# Regularized Multidimensional Scaling with Radial Basis Functions

Sohana Jahan and Hou-Duo Qi

*Abstract*—**The classical Multi-Dimensional Scaling (MDS) is an important method for data dimension reduction. Nonlinear variants have been developed to improve its performance. One of them is the MDS with Radial Basis Functions (RBF). A key issue that has not been well addressed in MDS-RBF is the effective selection of its centers. This paper treats this selection problem as a multi-task learning problem, which leads us to employ the $(2,1)$-norm to regularize the original MDS-RBF objective function. We then study its two reformulations: Diagonal and spectral reformulations. Both can be effectively solved through an iterative block-majorization method. Numerical experiments show that the regularized models can improve the original model significantly.**

***Keywords*—*Multi-Dimensional Scaling, Radial Basis Function, Iterative Majorization, Regularization.***

## I. Introduction

THE classical Multi-Dimensional Scaling (cMDS) and its nonlinear variants have found many applications in both social and engineering sciences and are well documented in the books by Cox and Cox [7], Borg and Groenen [4], and Pękalaska and Duin [16]. In this paper, we are interested in one of the important nonlinear variants involving Radial Basis Functions (RBF) that was first proposed by Webb [26], [27] in the context of MDS. The key issue in employing RBFs in MDS is to decide their centers. This includes the number of the centers to be used and then what they are. This issue has not been well addressed in existing literature. For example, Webb [26] suggests to randomly choose the centers and then uses an expensive cross-validation procedure to decide what they are. Here, we take a completely different route and regard the selection of the centers as a multi-task learning problem that has been widely studied in machine learning, see Argriou et al. [1], [2]. This will lead us to an optimization model that can be solved efficiently. Before we detail our method, we give a brief literature review and discuss how they have led to the current research.

The use of cMDS as a data dimension-reduction method (or data visualization method when the embedding dimension is 2 or 3) can be traced to the seminar work of Schoenberg [21] and the independent work of Young and Householder [28]. The method was made popular by Torgerson [24] and later by Gower [12] (see [14, Chapter 14] for details). cMDS performs well if the distance matrix, which consists of the pairwise distance among the data points, is close to a true Euclidean distance matrix with a low-embedding dimension. Otherwise, certain corrections have to be made on the distance matrix. Early methods include adding a same positive constant to every pairwise distance, which results in the additive constant or the partial additive constant problems (see [15], [6], [5], [3], [18]). More advanced corrections are obtained through optimizing certain loss functions. The `STRESS` function first proposed by Kruskal [13] is one of the most often used loss functions (for other `STRESS` type functions, see [4, Chapter 3]). The resulting optimization problems based on `STRESS` functions can be efficiently solved by the *majorization method* introduced by de Leeuw [8] (see [4, Chapter 8] for a detailed description of the method). We will also use a majorization procedure in our algorithm. Another class of corrections can be obtained through computing the nearest Euclidean distance matrix from the known distance matrix (see [10], [11], [17], [19], [20]). All of these methods can be regarded as nonlinear variants of cMDS because they make nonlinear corrections on the pairwise distances.

The nonlinear variant introduced by Webb [25] differs from those mentioned above in the following way. It regards the space where the original data lies the input space (also see [26]). The first stage of Webb's method is to map the data from the input space to a feature space through nonlinear functions such as RBFs. The dimension of the feature space is determined by the number of RBFs used and is equal to the number of centers used in RBFs. Assuming the first stage task is settled, the second stage is to find the *best* linear function that maps the feature space data to a low-dimensional embedding space (2 or 3 if the purpose is to visualize the data). Webb's method actually focuses on the second stage and suggests using a (potentially very expensive) cross-validation procedure to furnish the task in the first stage.

The purpose of this paper is to propose a computational model that deals with the two stages simultaneously. The key viewpoint here is to regard the selection of the centers for RBFs as a kind of multi-task learning problem, which has been widely studied in machine learning (see [1], [2]). We would like to emphasize that there are major differences between our learning problem and that in [2]. Roughly speaking, we have a non-convex optimization model while [2] has a convex one. But the principal idea of choosing the common tasks via minimizing the $(2,1)$-norm of the learning matrix in [2] is carried over to our model. This $(2,1)$-norm works as a regularizer to control the selection of centers for RBFs. We

First version: January 9, 2014

S. Jahan is with the School of Mathematics, University of Southampton, UK. The research of this author was supported by the Commonwealth Scholarship BDCS-2012-44. E-mail: sj1g12@soton.ac.uk.

Corresponding author. Hou-Duo Qi is with the School of Mathematics, University of Southampton, UK. The research of this author was supported in part by Engineering and Physical Science Research Council (UK) project EP/K007645/1. E-mail: hdqi@soton.ac.uk.

will develop an iterative block-majorization method for the resulting model.

The paper is organized as follows. In Section II, we will review the RBF-MDS model introduced by Webb [26] and single out the problem how to choose centers for the RBFs used. We will then introduce the $(2,1)$-norm as a regularizer to the model. On the way, we will also highlight the major differences as well as relationships between our model and the multi-task learning model in [2]. In Section III, we will study two reformulation models: diagonal and spectral. We will then develop an iterative block-majorization method for our model. Numerical results on three commonly used data sets are reported and explained in Section IV, where we demonstrate that the regularized models can significantly improve the original model of Webb [26]. We conclude the paper in Section V.

## II. The Problem of Learning Centers

In this section, we first introduce the RBF-MDS model of Webb [26]. We then treat the center selection problem in the model as a multi-task learning problem.

### A. RBF-MDS Model

Suppose we have $N$ data points $\{\mathbf{x}_i\}_{i=1}^N$ in the input space $\Re^n$ and their associated Euclidean distances $d_{ij}$ is defined to be $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, where $\|\cdot\|$ is the Euclidean norm in $\Re^n$. Due to practical reasons, the original data contains noises and they can be represented in a lower-dimensional space $\Re^m$ ($m \ll n$). For example, when it is for visualization, $m$ is often chosen to be 2 or 3. The representation is often done through nonlinear dimension reduction methodologies.

Webb [26] proposed the following methodology. Firstly, the data set is mapped to another space called feature space $\Re^\ell$ through nonlinear function $\Phi : \Re^n \mapsto \Re^\ell$. For example, $\phi$ can be radial basis functions. Let $\Phi(x) = (\phi_1(x), \ldots, \phi_\ell(x)) \in \Re^\ell$, with

$$\phi_i(x) = \exp\left\{-\|\mathbf{x} - \mathbf{c}_i\|^2/h^2\right\}, \qquad i = 1, \ldots, \ell$$

where $h$ is the bandwidth and $\mathbf{c}_i$ is the center of $\phi_i$. Secondly, the form of data representation in $\Re^m$, denoted as $\mathbf{f}$, is assumed to be a linear function of the feature vector $\Phi$. In terms of the original input space data, $\mathbf{f}$ is a nonlinear function from $\Re^n$ to $\Re^m$ and takes the following form:

$$\mathbf{f}(x) = W^T\Phi(x), \qquad \forall\ x \in \Re^n \tag{1}$$

where $W \in \Re^{\ell\times m}$. In other words, $\mathbf{f}$ is a composite of a linear function (represented by the matrix $W$) and the radial basis function $\Phi$. Finally, the method seeks the best linear function that minimizes the raw `STRESS` (i.e., loss function):

$$\sigma^2(W) = \sum_{i,j=1}^{N} \alpha_{ij}\left(q_{ij}\left(W\right) - d_{ij}\right)^2, \tag{2}$$

where for $i, j = 1, \ldots, N$, $\alpha_{ij} > 0$ are known weights and

$$q_{ij}(W) = \|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)\| = \|W^T(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))\|. \tag{3}$$

Hence, the optimization problem of Webb's model is

$$\min_{W\in\Re^{\ell\times m}} \ \sigma^2(W). \tag{4}$$

A majorization method is then used to solve (4). Let $\mathbf{v}^{ij} = \Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)$. We assume that the data set is rich enough such that the vectors

$$\left\{\mathbf{v}^{ij} :\ i < j = 2, \ldots, N\right\} \text{ span the feature space } \Re^\ell. \tag{5}$$

It is obvious that one of the key components of Webb's model is computing the feature vector $\Phi(x)$, which depends on its centers $\mathbf{c}_i$, $i = 1, \ldots, \ell$. There are two natural questions to be asked here. How many centers should be used (i.e., how to decide $\ell$)? What are the best choices of those centers? Webb [26] suggests to randomly choose the centers and to use a cross-validation scheme to pick the best one. However, the cross-validation scheme is often very expensive to run. In the following, we try to answer those questions from a fresh viewpoint of multi-task learning.

### B. As a Multi-Task Learning Problem

A general setting up for multi-task learning problems is described in [2, Sect. 2]. In this section, we will relate the center choosing problem to a multi-task learning problem. Suppose there are $\ell$ factors represented by $\phi_i(x)$, $i = 1, \ldots, \ell$ and there are $m$ tasks. Each task in our problem can be represented as a linear regression of the $\ell$ factors:

$$\mathbf{f}_i(x) = \langle W_{:i}, \Phi(x)\rangle = \sum_{j=1}^{\ell} W_{ji}\phi_j(x), \qquad i = 1, \ldots, m \tag{6}$$

where $W_{:i}$ (`Matlab` type of notation) is the $i$th column of $W$ and $\langle\cdot,\cdot\rangle$ is the standard inner product in $\Re^\ell$. The purpose is to learn the common factors (out of the $\ell$ factors) among all $m$ tasks, which is explained below.

Suppose $\phi_1(x)$ is not a common factor, then the corresponding coefficients $W_{1i}$, $i = 1, \ldots, m$ should be all zero. In other words, the factor $\phi_1(x)$ can be removed from the linear regression model (6). This corresponds to the 1st row of $W$ being zero. Now, the problem of learning common factors is equivalent to finding the zero rows of $W$. This can be well achieved by minimizing the $(2,1)$-norm of $W$ together with the original objective function $\sigma^2(W)$:

$$\|W\|_{2,1} = \|W_{1:}\| + \ldots + \|W_{\ell:}\|,$$

where $W_{i:}$ is the $i$th row of $W$. For more properties of the $(2,1)$-norm and why it is capable of selecting common factors, please see [2, Sect. 2.2].

Therefore, the optimization model that we are trying to solve becomes

$$\min_{W\in\Re^{\ell\times m}} \ P(W) = \sigma^2(W) + \gamma\|W\|_{2,1}^2, \tag{7}$$

where $\gamma > 0$ is the regularization parameter and $\|W\|_{2,1}$ is the regularization term in the model. Through (7), we can get rid of the centers that are less important in terms of their contributions to $\|W\|_{2,1}$, leading to effective selections of important centers. We should point out that in [2], the number of tasks ($m$) is larger than the number of factors ($\ell$). Here, we have the opposite ($m < \ell$). Furthermore, the objective function corresponding to the raw stress $\sigma^2(W)$ in [2] is convex with respect to $W$. Here, $\sigma^2(W)$ is nonconvex. We shall see that we can nicely combine the majorization strategy and the techniques in handling the $(2,1)$-norm developed in [2] to solve problem (7).

## III. Iterative Block-Majorization Methods

This section is devoted to numerical methods for solving problem (7). The $(2,1)$-norm is nonsmooth (not differentiable) and the `stress` function $\sigma^2(W)$ is not convex. Hence, problem (7) is difficult to solve. We will relate problem (7) to that of [2] in order to spare us from giving very involved technical proofs. This led us to two reformulations that are conducible to developing majorization methods later on.

### A. Diagonal and Spectral Reformulations

Let $\mathcal{S}^\ell$ denote the space of $\ell \times \ell$ symmetric matrices with the standard inner product $\langle \cdot, \cdot \rangle$. Let $\mathcal{S}^\ell_+$ denote the cone of positive semidefinite matrices in $\mathcal{S}^\ell$ and $\mathcal{S}^\ell_{++}$ denote the set of all positive definite matrices in $\mathcal{S}^\ell$. Let $\mathcal{O}^\ell$ denote the set of all $\ell \times \ell$ orthonormal matrices. That is, $U \in \mathcal{O}^\ell$ if and only if $U^T U = I$. For $C \in \mathcal{S}^\ell_+$, we let $C^\dagger$ denote the pseudo-inverse of $C$. For a constant $a \in \Re$, $a^\dagger = 1/a$ if $a \neq 0$ and $a^\dagger = 0$ otherwise. We let $\mathrm{Tr}(C)$ denote the trace of $C$.

Suppose $C \in \mathcal{S}^\ell_+$ has the following spectral decomposition

$$C = U \mathrm{Diag}(\lambda_1, \ldots, \lambda_\ell) U^T,$$

where $\lambda_1 \geq \ldots \geq \lambda_\ell \geq 0$ are the eigenvalues of $C$ in nonincreasing order, $\mathrm{Diag}(\lambda_1, \ldots, \lambda_\ell)$ is the diagonal matrix with $\lambda_i$ being on its diagonal, and $U \in \mathcal{O}^\ell$. The pseudo-inverse of $C$ is then given by

$$C^\dagger = U \mathrm{Diag}(\lambda_1^\dagger, \ldots, \lambda_\ell^\dagger) U^T.$$

Define the function

$$Q(W, C) = \sigma^2(W) + \gamma \langle W W^T, C^\dagger \rangle. \qquad (8)$$

By following the proof of [2, Thm. 1 and Cor. 2], we can obtain the following result.

**Theorem III.1.** *Problem (7) is equivalent to the problem*

$$\begin{array}{ll} \min & Q(W, \mathrm{Diag}(\lambda)) \\ \textit{s.t.} & \lambda = (\lambda_1, \ldots, \lambda_\ell) \geq 0, \ \sum_{i=1}^\ell \lambda_i \leq 1 \\ & \lambda_i \neq 0 \textit{ whenever } W_{i:} \neq 0, \ i = 1, \ldots, \ell. \end{array} \qquad (9)$$

*Moreover, if $(\widehat{W}, \widehat{\lambda})$ is the optimal solution of (9), it holds*

$$\widehat{\lambda}_i = \frac{\|\widehat{W}_{i:}\|}{\|\widehat{W}\|_{2,1}}, \quad i = 1, \ldots, \ell. \qquad (10)$$

Because of this theorem, we call (9) the diagonal reformulation of (7). We now present what we call the spectral reformulation, which has better numerical performance than the diagonal reformulation. We start from a simple observation.

$$W \in \Re^{\ell \times m} \ \text{ if and only if } \ W = UA \qquad (11)$$

for some $U \in \mathcal{O}^\ell$ and $A \in \Re^{\ell \times m}$. The `stress` function $\sigma^2(W)$ can then be written as

$$\begin{aligned} \sigma^2(W) &= \sigma^2(UA) \\ &= \sum_{i,j=1}^N \alpha_{ij} (q_{ij}(UA) - d_{ij})^2 \\ &= \sum_{i,j=1}^N \alpha_{ij} \left( \|A^T U^T (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\| - d_{ij} \right)^2. \end{aligned}$$

We consider the following problem:

$$\min_{A \in \Re^{\ell \times m}, \ U \in \mathcal{O}^\ell} E(A, U) = \sigma^2(UA) + \gamma \|A\|^2_{2,1}. \qquad (12)$$

We note that problem (12) is not equivalent to problem (7) under the transformation in (11). But they have a common term of the `stress` function. This time, $\|A\|_{2,1}$ is the regularizer instead of $\|W\|_{2,1}$. The benefit in using $\|A\|_{2,1}$ is that problem (12) has a nice characterization, which allows us to develop a majorization method. Problem (12) is similar in structure to [2, Problem (4)] and is equivalent to the following problem contained in the next result.

**Theorem III.2.** *Problem (12) is equivalent to the problem*

$$\inf \left\{ Q(W, D) : \ W \in \Re^{\ell \times m}, D \in \mathcal{S}^\ell_{++}, \mathrm{Tr}(D) \leq 1 \right\}. \qquad (13)$$

*In particular, any minimizing sequence of problem (13) is bounded and converges to a minimizer of problem (12). Moreover, if $(\widehat{W}, \widehat{D})$ is any limit of a minimizing sequence, then any $(\widehat{A}, \widehat{U})$ such that the columns of $\widehat{U}$ forms an orthogonormal basis of eigenvectors of $\widehat{D}$ and $\widehat{A} = \widehat{U}^T \widehat{W}$, is an optimal solution of problem (12).*

**Proof.** Let $\nu_s$ denote the infimum of (13). Suppose $\{W^k, D^k\}$ is a minimizing sequence. Then

$$\nu_s = \lim_{k \to \infty} Q(W^k, D^k) \geq \lim_{k \to \infty} \sigma^2(W^k) \geq 0, \qquad (14)$$

because the regularization term in (8) is always nonnegative. Due to the constraint $\mathrm{Tr}(D) \leq 1$ in (13), $\{D^k\}$ is bounded. Suppose that the sequence $\{W^k\}$ is unbounded. Without loss of generality, we assume that

$$\frac{W^k}{\|W^k\|} \to \overline{W} \neq 0. \qquad (15)$$

Dividing both sides of (14) by $\|W^k\|^2$ and taking limits, we obtain

$$0 = \sum_{i,j=1}^N \|\overline{W}^T \mathbf{v}^{ij}\|,$$

which implies

$$\overline{W}^T \mathbf{v}^{ij} = 0 \qquad \forall \ i < j = 2, \ldots, N.$$

Assumption (5) forces $\overline{W} = 0$, which contradicts (15). Hence, the sequence $\{W^k\}$ is bounded. This proves that any minimizing sequence is bounded. The remaining proof can be similarly constructed as in [2, Thm. 1 and Cor. 1]. □

It is because that $\widehat{U}$ is a normalized eigenvector matrix of $\widehat{D}$ and it can be obtained through a spectral decomposition of $\widehat{D}$, we refer to problem (13) as the spectral reformulation model. The next result shows that the spectral reformulation model (13) is a generalization of the diagonal reformulation model (9).

**Proposition III.3.** *Let $\nu_d$ be the optimal objective value of problem (9) and $\nu_s$ be the infimum of problem (13). Then we have*

$$\nu_d \geq \nu_s.$$

*Moreover, if $D$ is restricted to be diagonal in (13), the equality holds.*

**Proof**. Suppose $(W, \lambda)$ is an optimal solution of problem (9). Let $\mathcal{I}$ denote the indices of positive $\lambda_i$:

$$\mathcal{I} = \{i \mid \lambda_i > 0,\ i = 1, \ldots, \ell\} \quad \text{and} \quad \bar{\mathcal{I}} = \{1, \ldots, \ell\} \setminus \mathcal{I}.$$

Let $\ell_0 = |\mathcal{I}|$, the cardinality of $\mathcal{I}$. Define

$$\lambda_{\min} = \min_{i \in \mathcal{I}} \lambda_i.$$

Obviously $\lambda_{\min} > 0$. Define the sequence $\lambda^k \in \Re^\ell$, $k = 1, 2...$ by

$$\lambda_i^k = \left\{ \begin{array}{ll} \lambda_i - \frac{1}{2k}\lambda_{\min} & \text{if } i \in \mathcal{I} \\ \frac{\ell_0}{2k(\ell - \ell_0)}\lambda_{\min} & \text{if } \bar{\mathcal{I}} \neq \emptyset \text{ and } i \in \bar{\mathcal{I}}. \end{array} \right.$$

It is easy to verify that $\lambda^k > 0$ for all $k = 1, 2, \ldots$, and

$$\sum_{i=1}^{\ell} \lambda_i^k = \sum_{i=1}^{\ell} \lambda_i \leq 1.$$

Let $D^k = \text{Diag}(\lambda^k)$. Then, the sequence $(W, D^k)$ satisfies the constraints in (13).

Now we compute the respective objective function values. We first note that

$$\begin{array}{rcl} Q(W, \text{Diag}(\lambda)) & = & \sigma^2(W) + \gamma \langle WW^T, (\text{Diag}(\lambda))^\dagger \rangle \\ & = & \sigma^2(W) + \gamma \displaystyle\sum_{i=1}^{\ell} \left( \|W_{i:}\|^2 \lambda_i^\dagger \right) \\ & = & \sigma^2(W) + \gamma \displaystyle\sum_{i \in \mathcal{I}} \left( \|W_{i:}\|^2 / \lambda_i \right). \end{array}$$

It also follows from the constraints in (9) that

$$\lambda_i \in \mathcal{I} \quad \text{whenever} \quad W_{i:} \neq 0.$$

This property yields

$$\begin{array}{rcl} Q(W, D^k) & = & \sigma^2(W) + \gamma \langle WW^T, (D^k)^\dagger \rangle \\ & = & \sigma^2(W) + \gamma \displaystyle\sum_{W_{i:} \neq 0} \left( \|W_{i:}\|^2 / \lambda_i^k \right) \\ & \leq & \sigma^2(W) + \gamma \displaystyle\sum_{i \in \mathcal{I}} \left( \|W_{i:}\|^2 / \lambda_i^k \right). \end{array}$$

Taking limits on both sides, we have

$$\begin{array}{rcl} \displaystyle\liminf_{k \to \infty} Q(W, D^k) & \leq & \sigma^2(W) + \gamma \displaystyle\lim_{k \to \infty} \sum_{i \in \mathcal{I}} \left( \|W_{i:}\|^2 / \lambda_i^k \right) \\ & = & \sigma^2(W) + \gamma \displaystyle\sum_{i \in \mathcal{I}} \left( \|W_{i:}\|^2 / \lambda_i \right) \\ & = & Q(W, \text{Diag}(\lambda)) = \nu_d. \end{array}$$

As stated before, $(W, D^k)$ is a feasible sequence of problem (13). It is obvious that being the infimum of (13)

$$\nu_s \leq \lim_{k \to \infty} Q(W, D^k).$$

This proves $\nu_s \leq \nu_d$.

The above proof actually shows that if $D$ is restricted to be diagonal, we must have $\nu_s \leq \nu_d$. Now suppose that $D$ is restricted to be diagonal. Let $\{W^k, D^k\}$ be a minimizing sequence of (13). That is

$$\nu_s = \lim_{k \to \infty} Q(W^k, D^k). \tag{16}$$

Denote $D^k$ by $D^k = \text{Diag}(\lambda^k)$ and $\lambda^k > 0$ for $k = 1, 2, \ldots$. By Thm. III.2, the sequence $\{W^k, D^k\}$ is bounded. Without loss of any generality, we assume that

$$W^k \to W \qquad \text{and} \qquad \lambda^k \to \lambda.$$

Obviously, $\lambda \geq 0$ and $\sum_{i=1}^{\ell} \lambda_i \leq 1$. The sequence $\{\langle W^k (W^k)^T, (D^k)^\dagger \rangle\}$ is also bounded because $\{W^k, D^k\}$ is a minimizing sequence of (13) and $\sigma(W^k) \geq 0$ for all $k$. Assume that $W_{i:} \neq 0$ for some $i$. Then $W_{i:}^k \neq 0$ for sufficiently large $k$. We further have

$$\begin{array}{rcl} \infty & > & \displaystyle\lim_{k \to \infty} \langle W^k (W^k)^T, (D^k)^\dagger \rangle \geq \lim_{k \to} \|W_{i:}^k\|^2 (\lambda_i^k)^\dagger \\ & = & \left\{ \begin{array}{ll} \|W_{i:}\|^2 (\lambda_i)^\dagger & \text{if } \lambda_i > 0 \\ \infty & \text{if } \lambda_i = 0. \end{array} \right. \end{array}$$

This can only happen when $\lambda_i > 0$. Thus we have proved that $\lambda_i \neq 0$ whenever $W_{i:} \neq 0$. In other words, $(W, \lambda)$ is feasible with respect to the constraints in (9) and

$$\lim_{k \to \infty} \langle W^k (W^k)^T, (D^k)^\dagger \rangle = \langle WW^T, C^\dagger \rangle,$$

where $C = \text{Diag}(\lambda)$. By continuity of $\sigma^2(\cdot)$, (16) implies

$$\nu_s = \sigma^2(W) + \gamma \langle WW^K, C^\dagger \rangle \geq \nu_d.$$

Combining the first part, we have $\nu_s = \nu_d$. □

Although problem (9) is not exactly a special case of problem (13), Prop. III.3 allows us to treat it as if it was obtained through restricting $D$ to be positive diagonal matrices in (13). Comparing to (9), the matrix $D$ has more freedom to move in (13), hence leading to the lower objective function value $\nu_s$. This is likely to contribute to a lower objective function of $\sigma^2(W)$. This possibility has been confirmed by our extensive numerical experiments.

### B. Iterative Block-Majorization Method

In this section, we develop an algorithm for the spectral model problem (13). It can be straightforwardly applied to the diagonal model (9) with simple modifications.

As we mentioned before, problem (13) is not attainable , but the infimum is finite. Argyrion et. al [2] proved that such kind of problem is equivalent to the following problem, which is attainable:

$$\min \left\{ Q(W,D) : \begin{array}{c} W \in \Re^{\ell\times m} \\ D \in \mathcal{S}^{\ell}_{+}, \mathrm{Tr}(D) \le 1 \\ \mathrm{Range}(W) \subseteq \mathrm{Range}(D) \end{array} \right\}. \tag{17}$$

The optimal objective value of (17) equals the infimum of (13). An interesting result about (17) is that when $W$ is fixed, minimizing $Q(W,D)$ over $D$ in the feasible set of (17) has a closed-form solution:

$$D = \frac{\sqrt{WW^T}}{\mathrm{Tr}\sqrt{WW^T}}. \tag{18}$$

Here, the square root $\sqrt{D}$ of a matrix $D \in \mathcal{S}^{\ell}_{+}$ is defined to be the unique matrix $C \in \mathcal{S}^{\ell}_{+}$ such that $D = C^2$. The result (18) is stated below [2, Eq. (23)]. This is the key result that we are going to use in our block majorization method.

Formula (18) immediately suggests alternatively minimizing $Q(W,D)$ with respect to $W$ and $D$. However, it is well known that the stress function, which is part of $Q(W,D)$, is a very complicated function (nonsmooth, nonconvex) to minimize. A widely adopted method is to approximate it by a simpler function, which is less expensive to minimize. One of such functions is the majorization function used by Webb [26] (see also [4]).

For a given $V \in \Re^{\ell\times m}$ and $i,j = 1,\ldots,N$ define

$$c_{ij}(V) = \left\{ \begin{array}{ll} \alpha_{ij} d_{ij}/q_{ij}(V) & \text{if } q_{ij}(V) > 0 \\ 0 & \text{otherwise,} \end{array} \right.$$

and

$$B(V) = \sum_{i,j=1}^{N} c_{ij}(V)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T \in \mathcal{S}^{\ell}.$$

Let

$$C = \sum_{i,j=1}^{N} \alpha_{ij}(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T.$$

Finally, let

$$\sigma^2_m(W,V) = \mathrm{Tr}(W^TCW) - 2\mathrm{Tr}(V^TB(V)W) + \sum_{i,j=1}^{N} \alpha_{ij} d^2_{ij}.$$

Then, $\sigma^2_m(W,V)$ satisfies the following properties:

$$\sigma^2(W) \le \sigma^2_m(W,V) \qquad \forall\ W,V$$

and

$$\sigma^2(W) = \sigma^2_m(W,W).$$

Because of those properties, $\sigma^2_m(W,V)$ is called a majorization function of $\sigma^2$ at $W$. We note that $\sigma^2_m(W,V)$ is quadratic in $W$.

Now, define

$$Q_m(W,V,D) = \sigma^2_m(W,V) + \gamma\langle WW^T, D^{\dagger}\rangle.$$

Then $Q_m$ is a majorization function of $Q(W,D)$ in the sense that

$$Q_m(W,V,D) \ge Q(W,D), \qquad \forall\ W,V,D$$

and

$$Q_m(W,W,D) = Q(W,D).$$

We note that

$$\begin{aligned} Q_m(W,V,D) &= \langle WW^T, C + \gamma D^{\dagger}\rangle - 2\langle V^TB(V), W\rangle \\ &\quad + \sum_{i,j=1}^{N} \alpha_{ij} d^2_{ij}. \end{aligned}$$

We are ready to present our block-majorization algorithm.

**Algorithm III.4. Iterative Block-Majorization Method**

(S.0) *Initialization: Choose $W^0 \in \Re^{\ell\times m}$ and $D^0 \in \mathcal{S}^{\ell}_{+}$. Let $k = 0$.*

(S.1) *Set $V = W^k$ and update $W^k$ by*

$$W^{k+1} = \arg\min_{W\in\Re^{\ell\times m}} Q_m(W,V,D^k). \tag{19}$$

(S.2) *Update $D^k$ by*

$$D^{k+1} = \arg\min_{D\in\mathcal{S}^{\ell}_{+}} Q(W^{k+1},D). \tag{20}$$

The following remarks are useful in understanding this algorithm.

(i) We note that the update $D^{k+1}$ in (20) also satisfies

$$\begin{aligned} D^{k+1} &= \arg\min_{D\in\mathcal{S}^{\ell}_{+}} \gamma\langle W^{k+1}(W^{k+1})^T, D^{\dagger}\rangle \\ &= \arg\min_{D\in\mathcal{S}^{\ell}_{+}} Q_m(W^{k+1}, W^k, D). \end{aligned}$$

This view puts Algorithm III.4 in the general framework of the block majorization method studied by de Leeuw [9] when specialized to (17). This justifies why we call the algorithm the iterative block-majorization method. General convergence properties of Alg. III.4 can be similarly stated as in [9], to which we refer the interested reader for detailed analysis.

(ii) $D^{k+1}$ can be computed through formula (18) with $W = W^{k+1}$. The computation of $W^{k+1}$ is equivalent to solving the following equation:

$$\left(C + \gamma(D^k)^{\dagger}\right) W = B(W^k)W^k \tag{21}$$

with the positive semidefinite coefficient matrix $(C + \gamma(D^k)^{\dagger})$.

(iii) In our implementation, we terminated the algorithm whenever there was no significant change in $W$ or in $P(W)$. That is, whenever

$$\frac{\|W^{k+1} - W^k\|}{l^2} \le \epsilon$$

or

$$\frac{|P(W^{k+1}) - P(W^k)|}{|P(W^k)|} \le \epsilon$$

for a small tolerance $\epsilon > 0$, we stop the algorithm.

(iv) Alg. III.4 can be straightforwardly applied to (9) by replacing $D$ by $\mathrm{Diag}(\lambda)$ and updating $\lambda$ by formula (10).

## IV. Numerical Experiments

In this section, we first present a practical two-stage algorithm that utilizes Alg. III.4. We then test the algorithm against three well-known benchmarking dataset *iris data set*, *cancer data set* and *seed data set*, all from UCI machine learning repository[1]. We will demonstrate the effectiveness of our algorithm against Webb's approach [25].

### A. A Two-Stage Algorithm

The strong motivation in using the $(2,1)$-norm $\|W\|_{2,1}$ in problem (7) is that the more important a center $\mathbf{c}_i$ is, the farther away of the $i$th row of $W$ should be from origin. In other words, if the center $\mathbf{c}_i$ is more important than the center $\mathbf{c}_j$, it is then expected from the $(2,1)$-norm regularization that

$$\|W_{i:}\| > \|W_{j:}\|.$$

This immediately suggests the following heuristic procedure for selecting the most important centers.

Suppose $W \in \Re^{\ell \times m}$ is the final iterate of Alg. III.4. We compute the length of each row of $W$: $\{\|W_{1:}\|, \ldots, \|W_{\ell:}\|\}$. We sort the sequence in decreasing order and denote the resulting sequence by

$$\{t_1,\ t_2, \ldots, t_\ell\} \quad \text{and} \quad T = \sum_{j=1}^{\ell} t_i,$$

where $T$ is the total length of the sequence. Without loss of generality, we denote the corresponding sequence of centers by $\mathbf{c}_1, \ldots, \mathbf{c}_n$. The interpretation is that the centers are arranged in the order of decreasing importance. We then compute the cumulated percentage of the total length by the leading centers in the sequence:

$$p_i = \frac{\sum_{j=1}^{i} t_j}{T}, \quad i = 1, \ldots, \ell.$$

Obviously, $\{p_i\}$ is increasing and $p_\ell = 1$. Let $p$ be a pre-set high percentage (e.g., $p = 95\%$) and Choose

$$\ell_0 = \min\left\{i :\ p_i \ge p,\ i = 1, \ldots, n\right\}. \tag{22}$$

We may think that the first $\ell_0$ centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_{\ell_0}\}$ account at least $p$ percentage of the total effectiveness contributed by the $\ell$ centers. We expect that $\ell_0$ would be much less than $\ell$.

Having selected the $\ell_0$ effective centers by (22), we proceed to solve the following optimization problem:

$$\min_{W \in \Re^{\ell_0 \times m}} \ \sigma^2(W). \tag{23}$$

We note that problem (23) is of the type of Webb's problem (4), but in a reduced dimension because $\ell_0 < \ell$. We summarize this two-stage algorithm as follows.

**Algorithm IV.1. Two-Stage Algorithm**

S.1 *Apply Alg. III.4 to get its final iterative matrix $W \in \Re^{\ell \times m}$. Use (22) to select the most important $\ell_0$ centers.*

S.2 *Apply the iterative block majorization algorithm of Webb [25] to solve problem (23).*

### B. Parameter Setting and Performance Indicators

In the numerical experiment, the weight matrix $W$ was initialized with random values, where $W_{ij}$ are distributed uniformly over the range $[0, 1]$. The tolerance $\epsilon = 10^{-4}$ is chosen for terminating the both stages of Alg. IV.1 by the rules in Remark (iii) on Alg. III.4. The bandwidth parameter $h^2 = 10.0$ is taken from [25]. $\alpha_{ij}$ were taken to be unity. The penalty parameter $\gamma$ is 1. Singular Value Decomposition is used to calculate the pseudoinverse of the matrices. We set $p = 95\%$ in (22). For each of the data sets, a random of $20\%$ of the data was initially selected as centers. In order to speed up our algorithm, the maximum number of iterations for the first stage in Alg. IV.1 is set at $\lfloor 0.2N \rfloor$, where $N$ is the number of data samples in the data set and $\lfloor 0.2N \rfloor$ is the largest integer not greater than $0.2N$. Throughout, we set $m = 2$, which means that the original data was scaled to a data set in 2 dimensions.

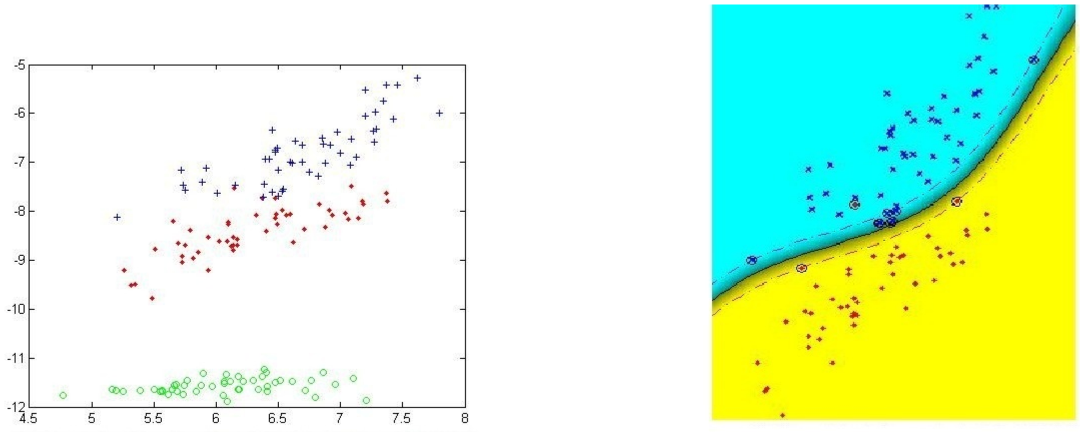
Two versions of Alg. IV.1 were compared with Webb's majorization algorithm [25], which is denoted by `MDS-M` for ease of comparison. One version refers to the case when the diagonal model (9) is used in (S.1) of Alg. IV.1. We denote this version by `RMDS-D`. The other version refers to the case when the spectral model (13) is used and is denoted by `RMDS-S`. We applied the three algorithms to each of the data sets. The results presented below were the average results on 100 runs, each of which had independent random initialization of the parameters (i.e., $W$ and centers) involved. Four quantities were calculated: `It` (number of iterations), $\sigma^2$ (the final stress), $\sigma_n^2$ (the final normalized stress), and `cpu` (time used). The normalized stress is widely used and its definition can be found in [4, p.42, Eq. (3.10)] (see the comments therein for justification of this quantity in explaining data):

$$\sigma_n^2(W) = \frac{\sum_{i,j=1}^{N} \alpha_{ij} (q_{ij}(W) - d_{ij})^2}{\sum_{i,j=1}^{N} d_{ij}^2}.$$

### C. Numerical Performance

In this subsection, we will demonstrate the good performance of Alg. IV.1 on the selected datasets, each of which will be projected to a 2-dimensional dataset (i.e., $m = 2$). We are going to use a number of graphs to show its behavior in `CPU` time, normalized stress as well as stress values. We will also take a further step to apply existing support vector machine (SVM) algorithms in [23] to the obtained 2-dimensional datasets to show the significant improvement over Webb's model. In order to shorten the paper, we will omit

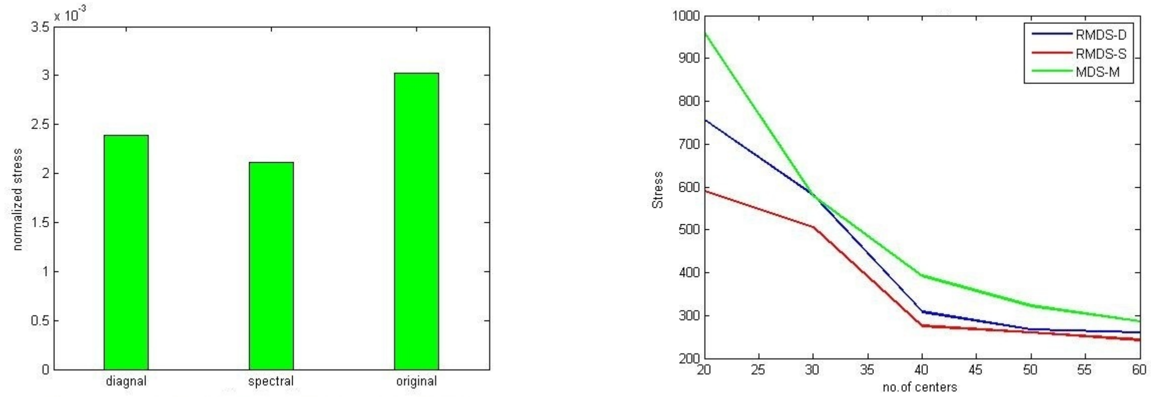
[1] http://archive.ics.uci.edu/ml/



(a) 2-dimensional projection of Iris data using RMDS−S

(b) SVM on Iris data projected by RMDS−S

Fig. 1. Fig. 1(a) is the projected 2-dimensional Iris data, consisting of 3 classes. One class represented by ”o” is completely separated from the other two, represented by ”+” and “◇”. Fig. 1(b) shows the separation of the nonseparable two classes by a support vector machine algorithm. Over 100 runs, our model (e.g., RMSD−S) yielded about 10 to 14 misclassified points, while the corresponding number for Webb's model is 16 to 20.



(a) Average normalized stress over 100 runs on Iris data

(b) Comparison of stress values of Iris data against selected centers

Fig. 2. Fig. 2(a) is the comparison of the average normalized stress values for the three models RMSD−D, RMSD−S and MDS−M over 100 random runs with 30 selected centers. Fig. 2(b) is the comparison of stress values when the number of centers ($\ell$) varies.

the SVM graphs for the Seeds data, where the one-against-all SVM algorithm (because the Seeds data has 3 classes) would need 3 graphs to demonstrates all the cases. But we will include some comments on those omitted graphs. All tests were carried out using the 64-bit version of MATLAB R2013a on a Windows 7 desktop with 64-bit operating system having Intel(R) Core(TM) 2 Duo CPU of 3.16GHz and 4.0GB of RAM.

**(a) Iris Data**. It is a very known data set used in pattern recognition literature. This data set consists of data from three classes, each has 50 samples. Each data item consists of four different real values and each value represents an attribute of each instance such as length and width of sepal or petal. One class is known to be linearly separable from the other two, which are not linearly separable from each other. Our purpose is to represent this 4-dimensional dataset as a 2-dimensional dataset.

For this purpose, we started with randomly selected 30 initial centers. At the first stage, our methods RMDS−D and RMDS−S select an average of 20-24 centers. 2-dimensional projection of Iris data is shown in Fig. 1(a), which clearly shows that one class is totally separable from other two classes. SVM algorithm [22, Sect. 18, Chap. 4] is applied to the two non-separable classes. Our models yielded an average of 10 to 14 of misclassified points, while for the original model this number is between 16 and 20. Fig. 1(b) illustrates SVM classification of Iris data obtained by RMDS−S. General performance information on 100 random run on the dataset can be found in Table I.

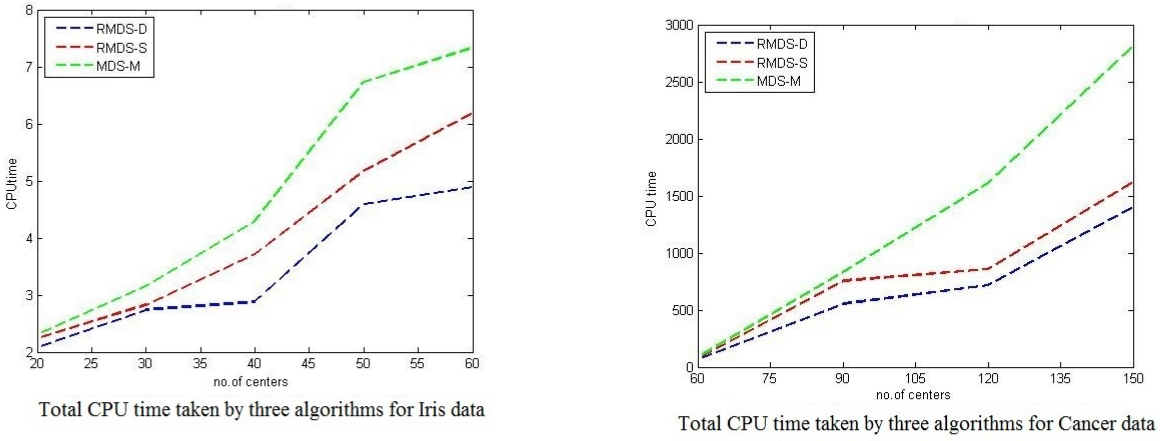
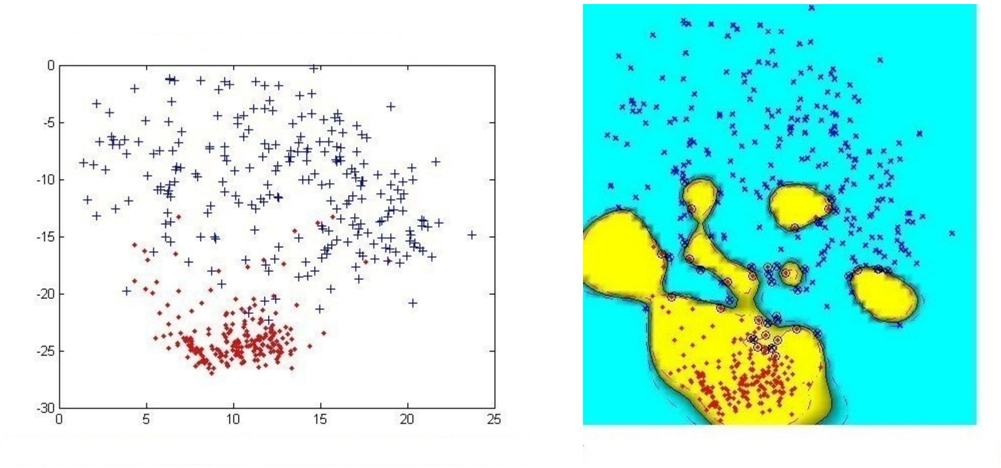


Fig. 3. CPU time comparison by RMDS-D, RMDS-S, and MDS-M on Iris and Cancer datasets when the number of centers varies



(a) 2-dimensional projection of Cancer data by RMSD-S

(b) SVM on Cancer data projected by RMDS-S

Fig. 4. Fig. 4(a) shows the Cancer data set projected in two dimensional space by RMDS-S. Fig. 4(b) shows the SVM separation on the projected Cancer data.

TABLE I. AVERAGE PERFORMANCE OF 100 RUNS FOR IRIS DATA

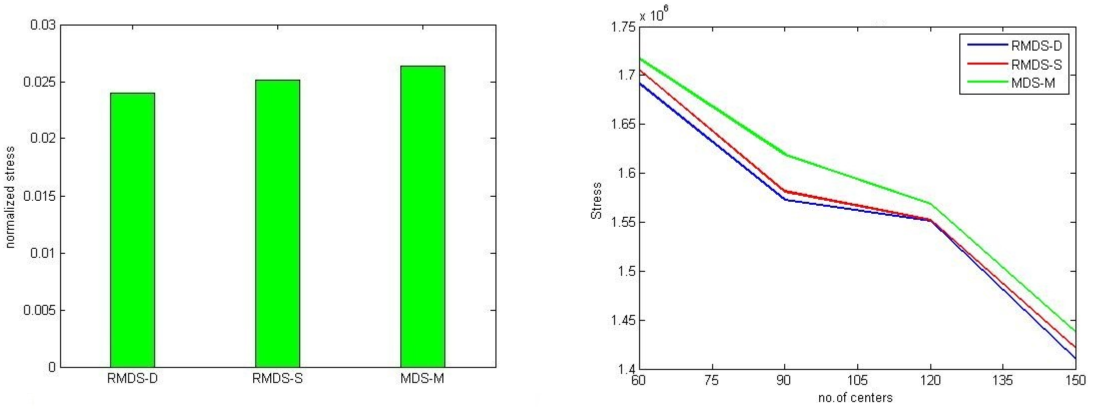
| Method | CPU Time (sec) | Iteration | Stress | Normalized stress |
|---|---|---|---|---|
| RMDS-D | 3.28 | 71.50 | 487.67 | 0.0024 |
| RMDS-S | 4.03 | 92.30 | 432.52 | 0.0021 |
| MDS-M | 4.02 | 112.10 | 617.15 | 0.0030 |

In Fig. 2(a), The mapping quality of the constructed configurations of Iris data by RMDS-D, RMDS-S and MDS-M is compared in terms of the average normalized stress values among 100 random runs each selecting 30 centers out of 60 random data points. Numerically, RMDS-D and RMDS-S improve mapping quality by 20% and 30% over MDS-M respectively in terms of the average stress value, which can be verified from Table I.

Fig. 2(b) illustrates that the proposed methods outperformed MDS-M in terms of stress value when the same number ($\ell$) of centers were selected from 100 random data points. The stress value decreases as the number of center increases for each of the three methods. CPU times taken by the three algorithms were plotted in Fig. 3.

**(b) Cancer Data**. The cancer data set is another well-known data set used by many researchers. It has two classes (benign and malignant). Each data item consists of 11 columns and the first and the last column respectively represents ID number and class information of the item. The remaining 9 columns are attribute values described in integer from 1 to 10. It contains 699 data items and 16 of them have some missing values. So we used 683 data items which have every attribute values. For this data set, the proposed algorithm selects an average of 51-53 effective centers from 60 randomly selected centers. The two dimensional projection of the 9 dimensional dataset using RMDS-S is given in Fig. 4.

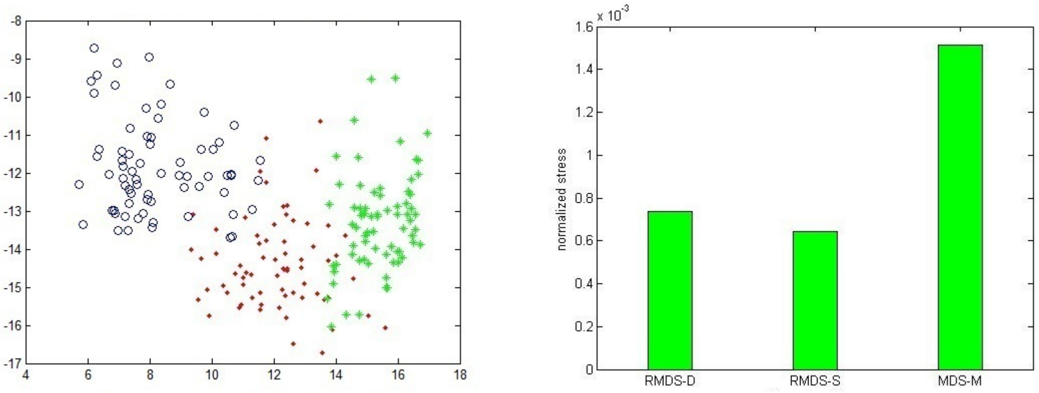
The number of misclassified vectors for this dataset projected by proposed methods is 53-55 whereas for the original model this number is 62-65. This shows that our methods improves the projection of the data and can separate the points



(a) Average normalized stress over 100 runs on Cancer data

(b) Comparison of stress values of Cancer data when the number of centers changes

Fig. 5. Fig. 5(a) is the comparison of the average normalized stress values for the three models RMSD−D, RMSD−S and MDS−M over 100 random runs with 60 selected centers. Fig. 5(b) is the comparison of stress values when the number of centers ($\ell$) varies.



(a) 2-dimensional projection of Seeds data by RMSD−S

(b) Average normalized stress over 100 runs on Seeds data

Fig. 6. Fig. 6(a) is 2-D projection of Seeds data. Fig. 6(b) shows comparison of the average normalized stress values for the three models RMSD−D, RMSD−S and MDS−M over 100 random runs with 40 selected centers.

of different classes better than the original model would do. Table II compares the average performance of 100 runs of the three methods for the cancer data using 60 centers out of 100 randomly selected points. It can be seen that though the proposed methods take a little more time than the original method, both the stress and the normalized stress values (Fig. 5) by RMDS−D and RMDS−S are lower than that by the original method.

TABLE II. AVERAGE PERFORMANCE OF 100 RUNS FOR CANCER DATA

| Method | CPU Time (sec) | Iteration | Stress | Normalized stress |
|---|---|---|---|---|
| RMDS-D | 264.5786 | 105.4 | 1.5862 $e^{06}$ | 0.0240 |
| RMDS-S | 231.1361 | 89.3 | 1.6636 $e^{06}$ | 0.0251 |
| MDS-M | 217.8957 | 103.0 | 1.7446 $e^{06}$ | 0.0264 |

**(c) Seeds Data** The seed data set is composed of 210 entities and each entity is represented by 7 real-valued attributes in addition to the class level contained in the last column. There are three classes, 70 points in each, representing three different varieties of wheat: Kama, Rosa and Canadian. We have selected 40 centers initially and the number of effective centers selected by our algorithm is 32-35.

As there are three classes, we applied *one-against-all* support vector machine algorithm to determine the misclassified data. The numbers of misclassified points by our algorithm are respectively 33 to 38, 16 to 17, 20 to 21. The corresponding numbers for the original model are 39-44, 19-21, 23-24. The normalized stress value comparison is illustrated in Fig. 6(b). The bar graph illustrates the average normalized stress value of 100 runs with 40 selected centers from 80 random initial points obtained by RMDS−D, RMDS−S and MDS−M. Our methods improve about 54-60% over the original model, which can also be verified from Table III. We note that for each of the tested datasets, as the number of center increases, our methods with a high percentage of selections (e.g., 95%) are less time consuming than MDS−M. This is demonstrated in Fig. 3 and 7.

We conclude this section by noting that the spectral model

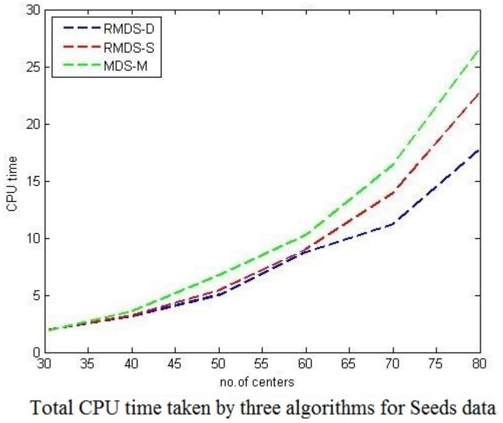


Fig. 7. Total CPU time taken by the three algorithms starting with different number of initial centers of Seeds data.

RMDS-S, when compared to the diagonal model RMDS-D, is less sensitive to the choice of the regularization parameter $\gamma$. For example, when $\gamma = 10$, there appeared a significant level of failure in RMDS-D in all three data sets, while RMDS-S worked almost same as we reported in this paper.

TABLE III. AVERAGE PERFORMANCE OF 100 RUNS FOR SEEDS DATA

| Method | CPU Time (sec) | Iteration | Stress | Normalized stress |
|---|---|---|---|---|
| RMDS-D | 11.54 | 62.00 | 843.7 | 0.0007 |
| RMDS-S | 12.35 | 68.00 | 732.7 | 0.0006 |
| MDS-M | 8.72 | 59.20 | 1727.1 | 0.0015 |

## V. CONCLUSION

In this paper, we have addressed a key problem in selecting the effective centers for a multidimensional scaling method, which involves radial basis functions. We took a novel approach that casts the problem as a multi-task learning problem. This approach has led to introducing the $(2, 1)$-norm as a regularization term to the stress function used by Webb [25]. We then developed two reformulations, namely the diagonal and the spectral, that aim to ease the difficulties in solving the $(2, 1)$-norm minimization problem. The two reformulation models were compared to the original model in [25] on three well-known data sets. Numerical results illustrate significant improvement. We would like to emphasize that the spectral model is more robust than the diagonal model, but with higher computational complexity.

However, the current algorithmic implementation of Alg. III.4 has its limitations on large data sets as the singular value decompositions were used to solve the linear equations encountered. It is therefore necessary to explore alternative algorithms that can handle larger data sets. We also note that the models do not take any advantages of some priori information concerning the data sets. For example, some data points may be known beforehand to belong to certain class. Hence, it would be interesting to include a discriminate analysis in our models as has already been done by Webb [25] and others. We leave those topics to our future research.

## REFERENCES

[1] A. Argyriou, T. Evgeniou, and M. Pontil, *Multi-task Feature Learning.* In B. Schoelkopf, J. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems, MIT Press, 2007.

[2] A. Argyriou, T. Evgeniou, and M. Pontil, *Convex Multi-task Feature Learning.* Machine Learning, Special Issue on Inductive Transfer Learning, 73 (2008), 243-272.

[3] J. BÉNASSÉNI, *Partial additive constant*, J. Statist. Comput. Simul. 49 (1994), pp. 179–193.

[4] I. Borg and P.J.F. Groenen, *Modern Multidimensional Scaling. Theory and Applications (2nd Ed.).* Springer Series in Statistics, Springer, 2005.

[5] F. Cailliez, *The analytical solution of the additive constant problem*, Psychometrika 48 (1983), 305–308.

[6] L.G. Cooper, *A new solution to the additive constant problem in metric and multidimensional scaling*, Psychometrika 37 (1972), 311–321.

[7] T.F. Cox and M.A. Cox, *Multidimensional Scaling (2nd Ed.)* Chapman and Hall/CRC (2002).

[8] J. de Leeuw, *Applications of convex analysis to multidimensional scaling*, in J. Barra, F. Brodeau, G. Romier, and B. van Cutsen, eds, 'Recent Developments in Statistics', North Holland Publishing Company, Amsterdem, The Netherlands, pp. 133–145.

[9] J. de Leeuw, *Block relaxation algorithms in statistics*. In: Bock, H.H. et al. (eds) Information Systems and Data Analysis, pp. 308–325, Springer, Berlin (1994).

[10] W. Glunt, T.L. Hayden, S. Hong, and J. Wells, *An alternating projection algorithm for computing the nearest Euclidean distance matrix*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 589–600.

[11] W. Glunt, T.L. Hayden, and R. Raydan, *Molecular conformations from distance matrices*, J. Computational Chemistry, 14 (1993), pp. 114–120.

[12] J.C. Gower, *Some distance properties of latent rootand vector methods in multivariate analysis*, Biometrika, 53 (1966), 315–328.

[13] J. Kruskal, *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, 29 (1964), 1–27.

[14] K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis* (Tenth printing). Academic Press, 1995.

[15] S.J. Messick and R.P Abelson, *The additive constant problem in multidimensional scaling*, Psychometrika 21 (1956), pp. 1–15.

[16] E. Pękalaska and R.P.W.Duin *The Dissimilarity Representation for Pattern Recognition: Foundations and Application*, Series in Machine Perception Artificial Intelligence 64, World Scientific 2005.

[17] H.-D. Qi, *A semismooth Newton method for the nearest Euclidean distance matrix problem*, SIAM Journal Matrix Analysis and Applications, 34 (2013), 67–93.

[18] H.-D. Qi and N. Xiu, *A convex quadratic semidefinite programming approach to the partial additive constant problem in multidimensional scaling*, Journal of Statistical Computation and Simulation, 82 (2012), 1317–1336.

[19] H.-D. Qi, N.H. Xiu, and X.M. Yuan, *A Lagrangian dual approach to the single source localization problem*, IEEE Transactions on Signal Processing, 61 (2013), 3815–3826.

[20] H.-D. Qi and X.M. Yuan, *Computing the nearest Euclidean distance matrix with low embedding dimensions*, Mathematical Programming, DOI 10.1007/s10107-013-0726-0.

[21] I.J. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la définition axiomatque d'une classe d'espaces vectoriels distanciés applicbles vectoriellement sur l'espace de Hilbet"*, Ann. Math. 36 (1935), pp. 724–732.

[22] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Elsevier Inc., 2009.

[23] S. Theodoridis and K. Koutroumbas. *An Introduction to Pattern Recognition, A MATLAB approach*. Elsevier Inc., 2010.

[24] W.S. Torgerson, *Theory and Methods for Scaling*. Wiley, New York, 1958.

[25] A.R. Webb, *Multidimensional Scaling by iterative majorization using radial basis functions.* Pattern Recognition, 28 (1995), 753-759.

[26] A.R. Webb, *Nonlinear feature extraction with radial basis functions using a weighted multidimensional scaling stress measure* Pattern Recognition, IEEE Conference Publications 4 (1996), 635-639.

[27] A.R. Webb, *An approach to nonlinear principal component analysis using radially-symmetric kernel functions.* Statistics and Computing, 6 (1996), 159-168.

[28] G. Young and A.S. Householder, *Discussion of a set of points in terms of their mutual distances*, Psychometrika 3 (1938), pp. 19–22.