GSML: A Unified Framework for Sparse Metric Learning

Kaizhu Huang
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China 100190
kzhuang@nlpr.ia.ac.cn

Yiming Ying, Colin Campbell
Department of Engineering Mathematics
University of Bristol
Bristol BS8 1TR, United Kingdom
{enxyy,C.Campbell}@bris.ac.uk

Abstract

There has been significant recent interest in sparse metric learning (SML) in which we simultaneously learn both a good distance metric and a low-dimensional representation. Unfortunately, the performance of existing sparse metric learning approaches is usually limited because the authors assumed certain problem relaxations or they target the SML objective indirectly. In this paper, we propose a Generalized Sparse Metric Learning method (GSML). This novel framework offers a unified view for understanding many of the popular sparse metric learning algorithms including the Sparse Metric Learning framework proposed in [15], the Large Margin Nearest Neighbor (LMNN) [21][22], and the D-ranking Vector Machine (D-ranking VM) [14]. Moreover, GSML also establishes a close relationship with the Pairwise Support Vector Machine [20]. Furthermore, the proposed framework is capable of extending many current non-sparse metric learning models such as Relevant Vector Machine (RCA) [4] and a state-of-the-art method proposed in [23] into their sparse versions. We present the detailed framework, provide theoretical justifications, build various connections with other models, and propose a practical iterative optimization method, making the framework both theoretically important and practically scalable for medium or large datasets. A series of experiments show that the proposed approach can outperform previous methods in terms of both test accuracy and dimension reduction, on six real-world benchmark datasets.

1 Introduction

Metric learning is an important concept in machine learning and data mining. Its objective is to learn a proper distance metric so as to improve prediction accuracy in supervised learning or benefit clustering performance in unsupervised learning. To this end, a number of approaches have been proposed, e.g. [5, 8, 19, 23].

Despite of its success, metric learning usually attempts to learn a distance function f with a full square transformation matrix $A^\top$ from the given data set $\mathcal{S} = \{\mathbf{x}_k \in \mathbb{R}^D | k = 1, \dots N, N \in \mathbb{N}\}$ (i.e., the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as $f(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top AA^\top (\mathbf{x}_i - \mathbf{x}_j)$), while satisfying certain extra constraints; these constraints can force similar (dissimilar) points to stay closer (further apart) with the new distance metric. However, learning a full matrix may cause some problems. Additionally, the presented data are often contaminated by noise, especially for high-dimensional datasets. A full matrix cannot suppress such noise and may limit accuracy as a consequence. On the other hand, a full matrix, which is inefficient to compute, would result in practical difficulties for many real applications.

In order to solve the above shortcomings involved in metric learning, many researchers proposed to learn a *sparse* metric from different perspectives. These work includes the Sparse Metric Learning (SML) framework [15], the Large Margin Nearest Neighbor (LMNN) [21, 22], and the D-ranking Vector Machine (D-ranking VM) [14]. All these methods are able to learn a good distance metric as well as a sparse or low-dimensional representation. Specifically, Rosales and Fung targeted the shortcomings of metric learning directly [15]. Their method has demonstrated considerable improvement over traditional metric learning approaches [15]. LMNN is motivated from the large margin concept and its solution can also lead to sparse metrics. Although not designed for sparse learning, D-ranking VM is shown to force a low-rank (usually rank one) on the metric. However, there are some limitations to these models. Although SML, proposed in [15], started from a reasonable motivation for achieving sparsity as well as good distance metric, they relaxed their model and did not lead to the optimal solution. LMNN touches the sparse concept marginally, which is shown not to hit the target of sparse metric learning directly. D-ranking VM made a strong restriction on the rank of the distance matrix, and hence may not be flexible for real application.

In contrast to previous sparse metric learning approaches, in this paper, we propose a general framework called Generalized Sparse Metric Learning (GSML) along with physical interpretation and practical optimization. This novel framework not only offers a unified view for understanding many of the popular sparse metric learning algorithms including the above three methods, but also it leads to an optimal model which generalizes the SML approach proposed in [15]. Specifically, the proposed GSML can learn both a good distance as well as the optimal low-dimensional representation.[1] Both theoretical justification and empirical verification demonstrate that the proposed GSML can outperform previous methods with many real-life datasets.

The proposed unified framework is important because it builds various connections with other existing models. Besides the above mentioned three models, GSML is shown to have a close connection with the Pairwise Support Vector Machines (PSVM) [20]. In addition, we also demonstrate that many existing metric learning approaches such as RCA and the method proposed by Xing et al [23] can readily be extended to their sparse versions based on our novel framework. Another appealing feature is that a simple iterative optimization algorithm can be applied to solve the original convex but computationally difficult Semi-Definite Programming (SDP) problem involved in the proposed framework, making the proposed model scalable for medium or large datasets.

In summary, we have proposed a unified framework that establishes various connections with many famous models. We present this unified picture in Figure 1. More details can be seen in Section 2.2 on how these connections can be built.

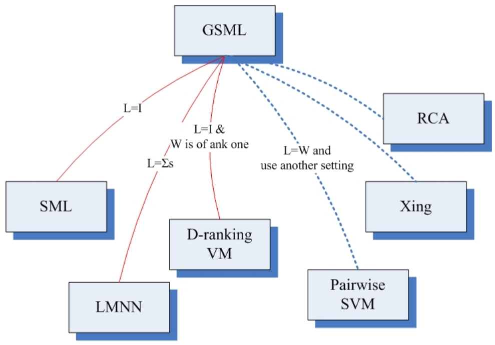

Figure 1. A unified framework establishing various connections with other models.

The paper is organized as follows. In the next section, we present our novel unified framework. The model definition, theoretical justification, connections with other approaches, and practical optimization will be discussed in turn in this section. In Section 3, we evaluate our algorithm and report experimental results. In Section 4, we discuss related work. Finally, we set out the conclusion with some remarks.

2 Unified Framework for Sparse Metric Learning

In this section, we present the general framework for sparse metric learning.

2.1 Notations and Model Definition

Assume that we are given a training data set $\mathcal{S}$ containing N D-dimensional data points $\mathbf{x}_k \in \mathbb{R}^D$, $k = 1, 2, \ldots, N$. Assuming A is a matrix, we denote its i-th row vector as A_i. $A \succeq 0$ means that the matrix A is a positive semi-definite matrix. Moreover, we denote the trace of matrix A by $\mathbf{tr}(A)$. For simplicity, let $\mathbf{O}^D$ be the set of all D by D orthonormal matrices i.e. columns of the matrices are orthonormal vectors.

The basic target of metric learning is to learn an appropriate distance metric f from $\mathcal{S}$ with extra constraints on a set of triplets $\mathcal{T} = \{(i,j,k) | f(\mathbf{x}_i, \mathbf{x}_j) \leq f(\mathbf{x}_i, \mathbf{x}_k)\}$ [15]. Such triplets provide a certain relative comparison among pairs, i.e. $\mathbf{x}_i$ is more similar to $\mathbf{x}_j$ than to $\mathbf{x}_k$.[2] In the context of sparse metric learning, f is assumed to be a linear transformation $A^\top : \mathbb{R}^D \to \mathbb{R}^d$ (with $d \ll D$ for obtaining sparsity) such that $\forall (i,j,k) \in \mathcal{T}, ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j||_2^2 \leq ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k||_2^2$, with $\hat{\mathbf{x}}_k = A^\top \mathbf{x}_k$.

The unified sparse metric learning framework is presented as follows:

$$\begin{aligned} \min_{W,\xi} \quad & \sum_t \xi_t + \gamma \mathbf{tr}(LAA^\top) \\ \text{s.t.} \quad & \forall (i,j,k) \in \mathcal{T}, ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j||_2^2 \leq ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k||_2^2 + \xi_t \quad (1) \\ & \forall t, \xi_t \geq 0 \,, \end{aligned}$$

where t indexes the set $\mathcal{T}$ and $\gamma \in \mathbb{R}$ is a positive trade-off parameter. And L is a pre-defined matrix, which could be the identity matrix, the covariance matrix, or some other specified matrix. We will show shortly that different choice of L can lead to different sparse metric learning models. Additionally, in order to avoid a trivial solution, the constraint (1) is often modified with an added margin, i.e., $||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j||_2^2 \leq ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k||_2^2 + \xi_t - 1$.

Denoting $W = AA^\top$, we can transform the above opti-

[1] It is noted that Principle Component Analysis or Fisher Discriminant Analysis can also lead to low-dimensional representation. However, they need to specify the dimension beforehand, and they are incapable of learning both a distance metric and the low-dimensional representation simultaneously.

[2] Other settings could be also used (e.g., a setting discussed in Section 2.2.4)

mization to the following equivalent form:

$$\min_{W,\xi} \quad \sum_t \xi_t + \gamma \mathbf{tr}(LW) \tag{2}$$

$$\text{s.t.} \quad \forall (i,j,k) \in \mathcal{T}, \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} \leq \mathbf{x}_{ik}^\top W \mathbf{x}_{ik} + \xi_t, \tag{3}$$

$$\forall t, \xi_t \geq 0 \ , \tag{4}$$

$$W \succeq 0 \ , \tag{5}$$

where $\mathbf{x}_{ij}$ is defined as $\mathbf{x}_i - \mathbf{x}_j$, and $\mathbf{x}_{ik}$ is similarly defined.

We interpret the unified model as follows. The basic motivation is that, a good choice of a distance metric should generally preserve the *distance structure* of the data: the distance between examples exhibiting *similarity* should be relatively smaller, in the transformed space, than between examples exhibiting *dissimilarity*. Such preservation is implied by the constraint (3). Moreover, for the purpose of generalization ability or noise suppression, the distance matrix defined by A should be regularized. This can be seen in the second term of (2). We will see why this term can be used as a regularized term for forcing the sparsity in Section 2.2.1. Finally, we can observe that the above model is a typical Semi-definite Programming (SDP) problem, which is convex but also well-known for its high computational complexity. In Section 2.3, we will show how to apply an iterative sub-gradient method to solve this difficult optimization problem.

2.2 General Framework for Sparse Metric Learning

We will demonstrate that the proposed general framework contains many existing sparse metric learning models as special cases. Importantly, we show that the model with $L = I$ corresponds to the optimal model that is able to achieve the goal of sparse metric learning directly. It also contains LMNN and D-ranking VM as special cases. In addition, it can also build a close connection with the Pairwise SVM model [20]. Finally, we demonstrate that many current non-sparse metric learning models including the method proposed in [23] and the Relevant Component Analysis (RCA) can readily be extended to their sparse versions by using our proposed framework.

2.2.1 Connection with Sparse Metric Learning [15]

The Sparse Metric Learning model proposed in [15] is presented as follows:

$$\min_{\xi,A} \sum_t \xi_t + \gamma \sum_{m=1}^{D} ||A_m||_1$$

$$\text{s.t.} \quad \forall (i,j,k) \in \mathcal{T}, ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j||_2^2 \leq ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k||_2^2 + \xi_t \ ,$$

$$\forall t, \xi_t \geq 0 \ ,$$

As a 1-norm can produce sparse solutions [11, 8], many columns of $A^\top$ would be expected to become zero vectors due to the existence of the second term in the above objective function. This is the basic motivation of SML.

However, there are several problems with the above model. Firstly, we observe that the low-dimensional mapping is directly performed in the original input space. More specifically, enforcing the l-th column in $A^\top$ to become a zero vector will naturally discard the l-th dimension in the input data. However, redundant features unnecessarily appear in the original input space. Thus certain transformed features may be redundant.[3] In other words, the useless features may be in the space $U^T\mathbf{x}_i, i = 1, 2, \ldots, N$, where U is an unknown $D \times D$ matrix. Secondly, the 1-norm regularization may present a problem. We would like to enforce some columns of $A^\top$ to be zero vectors. However, the 1-norm $\sum_{m=1}^{D} ||A_m||_1 = \sum_{m=1}^{D}\sum_{n=1}^{D} |A_{mn}|$ can force many elements of $A^\top$ to be zero but it does not necessarily force a whole column to be zero. Thirdly, in order to solve the above optimization problem, [15] further proposed to restrict the optimization of the matrix A in the space of diagonal dominance matrices. Although the final optimization is relaxed to a linear programming problem, such a restriction renders the final solution of A only sub-optimal instead of globally optimal even in the 1-norm matrix regularized framework [9].

In contrast to the original SML model, a more generalized SML can be described as follows:

$$\min_{U \in \mathbf{O}_D, A} \sum_t \xi_t + \gamma ||AA^\top||_{(2,1)} \tag{6}$$

$$\text{s.t.} \quad ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j||_2^2 \leq ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_k||_2^2 + \xi_t, \forall (i,j,k) \in \mathcal{T},$$

$$\xi_t \geq 0 \ \forall t,$$

$$\hat{\mathbf{x}}_i = A^\top U^\top \mathbf{x}_i \ ,$$

where $||AA^\top||_{(2,1)}$ means the mixed $(2,1)$-norm for the matrix $AA^\top$, which we will discuss shortly.

There are two major differences between our generalized model and the previous SML model. To see this, we let

$$B = AA^\top.$$

First, an extra matrix parameter $U \in \mathbf{O}^D$ is introduced. The input data samples are firstly transformed to a new space using U. The feature selection or the low-dimensional mapping is then pursued in the transformed space. More importantly, we do not need to specify the matrix U. Instead, the optimal U can be automatically learned from the model. Second, we use a mixed $(2,1)$-norm $||B||_{(2,1)}$ instead of the 1-norm. The mixed norm has been earlier used in multi-task learning [2] and the $(2,1)$-norm of B is obtained by first computing the 2-norm across the rows of B_i, and then the 1-norm of the vector $b(B) = (||B_1||_2, ||B_2||_2, \ldots, ||B_D||_2)$. A 1-norm will impose a sparsity on a vector, meaning that

[3] This is similar to Principle Component Analysis

some elements of $b(B)$ will be zeroed. Let B_k and $(A^\top)_k$ respectively denote the k-th row vector of B and k-th column vector of $(A^\top)$. Then it is easy to verify that

$$(B_k)^\top = 0 \text{ iff } (A^\top)_k = 0.$$

Hence, the mixed $(2,1)$-norm of B ensures that some columns of $A^\top$ can become zero vectors and thus this naturally achieves a small d.

We propose the following theorem showing that the above original non-convex programming problem can be equivalently transformed to a convex form. Moreover, this transformed convex optimization is a special case of our proposed general framework, if L is set to the identity matrix.

Theorem 1 *The problem in (6) is equivalent to the optimization problem (3) with L equal to the identity matrix, or the following problem*

$$\begin{aligned} \min_{W,\xi} \quad & \sum_t \xi_t + \gamma \mathbf{tr}(W) && (7)\\ s.t. \quad & \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} \le \mathbf{x}_{ik}^\top W \mathbf{x}_{ik} + \xi_t && (8)\\ & \forall (i,j,k) \in \mathcal{T}, \xi_t \ge 0, \forall t && (9)\\ & W \succeq 0. && (10) \end{aligned}$$

The proof can seen in the Appendix. From Theorem 1, we know immediately that our proposed unified framework contains the generalized SML as a special case. Moreover, as the problem in (6) presents the optimal model that achieves the goal of sparse metric learning directly, we can have the following proposition.

Proposition 1 *The optimization problem (3) with L equal to the identity matrix presents an optimal model that is able to achieve the objective of sparse metric learning directly.*

2.2.2 Connection with Large Margin Nearest Neighbor

We show that a special problem with L set to the "covariance" matrix Σ_s among the similar pairs, i.e., $L = \frac{1}{|\mathcal{T}_s|}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top$ ($\mathcal{T}_s$ defines the similar pair set), is identical to the Large-Margin Nearest Neighbor (LMNN) approach.[4]

This can be easily verified as follows:

$$\begin{aligned} \mathbf{tr}(LW) &= \frac{1}{|\mathcal{T}_s|}\mathbf{tr}(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top W)\\ &= \frac{1}{|\mathcal{T}_s|}\mathbf{tr}(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s}(\mathbf{x}_i-\mathbf{x}_j)^\top W(\mathbf{x}_i-\mathbf{x}_j))\\ &= \frac{1}{|\mathcal{T}_s|}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s}(\mathbf{x}_i-\mathbf{x}_j)^\top W(\mathbf{x}_i-\mathbf{x}_j)\ . \end{aligned}$$

Hence the modified problem can be changed to

$$\min_{W,\xi} \quad \sum_t \xi_t + \gamma \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s}(\mathbf{x}_i-\mathbf{x}_j)^\top W(\mathbf{x}_i-\mathbf{x}_j)\ .$$

Evidently, the above problem is identical to the optimization problem given in LMNN.[5]

Remarks. As indicated in Section 2.2.1, it is the term $\mathbf{tr}(W)$ that conveys the sparsity directly. The regularization term given by LMNN is motivated from the large margin concept, which is however not directly related to the sparsity of the distance matrix W.

2.2.3 Connection with D-ranking Vector Machine

We propose Theorem 2 showing that the D-ranking Vector Machine proposed in [14] is a special case of our proposed framework.

Theorem 2 *When the transformation matrix W is of rank one and L is equal to the identity matrix, i.e.,* $W = \mathbf{v}\mathbf{v}\top, L = \mathbf{I}$ *(*$\mathbf{v}$ *is a* $D\times 1$ *vector), problem (2) is equivalent to the D-ranking Vector Machine [14].*

Proof: If we substitute $M = \mathbf{v}\mathbf{v}\top$ into (7), we can obtain the following optimization problem

$$\begin{aligned} \min_{\mathbf{v},\xi} \quad & \sum_t \xi_t + \gamma \mathbf{v}\top\mathbf{v} && (11)\\ \text{s.t.} \quad & \mathbf{x}_{ij}^\top \mathbf{v}\mathbf{v}\top \mathbf{x}_{ij} \le \mathbf{x}_{ik}^\top \mathbf{v}\mathbf{v}\top \mathbf{x}_{ik} + \xi_t && (12)\\ & \forall (i,j,k) \in \mathcal{T}, \xi_t \ge 0, \forall t. \end{aligned}$$

(12) can be transformed to $\mathbf{x}_{ij}^\top\mathbf{v} \le \mathbf{x}_{ik}^\top\mathbf{v} + \xi_t$ in the sense that both forms imply the same meanings. On the other hand, the term $\mathbf{v}\top\mathbf{v}$, the L_2-norm, can be also defined as $||\mathbf{v}||_{\mathbb{H}}$, which is an L_2-norm in a reproducing kernel Hilbert space (RKHS). Hence, the above problem can be finally transformed to

$$\begin{aligned} \min_{\mathbf{v},\xi} \quad & \sum_t \xi_t + \gamma ||\mathbf{v}||_{\mathbb{H}}^2 && (13)\\ \text{s.t.} \quad & \mathbf{x}_{ij}^\top\mathbf{v} \le \mathbf{x}_{ik}^\top\mathbf{v} + \xi_t && (14)\\ & \forall (i,j,k) \in \mathcal{T}, \xi_t \ge 0, \forall t. \end{aligned}$$

The above optimization problem is exactly the D-ranking Vector Machine [14]. □

From Lemma 2, we know that D-ranking VM actually restricts the rank of the transformation matrix to one. Although such restriction simplifies the model, it reduces the flexibility and consequently would limit the performance.

[4] The matrix L changes to the real within-class covariance matrix, if we consider a pair with the same class label as the similar pair. See Lemma 1, page 7 of [18] for reference.

[5] The similar set defined in LMNN is slightly different. Such definition could be also used here.

2.2.4 Connections with Pairwise Support Vector Machine

We show that the proposed general framework has a close relationship with the Pairwise Support Vector Machine. More specifically, when L is equal to the distance matrix W, the GSML is reduced to a model very similar to the Pairwise SVM [20].

With $L = W$, we can have:

$$\mathbf{tr}(LW) = \mathbf{tr}(WW) = ||W||_{Fro}^2,$$

where $||W||_{Fro}^2$ is the Frobenius norm of W. Furthermore, suppose we change the triplet setting to another commonly-used setting, where a similar set $\mathcal{T}_s$ (containing similar pairs) and a dissimilar set $\mathcal{T}_d$ (containing dissimilar pairs) are defined. A good choice of distance metric should shrink the distance between each pair of similar points, while enlarging the distance between each pair of dissimilar points. In this sense, we have the following constraints

$$\begin{aligned} \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} &\le \rho - 1 + \xi_{ij}, (i,j) \in \mathcal{T}_s \ , \\ \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} &\ge \rho + 1 + \xi_{ij}, (i,j) \in \mathcal{T}_d \ , \end{aligned}$$

where ρ is a variable used to shift the margin among the dissimilar pairs and the similar pairs.

Consequently,the problem (2) can be written as

$$\begin{aligned} \min_{W,\rho,\xi} \quad & \sum_t \xi_t + \gamma ||W||_{Fro}^2 \\ & \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} \le \rho - 1 + \xi_{ij}, (i,j) \in \mathcal{T}_s \\ & \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} \ge \rho + 1 + \xi_{ij}, (i,j) \in \mathcal{T}_d \ . \end{aligned}$$

The above model is exactly the pairwise SVM model [20].

2.2.5 Extensions of Other Metric Learning Models

We now examine how some other existing metric learning problems can be extended into a sparse learning problem. We use the model of Xing et al [23] as an example. Other models such as RCA can be similarly extended.

We first write the model of [23] as follows:

$$\begin{aligned} \min_{\rho,B} \quad & \rho \\ \text{s.t.} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_d} (\mathbf{x}_i - \mathbf{x}_j)^\top B (\mathbf{x}_i - \mathbf{x}_j) \ge 1 \\ & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s} (\mathbf{x}_i - \mathbf{x}_j)^\top B (\mathbf{x}_i - \mathbf{x}_j) \le \rho \\ & B \succeq 0, \end{aligned}$$

where $\mathcal{T}_d$ and $\mathcal{T}_s$ are respectively a specified dissimilar set and a similar set.

By simply adding the regularization term, we transform the above problem to

$$\begin{aligned} \min_{\rho,W} \quad & \rho + \gamma \mathbf{tr}(LW) \\ \text{s.t.} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_d} (\mathbf{x}_i - \mathbf{x}_j)^\top W (\mathbf{x}_i - \mathbf{x}_j) \ge 1 \\ & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{T}_s} (\mathbf{x}_i - \mathbf{x}_j)^\top W (\mathbf{x}_i - \mathbf{x}_j) \le \rho \\ & W \succeq 0. \end{aligned}$$

It can be interestingly observed that the above sparse version can be regarded as a globalized sparse metric framework, while our proposed general framework is a local sparse metric learning method. In the above model, the constraint is constructed globally from the similar set and the dissimilar set. In comparison, our proposed GSML constructs constraints between each similar pair and dissimilar pair, which is of the local fashion. Both models share similar optimization forms. They can be efficiently solved based on the iterative method, which will be discussed shortly in Section 2.3.

2.2.6 Remarks

In the above subsections, we have established various connections between our proposed unified framework and the other models. We now make some remarks on the mentioned models. First, the generalized SML model (cf. problem (6)) presents the optimal model which can learn an accurate distance metric as well as sparse representations. LMNN is motivated from the large margin concept. It can achieve the similar objective as the generalized SML. LMNN can also yield sparsity because of the inherited nature of the regularization term; however, the sparsity of $\Sigma_s W$ does not mean the sparsity of W. This point can also be observed later in the experimental section. The D-ranking Vector Machine can be regarded as a simplified version of the proposed framework, since the distance matrix is restricted to rank one. Finally, L could be defined as other matrices. In particular, it is possible to combine the merits of LMNN and Generalized SML with L set to $\Sigma_s + \lambda\mathbf{I}$, where λ is a positive trade-off parameter.

2.3 Iterative Optimization

We have shown that the proposed sparse metric learning problem can be formulated as an SDP problem. However, SDP is still time-consuming, or even intractable for medium or large-scale problems due to the time complexity of $O(D^6)$. In this section, in a similar fashion to Weinberger et al [22], we propose an iterative optimization technique which scales well for large-scale tasks.

The basic strategy is that we first solve a linear programming problem with standard sub-gradient descent methods by removing the semi-definite constraint $W \succeq 0$. We then project the solution to the semi-definite matrices space. The process is iterated until a stable solution is obtained. We state this process as follows.

By defining the hinge loss $[z]_+ = \begin{cases} z & \text{if } z > 0, \\ 0 & \text{otherwise} \end{cases}$, we first transform the problem to an unconstrained optimization problem

$$\min_{W \succeq \mathbf{0}} \quad \sum_t [1 + \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} - \mathbf{x}_{ik}^\top W \mathbf{x}_{ik}]_+ + \gamma \mathbf{tr}(LW) \quad (15)$$

Let $\mathbf{C}_{ij} = \mathbf{x}_{ij}\mathbf{x}_{ij}^\top$ and let G^t denote the gradient of the objective function at the t-th iteration, then G^t can be calculated as follows:

$$G^t = \gamma L + \sum_{(i,j,k) \in \mathcal{H}^t} (\mathbf{C}_{ij} - \mathbf{C}_{ik}),$$

where $\mathcal{H}^t$ is the set of triplets $(i, j, k) \in \mathcal{T}$ with the positive slack variable. At each step, the metric matrix W can be updated by

$$W_{(t)} = W_{(t-1)} + \alpha G^t,$$

where α is a small positive step constant.

We then project the matrix $W_{(t)}$ to the cone of positive semi-definite matrices by finding the eigen-decomposition of the matrix W_t, i.e., $W_{(t)} = P^\top \Lambda P$, where Λ is the diagonal matrix with the diagonal elements λ_i being the eigen-values of $W_{(t)}$, and P is the the eigen-vector matrix. The optimal $W_{(t)}$ satisfying $W_{(t)} \succeq \mathbf{0}$ is the one, $W_{(t)} = P^\top \Lambda_+ P$, where $\Lambda_+ = \text{diag}(\max\{0, \lambda_1\}, \ldots, \max\{0, \lambda_D\})$.

Following [22], we can further speed up the optimization by updating the gradient as follows:

$$\begin{aligned} G^{t+1} &= G^t - \sum_{(i,j,k) \in \mathcal{H}^t - \mathcal{H}^{t+1}} (\mathbf{C}_{ij} - \mathbf{C}_{ik}) \\ &+ \sum_{(i,j,k) \in \mathcal{H}^{t+1} - \mathcal{H}^t} (\mathbf{C}_{ij} - \mathbf{C}_{ik}). \end{aligned}$$

The second technique to speed up the optimization is to use the active set method. At each iteration, we do not check all the triplets if they violate the margin (i.e., $\xi_t > 0$). Instead, we define the current active set $\mathcal{H}^t = \bigcup_{i=1}^{t-1} \mathcal{H}^i$. When the algorithm converges, we then check if the current working set contains all the triplets satisfying $\xi_t > 0$. Otherwise, we add the newly obtained triplets into the active set and continue the optimization until it converges.

3 Experiments

In this section, we compare our proposed method with other competing models on six benchmark data sets,which were obtained from the UCI machine learning repository.

As we proved in Section 2.2.1, the model with $L = I$ corresponds to the optimal model capable of simultaneously learning an accurate distance metric and a low-dimensional representation. In this section, our mention of GSML in numerical experiments will assume this $L = I$ model. However, we should still bear in mind that GSML is a general framework and thus other variations could have been used instead. With this choice we compare the proposed GSML method with four other competing methods including the naive Euclidean distance method (simply using Euclidean distance as the distance metric), the SML method [15], the algorithm proposed by Xing et al [23] (we call Xing from now on), and the Large Margin Nearest Neighbor method.

We follow [15] and use the category information to generate the relative similarity pairs. More specifically, given a randomly chosen triplet $\{i, j, k\}$ sampled from the training data, if two of them share the same label and a third has a different class, we then incorporate this triplet in the triplet set $\mathcal{T}$. Namely, $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$ has the relative distance relationship of $\mathbf{x}_i$ *is more similar to* $\mathbf{x}_j$ *than to* $\mathbf{x}_k$. For Xing's method, two sets need be used for evaluating the algorithm: a similar set $\mathcal{T}_s$ containing the similar pairs and a dissimilar set $\mathcal{T}_d$ consisting of the dissimilar pairs. In a similar fashion to [15], we generate these two sets from the triplet set $\mathcal{T}$ by placing (i, j) into the similar set $\mathcal{T}_s$ and (i, k) into the dissimilar set $\mathcal{T}_d$, provided (i, j, k) is a triplet in $\mathcal{T}$. As discussed in [15], such a strategy provides a fair level of supervision for the comparison methods, in the sense that roughly the same information is presented to different methods. For LMNN, the nearest neighbor points for each training sample need to be provided before training. In order to provide equal side information, for each training sample, we regard their similar samples appearing in the similar pairs as their nearest neighbors.

We use the same ratio, i.e., $85 : 15$ as used by [15], to split the data sets into a training and test set. From the same data set, 1500 triplets are generated from the training set based on the strategy mentioned above, while 1000 triplets are sampled from the test set. During testing, if two of the similar elements in the triplet has a smaller distance than the dissimilar pair (calculated based on the learned metric), then this triplet is regarded as classified correctly. We count the ratio of correctly classified triplets to give the final accuracy score. This procedure was used for all five methods. The final results are given as an average over 10 random splits of the data. The trade-off parameter γ used in GSML, SML, Xing's method, and LMNN is tuned in the range $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$ by cross validation, as suggested by previous research. All the experiments were pursued using Matlab V7.1 on a PC with 2G RAM, and a 3Ghz CPU.

3.1 Results

We now report the experimental results. We will first present the test accuracy and then provide the performance in terms of dimensionality reduction.

We report the test accuracy in Table 1. In the first two

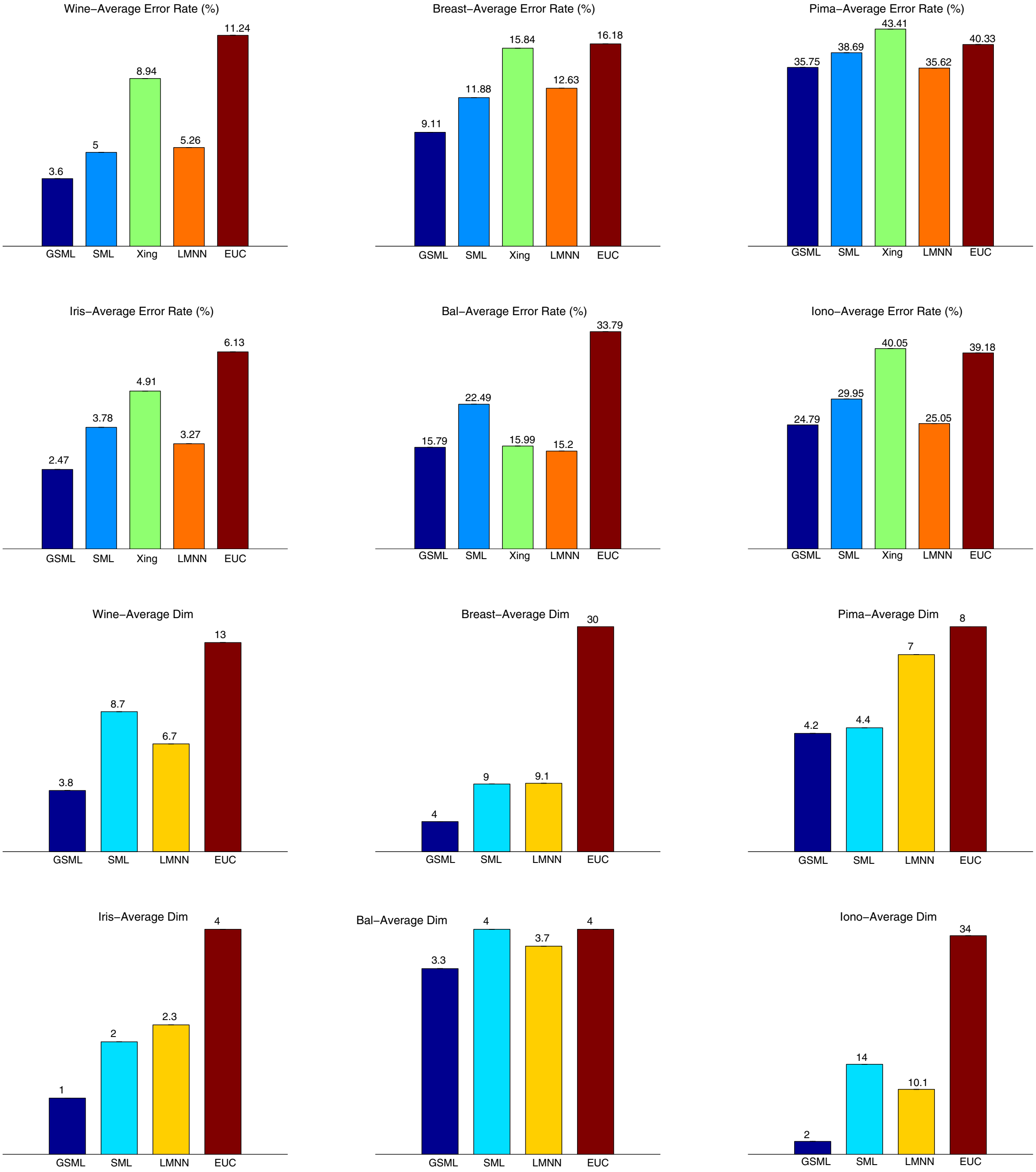

Figure 2. Performance comparison among different methods. The first two rows present the average test error rates, while the last two rows plot the average dimensionality used in different methods.

Table 1. Test Accuracy comparison for our model versus the four other competing methods. GSML demonstrated overall best performance compared with the other methods.

Data Set	GSML	SML	Xing	LMNN	Euclidean
Wine (%)	$\mathbf{96.45 \pm 1.97}$	95.00 ± 1.46	91.06 ± 3.12	94.74 ± 1.95	88.76 ± 2.85
Breast (%)	$\mathbf{90.89 \pm 2.71}$	88.12 ± 2.97	84.16 ± 5.01	87.37 ± 1.16	83.82 ± 5.21
Pima (%)	64.25 ± 3.06	61.31 ± 3.26	56.59 ± 1.84	$\mathbf{64.38 \pm 3.90}$	59.67 ± 2.75
Iris (%)	$\mathbf{97.53 \pm 2.31}$	96.22 ± 1.94	95.09 ± 3.76	96.73 ± 3.22	93.87 ± 3.34
Ionosphere (%)	$\mathbf{75.21 \pm 3.50}$	70.05 ± 2.95	59.95 ± 3.17	74.95 ± 4.57	60.82 ± 3.77
Balance (%)	84.21 ± 3.09	77.51 ± 3.21	84.01 ± 3.02	$\mathbf{84.80 \pm 3.71}$	66.21 ± 3.02

rows of Figure 2, we also visualize the results. Several important points should be highlighted. First, almost all the metric learning methods can outperform the Euclidean distance method except Xing's method of [23] which can sometimes be slightly worse than the Euclidean distance. This validates the data-dependant nature of distance or similarity metrics and indicates the importance of metric learning. Second, GSML demonstrates overall better performance than the other remaining methods, showing that sparse metric learning appears to be more suitable and appropriate. On the one hand, enforcing sparsity in metric learning can suppress possible noise in the data and consequently avoid over-fitting for subsequent classification or clustering tasks. Additionally, it will lead to much more efficient calculation of the learned distance functions. Third, our proposed GSML further lifts the performance of SML consistently across all six data sets used. The accuracy improvements are statistically significant for the data sets of Breast-Cancer, Pima Diabetes, Ionosphere, and Balance Scale according to a t-test at a significance level of 5%. Specifically, a roughly 5% and 7% accuracy lift has been observed respectively with the Ionosphere and Balance Scale data sets. Finally, LMNN can also present competitive results, however not as good as our method. These results clearly demonstrate the advantages of the proposed generalized framework.

We now examine dimensionality reduction performance (i.e. the number of non-zero vector for the transformation matrix A), used by different metric learning methods in the last two rows of Figure 2.[6] As observed, the optimal dimensionality given by the proposed GSML method has significantly fewer dimensions compared to SML and LMNN across all six data sets. As the feature correlations may appear best in an unknown linearly transformed space and SML can only perform feature selection in the original input space, the "optimal" dimensionality number given by SML is actually not optimal and still redundant. Moreover, although LMNN can impose partial sparsity, the associated term does not target sparsity directly. In comparison, our proposed GSML can learn the optimal low-dimensional feature representation as well as the distance metric at the same time. The dimension number given by GSML achieves the truly optimal solution within the framework of sparse metric learning and hence is much smaller than the other two methods.

4 Related Work

Metric learning is an active research topic in machine learning and data mining. Researchers have developed many approaches in this field. Among them are Information-Theoretic Metric Learning [7] (ITML), Relevant Component analysis (RCA) [4], and the method proposed by Xing et al. However, all the above methods can merely derive non-sparse metrics and only work within their special settings.

Some other related methods include non-Mahalanobis based metric learning, e.g., Neighborhood Component Analysis (NCA) [10], the method proposed in [5], Local Linear Embedding (LLE) [16], and Local Fisher Discriminant Analysis (LFDA) [18]. NCA and the method proposed in [5] are neural network based metric learning methods, usually suffering from non-convexity or suboptimal performance. LLE and LFDA aim to preserve the locality or within-class covariances using low-dimensional mappings. They do not try to improve the classification performance and hence are very different from our proposed method. Embedding approaches or manifold learning are also related to our framework in that both approaches seek low-dimensional representation. These approaches include Maximal Variance Unfolding (MVU) or its variant, Colored MVU [17], Multidimensional Scaling (MDS) [6], and its generalized case, Generalized MDS [1]. Principal Component Analysis (PCA) [12] and various other extensions can also be cast in this category. However, all these approaches are differently motivated, leading to very different objective formulations compared to our metric learning framework. The main purpose of embedding approaches is to find a low-dimensional embedding which maintains the distance ordering. As shown by many authors, embedding approaches do not necessarily benefit classification performance [17]. In contrast, we are seeking a metric which simultaneously yields high classification or clustering accuracy as well as a

[6] Xing's method and Euc are non-sparse methods. They simply use all the dimensions. For brevity, we only plot the number of dimensionality used by Euc.

low-dimensional representation [1] [17].

Within sparse metric learning, SML [15], LMNN [21], D-ranking Vector Machine [14], Large Margin Component Analysis (LMCA) [19], and the method developed in [3] also target high classification accuracy and dimensionality reduction simultaneously. However, SML is sub-optimal and can merely search low-dimensional representations in the input space. Our proposed Generalized SML solves these two problems systematically and is shown to contain SML, LMNN, and D-ranking Vector Machine as special cases. On the other hand, LMCA controls the sparsity by directly specifying the dimensionality of the transformation matrix and it is an extended version of LMNN [21, 22]. Its problem is that, one more parameter, i.e., the dimension d, needs tuning. In [3], a distance measure can be learned by combining multiple 1-D embeddings based on the AdaBoost [3]. However, it exploits some heuristics and cannot guarantee global optimum.

5 Conclusion

We propose a Generalized Sparse Metric Learning framework in this paper. This novel framework offers a unified view for understanding many of the popular sparse metric learning algorithms including the Sparse Metric Learning framework proposed in [15], the Large Margin Nearest Neighbor (LMNN) [21][22], and the D-ranking Vector Machine (D-ranking VM) [14]. Furthermore, the proposed framework is capable of extending many current non-sparse metric learning models such as Relevant Vector Machine [4] and a state-of-the-art method proposed in [23] to their sparse versions. We provide a conceptual interpretation why the proposed framework can generate both a good distance metric and low-dimension representations simultaneously. We also apply an iterative sub-gradient optimization method, making the original SDP problem solvable even for medium- or large-scale data sets. Experimental results show that the proposed unified approach can outperform previous methods in terms of both learning accuracy and dimension reduction on six real-world datasets.

Acknowledgement

This work is partially supported by the Excellent SKL Project of NSFC (No.60723005), China.

References

[1] S. Agarwal, J. Wills, L. Gayton, G. Lanckriet, D. Kriegman, and S. Belongie. Generalized non-metric multidimensional scaling. In *International Conference on Artificial Intelligence and Statistics (AISTAT'08)*, 2008.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems (NIPS) 18*, 2006.

[3] V. Athitsos, J. Alton, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[4] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR-2005)*, 2005.

[6] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.

[7] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Intermational Conference on Machine Learning (ICML)*, 2007.

[8] G. Fung, O. L. Mangasarian, and A. J. Smola. Minimal kernel classifers. *Journal of Machine Learning Research*, 3:303–321, 2002.

[9] G. Fung, R. Rosales, and R. B. Rao. Feature selection and kernel design via linear programming. In *Proceedings of Internet Joint Conference on Artificial Intelligence (IJCAI)*, pages 786–791, 2007.

[10] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[11] T. Hastie, R. Tibshirani, and R. Friedman. *The Elements of Statistical Learning*. Springer-Verlag New York, LLC, 2003.

[12] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1989.

[13] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

[14] H. Ouyang and A. Gray. Learning dissimilarities by ranking: from sdp to qp. In *Proceedings of the Twenty-five International Conference on Machine learning (ICML-2008)*, 2008.

[15] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *KDD*, pages 367–373, 2006.

[16] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, (290):123–137, 2000.

[17] L. Song, A. Smola, K. Borgwardt, and A. Gretton. Colored maximum variance unfolding. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[18] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 18:1027–1061, 2007.

[19] L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[20] Jean-Philippe Vert, Jian Qiu, and William S. Nobel. A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, 8, 2007.

[21] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[22] K. Weinberger and L. Saul. Fast solvers and efficient implentations for distance metric learning. In *Proceedings of the twenty-fifth international conference on Machine learning (ICML-2008)*, 2008.

[23] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side information. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

Appendix

To prove Theorem 1, we first present Lemma 1.

Lemma 1 , *Consider the following optimization problem.*

$$\begin{aligned} \min_{W,V,\xi} \quad & \sum_t \xi_t + \gamma\, \mathbf{tr}(WW^\top V^{-1})^{\frac{1}{2}} && (16) \\ \textit{s.t.} \quad & \mathbf{tr}(V) \le 1, V \in \mathbf{S}^D_{++} \\ & \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} \le \mathbf{x}_{ik}^\top W \mathbf{x}_{ik} + \xi_t && (17) \\ & \forall (i,j,k) \in \mathcal{T}, \\ & \xi_t \ge 0, \forall t && (18) \\ & W \succeq 0. && (19) \end{aligned}$$

The above sparse metric learning problem (6) is equivalent to the formulation in (16). In particular, if $(\hat{A}, \hat{U})$ is an optimal solution of (6), then

$$(\hat{W}, \hat{V}) = \Big(\ \hat{U}\hat{B}\hat{U}^\top, \quad \hat{U} Diag(\tfrac{||\hat{B}^i||_2}{||\hat{B}||_{2,1}})_{i=1}^D \hat{U}^\top\ \Big)$$

is an optimal solution of Problem (16). Moreover, the above problem (16) is a jointly convex problem with respect to W, ξ and V.

Proof: Let $W = UBU^\top$, then problem (6) is reduced to the following

$$\begin{aligned} \min_{W,U,\xi} \quad & \sum_t \xi_t + \gamma\, \|U^\top W U\|_{(2,1)} && (20) \\ \text{s.t.} \quad & \mathbf{x}_{ij}^\top W \mathbf{x}_{ij} \le \mathbf{x}_{ik}^\top W \mathbf{x}_{ik} + \xi_t \\ & \forall (i,j,k) \in \mathcal{T},\ \xi_t \ge 0, \forall t \\ & W \in \mathbf{S}^D_+,\ U \in \mathbf{O}^D. \end{aligned}$$

We know from [13] that $\min_{\lambda \in \triangle} \sum_i \frac{a_i^2}{\lambda_i} = (\sum_i |a_i|)^2$, where the minimization is achieved at $\hat{\lambda}_i = \frac{|a_i|}{||a||_1}$. Let $\lambda^{-1} = (\lambda_1^{-1}, \ldots, \lambda_n^{-1})$. Then, replacing a_i by $\|(U^\top W U)_i\|$ implies that

$$\begin{aligned} \|U^\top W U\|^2_{(2,1)} & = \min_{\lambda \in \triangle} \textstyle\sum_{i=1}^n \frac{\|(U^\top W U)_i\|^2}{\lambda_i} \\ & = \mathbf{tr}((U^\top W U)^\top (U^\top W U) \text{diag}(\lambda^{-1})) \\ & = \mathbf{tr}((U^\top W^\top W U) \text{diag}(\lambda^{-1})) \\ & = \mathbf{tr}(W^\top W (U^\top \text{diag}(\lambda^{-1}) U)) \end{aligned}$$

Putting this back into equation (20) and recalling that $\lambda \in \triangle$ and $U \in \mathbf{O}^D$ are arbitrary, then letting $V^{-1} = U^\top \text{diag}(\lambda^{-1}) U \in \mathbf{S}^D_{++}$, implies the equivalent formulation (16).

To establish the convexity of the formulation in (16), we note that it follows directly from the observation that $\mathbf{tr}(W^\top V^{-1} W)$ is jointly convex with respect to W and V and the constraint conditions are linear. This completes the proof. □

Proof of Theorem 1: If we recall that the set of all real-valued symmetric $D \times D$ matrices, denoted by $\mathbf{S}^D$, is a Hilbert space with trace norm as the inner product defined by, for any $A, B \in \mathbf{S}^D$, $\langle A, B \rangle_{\mathbf{tr}} = \mathbf{tr}(AB^\top)$. Consequently, Cauchy-Schwarz's inequality holds true, i.e.

$$|\langle A, B \rangle_{\mathbf{tr}}|^2 \le \langle A, A \rangle_{\mathbf{tr}} \langle B, B \rangle_{\mathbf{tr}}.$$

Applying the above inequality with $A = (WW^\top)^{\frac{1}{2}} V^{-\frac{1}{2}}$ and $B = V^{\frac{1}{2}}$ implies that

$$|\mathbf{tr}((WW^\top)^{\frac{1}{2}})|^2 \le \mathbf{tr}(V) \langle WW^\top, V^{-1} \rangle_{\mathbf{tr}}.$$

Since $\mathbf{tr}(V) \le 1$, we have that $\mathbf{tr}((WW^\top)V^{-1}) \ge |\mathbf{tr}((WW^\top)^{\frac{1}{2}})|^2$.

If we further notice that $W = UAA^\top U^\top$ is a symmetric and positive semi-definite matrix, we have

$$\mathbf{tr}((WW^\top)V^{-1}) \ge |\mathbf{tr}((WW^\top)^{\frac{1}{2}})|^2 = \mathbf{tr}(W)^2. \quad (21)$$

Moreover, the minimum is achieved when $V = \frac{(WW^\top)^{\frac{1}{2}}}{\mathbf{tr}((WW^\top)^{\frac{1}{2}})}$. If we substitute (21) into the problem (16), we can obtain the problem (7) immediately. Using Lemma 1, we complete the proof. □