As a result, each video was modeled as a point on the Grassmann manifold $\mathcal{G}(10, 1770)$, where 1,770 is the dimensionality of the LBP features. We used the projection kernel defined as

$$k_{\text{proj}}(\boldsymbol{S}_i, \boldsymbol{S}_j) = \|\boldsymbol{S}_i^T \boldsymbol{S}_j\|_F^2 \ .$$

While kDRML directly uses a kernel function, some baselines (*e.g.*, KISSME), do not. To still be able to report results for these baselines, we utilized kernel PCA, instead of PCA, to create their inputs.

Table 5 summarizes the performance of the metric learning techniques and the baseline (Wolf et al., 2011) using the same input, *i.e.*, subspaces of dimensionality 10. Here again, kDRML comfortably outperforms the other methods for all the error metrics. For example, the gap in accuracy between kDRML and its closest competitor, *i.e.*, KISSME, is more than 4%.

Remark 3. *Note that, as shown in (Feragen et al., 2015), an RBF kernel of the form* $\exp(-\sigma d_g^2(\cdot,\cdot))$ *with* d_g *being the geodesic distance on the Grassmann manifold is not a positive definite kernel. The projection kernel, however, has been shown to be positive definite (Hamm & Lee, 2008), which, ultimately, is all we require to make our algorithm applicable to manifold-valued data. Furthermore, this kernel has proven effective in a variety of applications (Hamm & Lee, 2008; Harandi et al., 2017).*

7. Conclusions and Future Work

In this paper, we have argued against treating dimensionality reduction as a pre-processing step to metric learning. We have therefore introduced a framework that learns a low-dimensional representation and a Mahalanobis metric in this space in a unified manner. We have shown that the resulting framework could be cast an optimization problem on the quotient space of the product space of two Riemannian manifolds with the orthogonal group. Our experiments have evidenced the benefits of our unified approach over state-of-the-art metric learning algorithms that rely on PCA as a pre-processing step. In the future, we plan to study the use of other cost functions within our unified framework, especially formulations based on the concept of large margin.

References

Absil, P-A, Mahony, Robert, and Sepulchre, Rodolphe. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Bartels, Richard H. and Stewart, GW. Solution of the matrix equation ax+ xb= c. *Communications of the ACM*, 15(9):820–826, 1972.

Bellet, Aurélien, Habrard, Amaury, and Sebban, Marc. *Metric Learning*. Morgan & Claypool Publishers, 2015.

Bohné, Julien, Ying, Yiming, Gentric, Stéphane, and Pontil, Massimiliano. Large margin local metric learning. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 679–694. Springer, 2014.

Boothby, William Munger. *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Gulf Professional Publishing, 2003.

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

Cunningham, John P and Ghahramani, Zoubin. Linear dimensionality reduction: Survey, insights, and generalizations. *JMLR*, 2015.

Davis, Jason V, Kulis, Brian, Jain, Prateek, Sra, Suvrit, and Dhillon, Inderjit S. Information-theoretic metric learning. In *Proc. Int. Conference on Machine Learning (ICML)*, pp. 209–216, 2007.

Edelman, Alan, Arias, Tomás A, and Smith, Steven T. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20 (2):303–353, 1998.

Feragen, Aasa, Lauze, Francois, and Hauberg, Soren. Geodesic exponential kernels: When curvature and linearity conflict. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Globerson, Amir and Roweis, Sam. Metric learning by collapsing classes. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 18, pp. 451–458, 2005.

Goldberger, Jacob, Roweis, Sam, Hinton, Geoff, and Salakhutdinov, Ruslan. Neighbourhood components analysis. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2004.

Guillaumin, Matthieu, Verbeek, Jakob, and Schmid, Cordelia. Is that you? metric learning approaches for face identification. In *Proc. Int. Conference on Computer Vision (ICCV)*, pp. 498–505, 2009.

Hamm, Jihun and Lee, Daniel D. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. Int. Conference on Machine Learning (ICML)*, pp. 376–383. ACM, 2008.

Harandi, Mehrtash, Salzmann, Mathieu, Jayasumana, Sadeep, Hartley, Richard, and Li, Hongdong. Expanding the family of Grassmannian kernels: An embedding perspective. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 408–423. Springer, 2014.

Harandi, Mehrtash, Salzmann, Mathieu, and Hartley, Richard. Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017.

Hoffman, Judy, Rodner, Erik, Donahue, Jeff, Kulis, Brian, and Saenko, Kate. Asymmetric and category invariant feature transformations for domain adaptation. *Int. Journal of Computer Vision*, 109(1-2):28–41, 2014.

Huang, Gary B, Ramesh, Manu, Berg, Tamara, and Learned-Miller, Erik. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

Jaakkola, Tommi, Diekhans, Mark, and Haussler, David. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pp. 149–158. AAAI Press, 1999.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., and Harandi, M. Kernel methods on Riemannian manifolds with Gaussian RBF kernels. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(12):2464–2477, 2015.

Journée, Michel, Bach, Francis, Absil, P-A, and Sepulchre, Rodolphe. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Kliper-Gross, Orit, Hassner, Tal, and Wolf, Lior. The action similarity labeling challenge. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(3):615–621, 2012.

Koestinger, Martin, Hirzer, Martin, Wohlhart, Paul, Roth, Peter M, and Bischof, Horst. Large scale metric learning from equivalence constraints. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2288–2295. IEEE, 2012.

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

Lee, John M. *Smooth manifolds*. Springer, 2003.

Lu, Jiwen, Zhou, Xiuzhuang, Tan, Yap-Pen, Shang, Yuanyuan, and Zhou, Jie. Neighborhood repulsed metric learning for kinship verification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36(2):331–345, 2014.

Mishra, B, Meyer, G, Bonnabel, S, and Sepulchre, R. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3-4): 591–621, 2014.

Ojala, Timo, Pietikäinen, Matti, and Mäenpää, Topi. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

Pennec, Xavier, Fillard, Pierre, and Ayache, Nicholas. A Riemannian framework for tensor computing. *Int. Journal of Computer Vision*, 66(1):41–66, 2006.

Schölkopf, Bernhard, Herbrich, Ralf, and Smola, Alex J. A generalized representer theorem. In *Computational learning theory*, pp. 416–426. Springer, 2001.

Short, Robert D and Fukunaga, Keinosuke. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27(5):622–627, 1981.

Turaga, P., Veeraraghavan, A., Srivastava, A., and Chellappa, R. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

Vishwanathan, S Vichy N, Schraudolph, Nicol N, Kondor, Risi, and Borgwardt, Karsten M. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.

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

Wolf, Lior, Hassner, Tal, and Maoz, Itay. Face recognition in unconstrained videos with matched background similarity. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 529–534, 2011.

Xiong, Fei, Gou, Mengran, Camps, Octavia, and Sznaier, Mario. Person re-identification using kernel-based metric learning methods. In *Proc. European Conference on Computer Vision (ECCV)*, pp. 1–16. Springer, 2014.

Zadeh, Pourya, Hosseini, Reshad, and Sra, Suvrit. Geometric mean metric learning. In *Proc. Int. Conference on Machine Learning (ICML)*, pp. 2464–2471, 2016.

Zheng, Wei-Shi, Gong, Shaogang, and Xiang, Tao. Associating groups of people. In *BMVC*, volume 2, pp. 6, 2009.