Distance Metric Learning by Optimization on the Stiefel Manifold

Ankita Shukla
ankitas@iiitd.ac.in
Saket Anand
anands@iiitd.ac.in

IIIT-Delhi
New Delhi, India

Abstract

Distance metric learning has proven to be very successful in various problem domains. Most techniques learn a global metric in the form of a $n \times n$ symmetric positive semidefinite (PSD) Mahalanobis distance matrix, which has $\mathcal{O}(n^2)$ unknowns. The PSD constraint makes solving the metric learning problem even harder making it computationally intractable for high dimensions. In this work, we propose a flexible formulation that can employ different regularization functions, while implicitly maintaining the positive semidefiniteness constraint. We achieve this by eigendecomposition of the rank p Mahalanobis distance matrix followed by a joint optimization on the Stiefel manifold $\mathcal{S}_{n,p}$ and the positive orthant $\mathbb{R}^p_+$. The resulting nonconvex optimization problem is solved by employing an alternating strategy. We use a recently proposed projection free approach for efficient optimization over the Stiefel manifold. Even though the problem is nonconvex, we empirically show competitive classification accuracy on UCI and USPS digits datasets.

1 Introduction

Distance metric learning has received a lot of attention in the last decade owing to its success in many application domains like computer vision, text analysis, information retrieval, classification and clustering. The default Euclidean distance equally weights each dimension in the input space and is often inadequate to capture the semantics of the data. Metric learning techniques use training examples to learn a distance function that is semantically consistent with the data. The most commonly used approach is to learn a Mahalanobis distance based metric characterized by a symmetric positive semidefinite (PSD) matrix. This popularity is mainly due to their simple formulations and ease of extensibility to nonlinear spaces via kernelization.

Learning the Mahalanobis distance amounts to learning a transformed input space that ensures that similar points are closer, while dissimilar points are farther apart. The notion of similarity and dissimilarity is based on the semantics of the application. Many popular techniques [7, 9, 11, 12] set up the metric learning problem in a constrained optimization framework. The imposed constraints capture the intuition that same class point pairs have small distances, while sample points from different classes have a large distance. The challenge in solving such problems is efficient projection on to the constraint space while maintaining the positive semidefiniteness of the Mahalanobis distance matrix.

Metric learning has been solved using conventional solvers like SDPT3 [19] or SeDuMi [18] for semidefinite programs (SDPs), but several specialized algorithms have been proposed to circumvent their high complexity ($\mathcal{O}(n^{6.5})$). These metric learning algorithms can broadly be classified into two categories: one that perform a projection onto $\mathcal{S}_+^n$, the set of all $n \times n$ positive semidefinite matrices and the other that take a projection free approach. These specialized algorithms scale better than generic SDP solvers, however the time and memory requirements often become prohibitive for large-scale data.

Techniques that rely on projection on to $\mathcal{S}_+^n$ like [12, 20, 22], usually require an eigen-decomposition or SVD in each iteration resulting in an additional cost of $\mathcal{O}(n^3)$. Projection free approaches like [7, 9, 11] use special regularization functions leading to updates that guarantee positive semidefiniteness. In this paper, we explore a projection free approach that permits the flexibility to use different regularization functions.

The remainder of the paper is organized as follows. In Section 2, we discuss recent related work in the area of metric learning. We introduce the notation and some preliminaries in Section 3 followed by the details of the proposed method in Section 4. The experimental results are presented in Section 5 and we conclude with a discussion and future directions in Section 6.

2 Related Work

Due to the large body of work on metric learning, a comprehensive survey of existing techniques is out of the scope of this paper. We therefore restrict the discussion in this section to important developments in metric learning that are relevant in the context of our work. For a more complete view of research in metric learning, we encourage the interested reader to see the recent survey articles [3, 4, 10].

A convex optimization based approach for metric learning was first proposed by Xing *et al.* in [22], which attempted to maximize the distance between dissimilar points while bounding the distance between similar points. The solution was attained by a modified gradient ascent algorithm that incorporated a projection on to the constraint sets. The projection on to $\mathcal{S}_+^n$ was done by clipping the negative eigenvalues at zero. Shalev-Shwartz *et al.* followed this work by introducing the Pseudo-metric Online Learning Algorithm (POLA) [16], which used rank one updates based on a single constraint at a time. Due to rank one updates, the semidefiniteness constraint only required the smallest eigenvalue to be determined as opposed to a full eigen-decomposition. Weinberger *et al.* combined the ideas of convex optimization and margin maximization with k-Nearest Neighbors (kNN) and proposed the Large Margin Nearest Neighbor (LMNN) method [20] aiming at improving performance of a kNN classifier. Neither of these algorithms used a regularization function in the objective function and relied on the margin maximization strategy to achieve good generalization. Additionally, the aforementioned methods also learned a full $n \times n$ matrix limiting their scalability to data in high-dimensional spaces.

More recent methods have focused on learning distance functions that can handle high-dimensional input spaces. In [23], the authors add a regularization function to the convex objective that biases the solution such that similar points in a small neighborhood lie on a low-dimensional manifold. Law *et al.* [12] also use a regularization function that elegantly incorporates explicit control over the rank of the learned Mahalanobis matrix in the objective function. An ADMM based algorithm is proposed to learn the low-rank metric. A Frobenius norm based regularization function is proposed in [17], where the authors show that solving the dual is significantly simpler. In [13], the authors design a linearized ADMM algorithm for minimizing the trace norm regularized metric learning problem. All the methods rely on eigen-decomposition in each iteration to ensure positive semidefiniteness of the learned metric.

Mignon *et al.* [14] directly learn a linear transformation $\mathbf{L}$ by optimizing a generalized logistic loss function. Without a regularizer in the objective function, the technique requires an early stopping criterion, which is difficult to tune for each dataset [12]. Other methods [7, 9, 11] take a projection free approach and use special regularization functions like the $\log\det$ Bregman divergence, which implicitly maintains the positive semidefiniteness constraint and the rank of the learned Mahalanobis matrix.

Some recent work has used Riemannian geometry in context of learning similarity metrics. Hauberg *et al.* [15] propose a feature space that is modeled as a smooth manifold for which a Riemannian metric is learned as a smoothly varying linear combination of multiple *pre-learned* metric tensors. The learned feature space is guaranteed to be a metric space, which is then exploited to perform generalized PCA and regression. In [8], the author uses a framework for learning a non-symmetric and non-square similarity matrix. This approach learns a more general form of similarity metric that can be used for computing similarities between point pairs arising from different representations. The optimization problem is solved by the Riemannian trust-region method over the manifold of fixed rank matrices. In each iteration when the descent direction is accepted, a retraction operation [2] involving an SVD is performed.

In this paper, we exploit the Stiefel manifold $\mathcal{S}_{n,p}$ and perform a joint optimization over $\mathcal{S}_{n,p} \times \mathbb{R}^p_+$ to learn the p orthonormal eigenvectors and nonnegative eigenvalues that parametrizes the learned metric. We use a block coordinate descent like updates similar to [6] for optimizing over $\mathcal{S}_{n,p}$.

3 Preliminaries

In this section, we will introduce the notation and lay the necessary background for our proposed work. We denote the set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, with $\mathbf{x}_i \in \mathbb{R}^n, \quad i = 1, \ldots, m$ and their corresponding class labels by ℓ_i. The n-dimensional real space is denoted by $\mathbb{R}^n$ and its positive orthant as $\mathbb{R}^n_+$. The space of $n \times n$ symmetric positive semidefinite matrices is denoted by $\mathcal{S}^n_+$ and $\mathcal{S}_{n,p}$ represents the Stiefel manifold, *i.e.*, the space of $n \times p$ orthonormal matrices. As we use pairwise constraints for metric learning, the constraint point pairs are grouped into two sets: $\mathcal{C}_s$, the set of similar pairs and $\mathcal{C}_d$, the set of dissimilar pairs. The complete set of constraints is denoted by $\mathcal{C} = \mathcal{C}_s \cup \mathcal{C}_d$.

3.1 Mahalanobis Distance Learning

Given the points $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$, the Mahalanobis distance function between two data points is given by

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)} \tag{1}$$

where $\mathbf{M} \in \mathcal{S}^n_+$. The PSD constraint is necessary for this distance function to be a metric and satisfy the properties of nonnegativity $(d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) \geq 0)$, symmetry $(d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) = d_{\mathbf{M}}(\mathbf{x}_2, \mathbf{x}_1))$ and triangle inequality $(d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_3) \leq d_{\mathbf{M}}(\mathbf{x}_1, \mathbf{x}_2) + d_{\mathbf{M}}(\mathbf{x}_2, \mathbf{x}_3))$.

The task of Mahalanobis distance metric learning is to learn a suitable PSD matrix $\mathbf{M}$ that satisfies the distance constraints imposed on the training data. Different approaches use training data to model distance constraints using point pairs, triplets or quadruplets. While we use pairwise constraints, it is straightforward to adapt it to use triplets or quadruplets. The set of similar point pairs $\mathcal{C}_s = \{(i,j) : \ell_i = \ell_j\}$ can be generated by using data points having the same class label, while the set of dissimilar pairs $\mathcal{C}_d = \{(i,j) : \ell_i \neq \ell_j\}$ can be generated from points with different class labels. Note that the pairwise constraints do not necessarily need the class labels and can therefore be easily extracted from metadata *e.g*. text tags in images or documents.

A generic metric learning problem can be written as

$$\min_{\mathbf{M} \in \mathcal{S}_+^n} \quad \mathcal{L}(\mathbf{M}, \mathcal{C}) + \lambda R(\mathbf{M}) \tag{2}$$

where, $\mathcal{L}$ is the loss function penalizing the violated constraints, $R(\mathbf{M})$ is the regularization function used for smoothly learning $\mathbf{M}$ and λ is the regularization trade-off parameter. Our formulation for metric learning is a modified version of (2)

$$\min_{\mathbf{w} \in \mathbb{R}_+^p, \mathbf{U} \in \mathcal{S}_{n,p}} \quad \mathcal{L}(\mathbf{U}\mathbf{W}\mathbf{U}^\top, \mathcal{C}) + \lambda R(\mathbf{U}\mathbf{W}\mathbf{U}^\top) \tag{3}$$

where $\mathbf{W} = \mathrm{Diag}(\mathbf{w})$. We solve for $\mathbf{w}$ and $\mathbf{U}$ by alternately optimizing over $\mathbb{R}_+^n$ and the Stiefel manifold $\mathcal{S}_{n,p}$. The details of the algorithm are discussed in Section 4.2.

3.2 Optimization on Stiefel Manifold

The set of $n \times p$ orthornormal matrices has a Riemannian structure and is called the Stiefel manifold, $\mathcal{S}_{n,p} = \{\mathbf{U} \in \mathbb{R}^{n \times p} : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_p, n \geq p\}$ [8]. An alternate interpretation is that of a quotient space of the orthogonal group $\mathbf{O}_n = \{\mathbf{Q} \in \mathbb{R}^{n \times n} : \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_n\}$, *i.e*., $\mathcal{S}_{n,p} = \mathbf{O}_n / \mathbf{O}_{n-p}$. The tangent space at a point $\mathbf{U} \in \mathcal{S}_{n,p}$ is given by $T_{\mathbf{U}} = \{\Delta \in \mathbb{R}^{n \times p} : \Delta^\top \mathbf{U} = -\mathbf{U}^\top \Delta\}$.

An optimization problem of the form $\min_{\mathbf{U} \in \mathcal{S}_{n,p}} \mathcal{F}(\mathbf{U})$ can be solved by moving along the manifold in a direction that decreases the value of the objective function. Various optimization algorithms like conjugate gradient and Newton's method have been adapted that account for the underlying geometry of the Stiefel manifold to maintain orthogonality constraints during iterations[2, 8]. However, most existing algorithms for manifold optimization either perform re-orthogonalization by SVD like matrix factorization or move along the geodesics which usually require computation of matrix exponentials.

Wen and Jin [21] proposed an efficient constraint preserving update on the Stiefel manifold based on the Cayley transformation. The key idea is to relax the constraint of moving along geodesics and use retraction [2] to smoothly map a tangent vector to manifold. For a given point $\mathbf{U} \in \mathcal{S}_{n,p}$, let $\mathbf{G}$ be the gradient of $\mathcal{F}(\mathbf{U})$. A skew symmetric matrix $\mathbf{A} = \mathbf{G}\mathbf{U}^\top - \mathbf{U}\mathbf{G}^\top$ is then defined to get the following update in closed form [21]

$$\mathbf{V}(\tau) = \mathbf{Q}\mathbf{X} \quad , \qquad \text{where } \mathbf{Q} := \left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1} \left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}\right). \tag{4}$$

Since we seek fast updates on the Stiefel manifold, we resort to this update scheme in designing our metric learning algorithm.

4 Proposed Framework

Our formulation for metric learning is based on the premise that PSD matrices have non-negative eigenvalues and orthogonal eigenvectors. Thus we work with the representation of the rank p Mahalanobis matrix obtained by its eigendecomposition, $\mathbf{M}_{n\times n} = \mathbf{U}\mathbf{W}\mathbf{U}^\top, \ \mathbf{W} = \mathrm{Diag}(\mathbf{w})$, where $\mathbf{U} \in \mathcal{S}_{n,p}$ is the orthonormal matrix of eigenvectors, and $\mathbf{w} \in \mathbb{R}^p_+$ is the vector of eigenvalues. We rewrite the metric learning problem as a joint optimization over $\mathcal{S}_{n,p} \times \mathbb{R}^p_+$ and use $||\mathbf{w}||_2^2$ as the regularization function, which is equivalent to $||\mathbf{M}||_F^2$, the squared Frobenius norm of $\mathbf{M}$.

4.1 Problem Formulation

The convex metric learning problem with the Frobenius norm regularizer is

$$\begin{aligned} \min_{\mathbf{M}\in\mathcal{S}^n_+} \quad & ||\mathbf{M}-\mathbf{M}_0||_F^2 \\ \text{subject to} \quad & \mathbf{z}_{ij}^\top \mathbf{M}\mathbf{z}_{ij} \leq s, \qquad \forall\, i,j \in \mathcal{C}_s \\ & \mathbf{z}_{ij}^\top \mathbf{M}\mathbf{z}_{ij} \geq d, \qquad \forall\, i,j \in \mathcal{C}_d \end{aligned} \tag{5}$$

where the vectors $\mathbf{z}_{ij}$ are the difference vectors $\mathbf{x}_i - \mathbf{x}_j$ obtained from the constraint pairs in $\mathcal{C}$, s and d are the desired distances for constraints in $\mathcal{C}_s$ and $\mathcal{C}_d$ respectively. $\mathbf{M}_0$ is the initial Mahalanobis distance matrix, often initialized to identity or the data covariance matrix. Since the problem in (5) could be infeasible, we introduce slack variables ξ and rewrite the relaxed problem as

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^p_+,\mathbf{U}\in\mathcal{S}_{n,p},\xi\in\mathbb{R}^{|\mathcal{C}|}} \quad & ||\mathbf{w}-\mathbf{w}_0||_2^2 + \gamma||\xi-\xi_0||_2^2 \\ \text{subject to} \quad & \mathbf{z}_{ij}^\top \mathbf{U}\,\mathrm{Diag}(\mathbf{w})\mathbf{U}^\top \mathbf{z}_{ij} \leq \xi_{ij}, \qquad \forall\, i,j \in \mathcal{C}_s \\ & \mathbf{z}_{ij}^\top \mathbf{U}\,\mathrm{Diag}(\mathbf{w})\mathbf{U}^\top \mathbf{z}_{ij} \geq \xi_{ij}, \qquad \forall\, i,j \in \mathcal{C}_d \end{aligned} \tag{6}$$

where $\mathbf{w}_0$ is the vector of eigenvalues of $\mathbf{M}_0$. The initial vector of slack variables ξ_0 of length $|\mathcal{C}|$ takes values $(\xi_0)_{ij} = \{s,d\}$ based on whether $i,j \in \mathcal{C}_s$ or $i,j \in \mathcal{C}_d$. Note that the problem becomes nonconvex because of the domain of $\mathbf{U}$, which is the Stiefel manifold. The solution to the problem (6) yields $\widehat{\mathbf{w}}$ and $\widehat{\mathbf{U}}$, which are used to reconstruct the Mahalanobis matrix $\widehat{\mathbf{M}} = \widehat{\mathbf{U}}\mathrm{Diag}(\widehat{\mathbf{w}})\widehat{\mathbf{U}}^\top$.

Intuitively, the solution to (6) gives an orthogonal basis $\widehat{\mathbf{U}}$ of the p-dimensional subspace of $\mathbb{R}^n$, along which minimal scaling is required to satisfy the distance constraints. While we cannot theoretically guarantee good generalization, our experiments in Section 5 show that results are competitive with metric learned by solving (5). It should be noted that our formulation (6) can be conveniently extended to kernel spaces by modifying the difference vector $\mathbf{z}_{ij} = \mathbf{e}_i - \mathbf{e}_j$ and representing the positive semidefinite kernel matrix $\mathbf{K} = \mathbf{U}\mathrm{Diag}(\mathbf{w})\mathbf{U}^\top$.

4.2 Algorithm

We solve the problem developed in (6) jointly over $\mathcal{S}_{n,p} \times \mathbb{R}^p_+$ by taking an alternating minimization approach. We initialize the algorithm with the Euclidean metric in a p-dimensional space with $\mathbf{w}_0$ as a vector of ones and $\mathbf{U}_0$ as a randomly picked point on $\mathcal{S}_{n,p}$.

Keeping $\mathbf{U}$ fixed, we solve the following constrained least squares problem to update $\mathbf{w}$

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & ||\mathbf{w}-\mathbf{w}_0||_2^2 + \gamma||\xi - \xi_0||_2^2 \\ \text{subject to} \quad & \mathbf{z}_{ij}^\top \mathbf{U}\,\mathrm{Diag}(\mathbf{w})\mathbf{U}^\top \mathbf{z}_{ij} \leq \xi_{ij}, \qquad \forall\, i,j \in \mathcal{C}_s \\ & \mathbf{z}_{ij}^\top \mathbf{U}\,\mathrm{Diag}(\mathbf{w})\mathbf{U}^\top \mathbf{z}_{ij} \geq \xi_{ij}, \qquad \forall\, i,j \in \mathcal{C}_d \\ & \mathbf{w} \geq \mathbf{0},\ \xi \geq \mathbf{0} \end{aligned} \tag{7}$$

At the t^{th} iteration, the KKT conditions yield the following updates for a single constraint $(i,j) \in \mathcal{C}$

$$\begin{aligned} \lambda_{ij}^t &= \operatorname*{argmin}_{\lambda_{ij}^t} \quad \lambda_{ij}^t\, y_{ij}\, (\mathbf{z}_{ij}^\top \mathbf{U}^{t-1}\mathrm{Diag}(\mathbf{w}^{t-1})\mathbf{U}^{t-1\top}\mathbf{z}_i - \xi_{ij}^{t-1}) \\ \mathbf{w}^t &= \operatorname*{argmin}_{\mathbf{w}^t} \quad ||\mathbf{w}^t - \mathbf{w}^{t-1}||_2^2 + \lambda_{ij}^t\, y_{ij}(\mathbf{z}_{ij}^\top \mathbf{U}^{t-1}\mathrm{Diag}(\mathbf{w}^t)\mathbf{U}^{t-1\top}\mathbf{z}_{ij} - \xi_{ij}^{t-1}) \\ \xi_{ij}^t &= \operatorname*{argmin}_{\xi_{ij}^t} \quad -\xi_{ij}^t + \gamma||\xi^t - \xi^{t-1}||_2^2 \end{aligned} \tag{8}$$

where $y_{ij} = -1$, if $i,j \in \mathcal{C}_s$ and $y_{ij} = 1$ if $i,j \in \mathcal{C}_d$ and $\lambda_{ij} \geq 0$ are the Lagrange multipliers. With an updated $\mathbf{w}$, we then solve for $\mathbf{U}$ for the same constraint pair (i,j). This is achieved by solving the following problem over the Stiefel manifold using updates in (4)

$$\min_{\mathbf{U}^t \in \mathcal{S}_{n,p}} \quad \lambda_{ij}^t\, y_{ij}(\mathbf{z}_{ij}^\top \mathbf{U}^t \mathrm{Diag}(\mathbf{w}^t)\mathbf{U}^{t\top}\mathbf{z}_{ij} - \xi_{ij}^t). \tag{9}$$

We pick another constraint and repeat the updates (8) and (9) till convergence.

Since p could be as large as n, and the updates (4) require inversion of a $2p \times 2p$ matrix [21], we use a block coordinate descent like strategy proposed in [6] to speed up this step. The key idea in [6] is to parametrize $\mathbf{U}$ by a point on a smaller Stiefel manifold. To obtain this parametrization, a set of $k \leq n$ rows $\mathcal{K}$, is selected from $\mathbf{U}$ to construct a smaller matrix $\mathbf{H}_{k\times p}$. If $\mathcal{I}$ is the set of linearly independent columns of $\mathbf{H}$, the parametrization is given as [6]

$$\mathbf{U}(\mathbf{V}) = \begin{bmatrix} \mathbf{V}\mathbf{P}^{1/2} & \mathbf{V}\mathbf{P}^{1/2}\mathbf{R} \\ \mathbf{U}_{\bar{\mathcal{K}},\mathcal{I}} & \mathbf{U}_{\bar{\mathcal{K}},\bar{\mathcal{I}}} \end{bmatrix} \tag{10}$$

where $\mathbf{P} = \mathbf{H}_{\cdot,\mathcal{I}}^\top \mathbf{H}_{\cdot,\mathcal{I}}$ is positive definite, the $\bar{\mathcal{K}}$ and $\bar{\mathcal{I}}$ denote the complementary sets of $\mathcal{K}$ and $\mathcal{I}$ respectively. The matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{I}|\times|\bar{\mathcal{I}}|}$ is the linear transformation that maps $\mathbf{H}_{\cdot,\mathcal{I}}$ to $\mathbf{H}_{\cdot,\bar{\mathcal{I}}}$ and the orthonormal matrix $\mathbf{V}$ is a point on the smaller Stiefel manifold $\mathcal{S}_{k,|\mathcal{I}|}$. Collins *et al.* [6] show that a descent curve on $\mathcal{S}_{k,|\mathcal{I}|}$ gets mapped to the original manifold $\mathcal{S}_{n,p}$ by (10) in a direction of descent. As each block of k rows is updated on a smaller Stiefel manifold, we get efficient updates for $\mathbf{U}$. Moreover, this block coordinate descent type strategy can be parallelized by using disjoint sets of rows $\mathcal{K}_i$ such that $|\cup_i \mathcal{K}_i| \leq n$. We summarize our overall metric learning algorithm in Algorithm 1.

5 Experiments

We evaluated the proposed learning algorithm called Stiefel Manifold based Metric Learning (SMML) on the UCI benchmark data sets and USPS digits. The learned metric is evaluated

Algorithm 1 Metric Learning Algorithm on $S_{k,p}$ (SMML)

```
Input: $\mathcal{C}_s$ :set of similar pair, $\mathcal{C}_d$ : set of dissimilar pairs, $s$, $d$: distance thresholds, $\gamma$:slack parameter, $K$ ; number of rows
Output: $\mathbf{M}$: Learned Distance Matrix

  1. Initialize $\mathbf{U}$ and $\mathbf{w}$
     $\mathbf{w}^0$: $\mathbf{1}_{p\times 1}$ (vector of ones)
     $\mathbf{U}^0$: Random Orthogonal matrix

  2. Initialize slack variables
     $\xi^0_{i,j} = s$ for $(i,j) \in \mathcal{C}_s$ and $\xi^0_{i,j} = d$ for $(i,j) \in \mathcal{C}_d$

  3. repeat for $t = 1, \dots$
      (a) Pick a constraint (i,j) $\in \mathcal{C}_s$ or $\mathcal{C}_d$
      (b) Compute $\lambda^t_{i,j}$, $\xi^t_{ij}$ and $\mathbf{w}^t$ using (8)
      (c) Update $\mathbf{U}^t \in S_{n,p}$
            i. Select $k$ rows from $(1,2,....n)$
           ii. Construct $\mathbf{U}^t_{\mathcal{K},\cdot}$ : $k \times p \subseteq \mathbf{U}^t$
          iii. Find $\mathcal{I}$, the set of linearly independent columns in $\mathbf{U}^t_{\mathcal{K},\cdot}$
           iv. Compute $\mathbf{V}$ on $S_{k,|\mathcal{I}|}$ for an appropriate $\tau$ (4)
            v. Update $\mathbf{U}^t$ with the modified $k$ rows based on $\mathbf{V}$ (10)

  4. until convergence

return $\widehat{\mathbf{M}} = \widehat{\mathbf{U}}\ \mathrm{Diag}(\widehat{\mathbf{w}})\widehat{\mathbf{U}}^\top$
```

and compared against the Euclidean distance metric in terms of classification accuracy of a 3-nearest neighbor classifier. We compared the running time of SMML (Algorithm 1) to solve (6) with that of SeDuMi [18] to solve the relaxed version of (5). The experiments ran on a laptop with a core i7 quad core processor and 8 GB RAM with only two cores enabled. The threshold values s and d in (6) for similarity and dissimilarity constraints are set to the 1^{st} and 99^{th} percentile of all pairwise distances.

UCI datasets: We compared running time and accuracy of three UCI data set of varying dimension with the convex counterpart. For high dimensional data, we optimize simultaneously over multiple $\mathcal{S}_{k,p}$ by selecting disjoint sets of k rows, whereas a sequential approach is used in case of low dimension data to avoid communication overheads between parallel threads. The optimal choice of k is found heuristically.

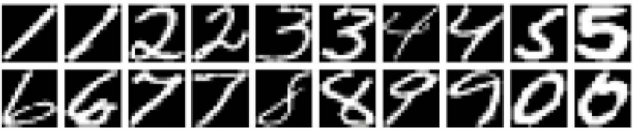

Figure 1: USPS digits

USPS digits [1]: The dataset consists of 16×16 grayscale images with 1100 images for each digit. A few sample images from the dataset are shown in Fig. 1. The images are represented as 256 dimensional vector formed by concatenating the columns of image. A set of 15 labeled points from each class are selected to create $2 \times \binom{15}{2}$ similarity and dissimilarity constraints. The results for both, the UCI datasets and USPS digits is summarized in Table 1.

We used PCA to reduce the dimensionality of the data with 99%, 95% and 90% energy. While the results for convex formulation and our proposed method are same for lower dimension representation with improvement in computation time. However, in case of higher dimensions, the learning with SeDuMi solver becomes computationally expensive in terms of memory usage with impractical run times. The accuracy and running time comparisons are summarized in Table 2.

Table 1: Classification Accuracy and Run time Comparison Results

	USPS	Wine	Inosphere	Haberman's Survival
# samples	11000	178	351	306
# constraints ($\|\mathcal{C}\|$)	900	630	900	900
# dimension	256	13	34	3
# dimension after PCA	114			
# Training points	150	45	30	30
# Testing points	2000	133	148	276
# classes	10	3	2	2
$\|\mathcal{K}\|$	24	5	8	2
Classification Accuracy %				
Euclidean	76.10	72.3	69.4	58
CVX	93.7	94.6	98	96.3
SMML	93.1	95	97.3	98
Computational Time(in secs)				
CVX	846.3	7.2	22.6	2.3
SMML	346	13.6	39.2	7.8

Table 2: Comparison Results: USPS digits for different dimension from PCA

	Run time(in mins)		Accuracy(%)	
PCA dimension, $\|\mathcal{K}\|$	CVX	SMML	Euclidean	SMML/CVX
38,10	10	8.7	76	82 /83
66,22	54	17	76.7	89.4/80
152,30	-	39	79	93.7/-

6 Conclusion

We proposed a metric learning formulation that poses a joint optimization problem over $\mathcal{S}_{n,p} \times \mathbb{R}_+^p$ to find the eigenvectors and eigenvalues of the learned Mahalanobis distance matrix $\mathbf{M}$. The objective function used was the sum of squared eigenvalues, which is equivalent to the squared Frobenius norm of $\mathbf{M}$. This formulation allows the flexibility to replace the regularizer with any convex spectral function of $\mathbf{M}$ without significant changes in the solving strategy.

We took an alternate minimization approach by iteratively updating the eigenvalues and the eigenvectors of $\mathbf{M}$ to solve the ensuing nonconvex problem. We compared the proposed method with the convex counterpart and showed competitive performance in classification tasks. While these experiments show encouraging results with respect to accuracy and running time, the inherent nonconvexity of the problem demands a deeper theoretical analysis that will be pursued in the future. Finally, we plan to explore the impact of other regularizers, especially convex spectral functions like $\log\det$ or Burg entropy.

7 Acknowledgments

The authors would like to acknowledge Peter Meer, as this idea spawned when the second author was at the Robust Image Understanding Lab, Rutgers University.

References

[1] USPS digits data set. http://cs.nyu.edu/âĹijroweis/data.html.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[3] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[4] Aurélien Bellet, Amaury Habrard, and MarcSebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015.

[5] Li Cheng. Riemannian similarity learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML*, pages 540–548, 2013.

[6] Maxwell D Collins, Ji Liu, Jia Xu, Lopamudra Mukherjee, and Vikas Singh. Spectral clustering with a convex regularizer on millions of images. In *Computer Vision–ECCV 2014*, pages 282–298. Springer, 2014.

[7] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

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

[9] Prateek Jain, Brian Kulis, Jason V Davis, and Inderjit S Dhillon. Metric and kernel learning using a linear transformation. *The Journal of Machine Learning Research*, 13(1):519–547, 2012.

[10] Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4): 287–364, 2012.

[11] Brian Kulis, Mátyás A Sustik, and Inderjit S Dhillon. Low-rank kernel learning with bregman matrix divergences. *The Journal of Machine Learning Research*, 10:341–376, 2009.

[12] M.T. Law, N. Thome, and M. Cord. Fantope regularization in metric learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1051–1058, June 2014.

[13] Wei Liu, Cun Mu, Rongrong Ji, Shiqian Ma, John R. Smith, and Shih-Fu Chang. Low-rank similarity metric learning in high dimensions. In *AAAI Conference on Artificial Intelligence (AAAI)*, Austin, Texas, USA, 2015.

[14] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2666–2672, June 2012.

[15] Søren Hauberg, Oren Freifeld, and Michael J. Black. A geometric take on metric learning. In *Advances in Neural Information Processing Systems NIPS*, pages 2024–2032. 2012.

[16] Shai Shalev-Shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudo-metrics. In *ICML*, pages 94–, 2004.

[17] Chunhua Shen, Junae Kim, Fayao Liu, Lei Wang, and A. van den Hengel. Efficient dual approach to distance metric learning. *Neural Networks and Learning Systems, IEEE Transactions on*, 25: 394–406, 2014.

[18] Jos F. Sturm. Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[19] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95:189–217, 2003.

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

[21] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[22] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems NIPS*, pages 521–528. 2002.

[23] Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu. Low rank metric learning with manifold regularization. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 1266–1271, 2011.