Classical Multidimensional Scaling: A Subspace Perspective, Over-Denoising and Outlier Detection

Lingchen Kong, Chuanqi Qi and Hou-Duo Qi

***Abstract*—The classical Multi-Dimensional Scaling (cMDS) has become a cornerstone for analyzing metric dissimilarity data due to its simplicity in derivation, low computational complexity and its nice interpretation via the principle component analysis. This paper focuses on its capability of denoising and outlier detection. Our new interpretation shows that cMDS always overly denoises a sparsely perturbed data by subtracting a fully dense denoising matrix in a subspace from the given data matrix. This leads us to consider two types of sparsity-driven models: Subspace sparse MDS and Full-space sparse MDS, which respectively uses the ℓ_1 and ℓ_{1-2} regularization to induce sparsity. We then develop fast majorization algorithms for both models and establish their convergence. In particular, we are able to control the sparsity level at every iterate provided that the sparsity control parameter is above a computable threshold. This is a desirable property that has not been enjoyed by any of existing sparse MDS methods. Our numerical experiments on both artificial and real data demonstrates that cMDS with appropriate regularization can perform the tasks of denoising and outlier detection, and inherits the efficiency of cMDS in comparison with several state-of-the-art sparsity-driven MDS methods.**

***Index Terms*—Classical multidimensional scaling, Euclidean distance matrix, sparse optimisation, ℓ_1 and ℓ_{1-2} regularization, majorization.**

I. Introduction

THE classical Multi-Dimensional Scaling (cMDS) has become a cornerstone for analysing metric data commonly known as (metric) dissimilarity data. cMDS and its variants (metric MDS) have been well documented in the two books [1], [2]. It was initially studied by Schoenberg [3] and Young and Householder [4] when the dissimilarities are Euclidean distances. And for this case, it is later discovered by Gower [5] to be equivalent to Principle Component Analysis (PCA) provided that the covariance matrix used by PCA is calculated from the same data for the Euclidean distances. Thus, Gower named cMDS Principle Co-ordinates Analysis. cMDS also became an essential element in the nonlinear dimensionality reduction method ISOMAP [6]. The purpose of this paper is to study the capability of cMDS in detecting outliers under the framework of denoising. Our major observation is a new optimization interpretation of cMDS having a tendency of over-denoising. To overcome this drawback we propose two sparse variants of cMDS, namely the subspace sparse MDS and the full-space sparse MDS, which will greatly enhance the capability of cMDS in outlier detection.

Suppose there are n items and their pairwise Euclidean distances d_{ij} can be measured through the pairwise dissimilarities δ_{ij}, i.e., $\delta_{ij} \approx d_{ij}$. cMDS is a simple computational procedure to generate a set of n points $\mathbf{y}_i \in \Re^r$ such that

$$d_{ij}^2 := \|\mathbf{y}_i - \mathbf{y}_j\|^2 \approx \ \delta_{ij}^2, \quad i,j = 1, \ldots, n, \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm and ":=" means "define". In practice, the embedding dimension r is small (e.g., $r = 2$ or 3 for visualization).

When each δ_{ij} is a true Euclidean distance from a set of n points, cMDS will recover a set of embedding point $\mathbf{y}_i$ such that $\|\mathbf{y}_i - \mathbf{y}_j\| = \delta_{ij}$, $i,j = 1,\ldots,n$. If some δ_{ij} contains noise, e.g., $\delta_{ij} = d_{ij} + \epsilon_{ij}$ with ϵ_{ij} being the corresponding noise, then cMDS works well when the noise is small. A theoretical justification for using cMDS in such a situation can be found in Sibson [7] based on a perturbation analysis. However, when some δ_{ij} takes the form: $\delta_{ij} = d_{ij} + \epsilon_{ij} + \eta_{ij}$ with η_{ij} being large measurement error (such δ_{ij} is deemed to be of outlier), the quality of cMDS alarmingly degrades because it would spread the large error $(\epsilon_{ij} + \eta_{ij})$ to all other δ_{ij}. This phenomenon has been highlighted in [8] and motivated Forero and Giannakis [9] to propose a sparsity-exploiting robust MDS (RMDS) for outlier removal. It makes use of the Kruskal stress function [10] as MDS criterion with ℓ_1-based regularizations, a sparsity-induced technique used in machine learning and compressed sensing. To improve the robustness of RMDS, Mandanas and Kotropoulos [11] replaced the least-square solution of the residual equation at each step of RMDS by a M-estimator, resulting in several robust algorithms depending on the M-estimator being used. We refer to [12] for further development along this line, in particular on using ℓ_{21} regularization. We note that the models behind those methods are non-convex optimization.

ℓ_1 regularized methods also appeared in the field of sensor network localization with non-light-of-sight (NLOS) distance measurements, see e.g., [13]–[15]. NLOS measurements occur when the LOS (line-of-sight) path is blocked due to environmental limitations such as the indoor environment depicted in the example of locating Motorola facilities [16]. We refer to the papers [13], [17] and references therein for diverse models in handling different scenarios involving NLOS links. A typical feature among NLOS links is that the measured (metric) dissimilarities δ_{ij} is significantly larger than the true distances d_{ij} and their locations are usually unknown. Hence, such links can be treated as outliers. A dominating approach is

First version: November 6, 2018, Revised March 22, 2019.

L. Kong is with the Department of Applied Mathematics, Beijing Jiaotong University, Beijing, China. Email: lchkong@bjtu.edu.cn.

C. Qi is with School of Mathematics, University of Southampton. Email: cq1e10@soton.ac.uk.

H.-D. Qi is with School of Mathematics, University of Southampton, UK. Email: hdqi@soton.ac.uk.

the convex relaxation/optimization, which often involves semidefinite programming (SDP) with ℓ_1 regularization, see [13]–[15] and [17].

In this paper, we develop an entirely different approach for outlier detection and removal. We begin with asking an important question why cMDS fails to accomplish those tasks. We provide a mathematically precise explanation for this known phenomenon [8]. The culprit is that cMDS always subtracts a dense matrix from the squared dissimilarity matrix $\overline{\Delta} := (\delta_{ij}^2)$ before computing a set of embedding points (see Thm. 3.1). This result reveals the true mechanism behind the computational formula of cMDS [1], [2]. This detour to the desired purpose in (1) is bad because cMDS would punish every δ_{ij} even there is only one of them being outlier. Moreover, the dense matrix belongs to a subspace of rank-2 matrices. This motivates us to enforce sparsity within this subspace, leading to what we call a subspace sparse MDS model (SSMDS). We will show that SSMDS is particularly useful for the problem of single source localization [14], [18], [19]. When the outliers do not have any structural pattern, it is reasonable to extend the sparsity from the subspace to the whole space and this consideration leads to a full-space sparse MDS model (FSMDS). For both models, we use ℓ_1-based regularization to induce the sparsity.

In addition to the new interpretation of cMDS discussed above, its implications to denoising and the two sparse models (SSMDS and FSMDS), we highlight the other major contributions below.

(i) We develop fast algorithms for the two models by making use of the majorization-minimization technique and the elegant properties of Euclidean distance matrices (EDM). We establish the global convergence of the proposed methods, see Thm. 5.1.

(ii) We are able to control the sparsity level in every step of our calculation, thanks to the ℓ_1-based regularization coupled with the nice objective function of cMDS, see Thm. 5.2. This is in contrast to the ℓ_1-regularized methods in [9], [11], [12] where it still remains unknown how to control the sparsity level.

(iii) Numerically, we demonstrate the capability and efficiency of the proposed methods in denoising and outlier detection in comparison with the state-of-the-art MDS methods, using both artificial and real test data.

The powerful framework of our study is through the Euclidean distance matrix optimization, which is drastically different from the studies in [9], [11], [12], where the coordinates descent optimization was employed. However, they share a same feature that both approaches are of non-convex optimization. The paper is organized as follows. In next section, we describe the necessary background on EDMs for proving our new reformulation of cMDS (Thm. 3.1) in Sect. III. The SSMDS and FSMDS model are respectively treated in Sect. IV and Sect. V, which also include a complete set of convergence analysis (Thm. 5.1) and the sparsity-control theorem (Thm. 5.2). Numerical experiments are reported in Sect. VI. The paper concludes in Sect. VII.

II. Background, EDM and cMDS

This section includes the necessary background for proving our main theorems and for developing the fast algorithms later on. The key concept is the Euclidean Distance matrix (EDM). Due to the space limitation, we are only able to give a brief introduction of EDM. We refer to [19], [21], [22] for a more detailed account. We set up common notation first.

A. Notation

Throughout, we use boldfaced letters to denote (column) vectors (e.g., $\mathbf{x} \in \Re^n$ is a column vector, its ith element is x_i, and its transpose $\mathbf{x}^T$ is a row vector). In particular, $\mathbf{1}$ is the vector of all ones in $\Re^n$. $\|\cdot\|$ is the Euclidean norm in $\Re^n$ and the ℓ_1-norm is $\|\mathbf{x}\|_1 = |x_1| + \cdots + |x_n|$. Let $\mathcal{S}^n$ denote the space of $n \times n$ symmetric matrices, endowed with the standard trace inner product. The induced norm is the Frobenius norm $\|\cdot\|$. For a matrix $A \in \mathcal{S}^n$, we often use A_{ij} to denote its (i,j)th element, with the exception of the dissimilarity matrix Δ consisting of δ_{ij} (to follow the tradition in MDS [2]).

We let $\mathcal{S}^n_+$ denote the cone of all positive semidefinite matrices in $\mathcal{S}^n$. For a closed and convex set $\mathcal{C}$ in $\mathcal{S}^n$, $\Pi_{\mathcal{C}}(A)$ denotes the orthogonal projection of a given matrix $A \in \mathcal{S}^n$ onto $\mathcal{C}$:

$$\Pi_{\mathcal{C}}(A) := \arg\min \left\{ \|A - X\| : \quad X \in \mathcal{C} \right\}.$$

In our algorithmic development, the soft-thresholding operator is important to us. Consider the one-dimensional quadratic problem:

$$\min_{x \in \Re} \ \frac{1}{2}(x-t)^2 + \beta|x|,$$

where $t \in \Re$ and $\beta > 0$ are given. Its optimal solution is given by the thresholding operator [23]

$$\mathcal{S}_\beta(t) := \max\{|t| - \beta, \ 0\} \mathrm{sgn}(t), \tag{2}$$

where sgn is the sign function.

B. Euclidean Distance Matrix

We say that a matrix $D \in \mathcal{S}^n$ is an Euclidean Distance Matrix (EDM) if there exist a set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \Re^p$ from some positive integer p such that the (i,j)th element of D is given by the squared Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad i,j = 1, \ldots, n. \tag{3}$$

The set of all $n \times n$ EDMs forms a closed convex cone, denoted by $\mathcal{D}^n$. For a given EDM $D \in \mathcal{D}^n$, the smallest dimension p such that (3) holds is known as the embedding dimension of D and it is $r = \mathrm{rank}(JDJ)$, where J is the centralizing matrix: $J := I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ with I being the identity matrix in $\mathcal{S}^n$. One characterization of EDM is due to [3]:

$$D \in \mathcal{D}^n \quad \text{if and only if} \quad \mathrm{diag}(D) = 0, \quad -D \in \mathcal{K}^n_+, \tag{4}$$

where $\mathcal{K}^n_+$ is the conditionally positive semidefinite cone:

$$\mathcal{K}^n_+ := \left\{ A \in \mathcal{S}^n \mid \mathbf{v}^T A \mathbf{v} \geq 0, \ \forall \ \mathbf{v} \in \Re^n, \ v_1 + \cdots + v_n = 0 \right\}.$$

By using the centralizing matrix J, we have

$$\mathcal{K}_+^n = \left\{ A \in \mathcal{S}^n \mid JAJ \in \mathcal{S}_+^n \right\}. \tag{5}$$

The projection onto $\mathcal{K}_+^n$ can be calculated by the formula of [24, Eq.(29)]:

$$\Pi_{\mathcal{K}_+^n}(A) = A + \Pi_{\mathcal{S}_+^n}(-JAJ), \quad \forall\ A \in \mathcal{S}^n. \tag{6}$$

We are also able to compute how "close" a given EDM D has an required embedding dimension. Let $\mathcal{K}_+^n(r)$ denote the set of all matrices in $\mathcal{K}_+^n$ with the embedding dimension not greater than r:

$$\mathcal{K}_+^n(r) := \left\{ D \in \mathcal{K}_+^n \mid \mathrm{rank}(JDJ) \le r \right\}. \tag{7}$$

We call it the rank-r cut of the conditionally positive semidefinite cone. It is extensively studied in [20], [22]. Let $A \in \mathcal{S}^n$ be given, define the distance from A to $\mathcal{K}_+^n(r)$:

$$\mathrm{dist}(A,\ \mathcal{K}_+^n(r)) := \min\{\|A - D\| : \quad D \in \mathcal{K}_+^n(r)\},$$

and define the squared distance function

$$g_r(A) := \frac{1}{2}\mathrm{dist}^2(-A,\ \mathcal{K}_+^n(r)). \tag{8}$$

Obviously, $-A \in \mathcal{K}_+^n(r)$ if and only if $g_r(A) = 0$. The following characterization will be used when we come to designing our algorithm:

$$\begin{aligned} & D \in \mathcal{D}^n,\ \mathrm{rank}(JDJ) \le r \\ \overset{(4)}{\Longleftrightarrow}\quad & \mathrm{diag}(D) = 0,\ -D \in \mathcal{K}_+^n,\ \mathrm{rank}(JDJ) \le r \\ \overset{(7)}{\Longleftrightarrow}\quad & \mathrm{diag}(D) = 0,\ -D \in \mathcal{K}_+^n(r) \\ \overset{(8)}{\Longleftrightarrow}\quad & \mathrm{diag}(D) = 0,\ g_r(D) = 0. \end{aligned} \tag{9}$$

Moreover, [22, Lemmas 2.1, 2.2] implies that the function

$$h(A) := \frac{1}{2}\|A\|^2 - g_r(-A) \tag{10}$$

is convex and we can calculate one of its subgradients by

$$\Pi_{\mathcal{K}_+^n(r)}(A) \in \partial h(A), \tag{11}$$

where $\Pi_{\mathcal{K}_+^n(r)}(A)$ denotes a projection of A onto $\mathcal{K}_+^n(r)$. We will address how to compute $\Pi_{\mathcal{K}_+^n(r)}(A)$ in the numerical part.

It follows from (10), the convexity of $h(\cdot)$ and (11) that

$$\begin{aligned} g_r(D) &= \frac{1}{2}\|D\|^2 - h(-D) \\ &\le \frac{1}{2}\|D\|^2 - h(-A) + \langle \Pi_{\mathcal{K}_+^n(r)}(-A), D - A\rangle \\ &=: g_r^m(D, A), \qquad \forall\ D, A \in \mathcal{S}^n. \end{aligned} \tag{12}$$

We call $g_r^m(D, A)$ a majorization of $g_r(D)$.

C. cMDS and Noise Spreading

We describe how cMDS computes a set of embedding points $\mathbf{y}_i$ trying to satisfy the approximation in (1) under certain optimal criterion. Let $\overline{\Delta} \in \mathcal{S}^n$ consist of $\overline{\Delta}_{ij} = \delta_{ij}^2$ (the squared dissimilarities). Compute the B-matrix and its orthogonal projection onto $\mathcal{S}_+^n$:

$$B := -\frac{1}{2}J\overline{\Delta}J, \qquad B_+ := \Pi_{\mathcal{S}_+^n}(B). \tag{13}$$

Note that J is the centering matrix. The double-centering in B was introduced to cMDS by Torgerson [25]. It further decomposes B_+ as a Gram matrix

$$B_+ = Y^T Y \quad \text{with} \quad Y := [\mathbf{y}_1, \ldots, \mathbf{y}_n], \tag{14}$$

and the embedding points are $\mathbf{y}_i \in \Re^r$, $r = \mathrm{rank}(B_+)$. The resulting EDM is

$$D^{\mathrm{mds}} = \left(\|\mathbf{y}_i - \mathbf{y}_j\|^2 \right)_{i,j=1}^n.$$

Due to its simplicity, low-computational complexity and its nice mathematical interpretation via PCA, cMDS has become a popular method [2].

The main drawback that cMDS suffers is its noise spreading, which was highlighted in [8]. For example, if there is just one δ_{ij} containing noise ϵ and a measurement error η (i.e., $\delta_{ij} = d_{ij} + \epsilon + \eta$) (all other δ_{ij} are true Euclidean distances), the double-centering operation in B (13) spreads the error ($\epsilon +$ η to every entry. This would result in poor approximation, particularly when η is caused by an outlier (η is large). In other words, if $\overline{\Delta}$ is sparsely perturbed, cMDS will spread the sparse noise everywhere. This raises the issue how to remove the sparse noise. Our new result on cMDS will show that cMDS alone is incapable of doing so.

An alternative way to derive cMDS is through the fact that D^{mds} is the solution of the optimization problem [26]:

$$D^{\mathrm{mds}} = \arg\min \|J(D - \overline{\Delta})J\|^2, \quad \text{s.t.} \quad D \in \mathcal{D}^n. \tag{15}$$

We can obtain the matrix B_+ by

$$B_+ = -\frac{1}{2}JD^{\mathrm{mds}}J \qquad (\text{also} \quad r = \mathrm{rank}(JD^{\mathrm{mds}}J). \tag{16}$$

Decomposing B_+ as in (14) to get the embedding points $\mathbf{y}_i$. As done in [27], if we define the semi-norm $\|A\|_J := \|JAJ\|$, then

$$D^{\mathrm{mds}} = \arg\min \|D - \overline{\Delta}\|_J^2, \quad \text{s.t.} \quad D \in \mathcal{D}^n.$$

However, this semi-norm does not directly measure the distance between D and $\overline{\Delta}$. A more natural matrix nearness problem is the so-called the nearest EDM problem (under the Frobenius norm $\|\cdot\|$):

$$D^{\mathrm{edm}} = \arg\min \|D - \overline{\Delta}\|^2, \quad \text{s.t.} \quad D \in \mathcal{D}^n. \tag{17}$$

We refer to [24], [28], [29] for more reading on this problem and its applications. We will see that the problems (15) and (17) sit at the each end of a class of optimization problems over a subspace.

III. New Interpretation: Over-Denoising and Sparse Remedy

In this section, we will cast cMDS as an EDM optimization problem, which will yield our first major result on a new interpretation. A direct consequence is that cMDS has a tendency of over-denoising even when the dissimilarity data has sparse outliers. This confirms the widely accepted fact that cMDS is not capable of detecting and removing outliers. Our result also motivates us to propose its sparse variants.

A. Subspace Perspective of cMDS and over-denoising

We have already seen that $D^{\rm mds}$ is the optimal solution under the semi-norm $\|\cdot\|_J$ in (15). In this part, we will show that it is also an optimal solution under the Frobenius norm $\|\cdot\|$. We let $\mathcal{S}^n_2$ be the subspace of rank-2 matrices in $\mathcal{S}^n$:

$$\mathcal{S}^n_2 := \left\{ Z \in \mathcal{S}^n \mid Z = \mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T, \ \forall\ \mathbf{z} \in \Re^n \right\}, \qquad (18)$$

and consider the optimization problem:

$$\min_{D,\ Z} \ \|(D+Z) - \overline{\Delta}\|^2 \quad \text{s.t.} \quad D \in \mathcal{D}^n, \quad Z \in \mathcal{S}^n_2. \qquad (19)$$

Our new interpretation of `cMDS` in terms of $\|\cdot\|$ is stated as follows, whose proof is in Appendix A.

Theorem 3.1: The optimization problem (19) has a unique solution $(\widehat{D}, \widehat{Z})$ and $\widehat{D} = D^{\rm mds}$. Moreover

$$\widehat{Z} = \hat{\mathbf{z}}\mathbf{1}^T + \mathbf{1}\hat{\mathbf{z}}^T,$$

where, $\hat{\mathbf{z}} := \mathbf{c} - \frac{1}{2}\bar{c}\mathbf{1}$, and

$$C := \overline{\Delta} - D^{\rm mds}, \quad \mathbf{c} := \frac{1}{n}C\mathbf{1}, \quad \bar{c} := \frac{1}{n^2}\mathbf{1}^T C\mathbf{1}.$$

Thm. 3.1 reveals what the `cMDS` is trying to achieve and provides an iterative interpretation of its simple computational steps in (13)-(14), as we explain below. For noisy $\overline{\Delta}$, `cMDS` tries to find a correction (or denoising) matrix Z and then compute the best EDM from $(\overline{\Delta} - Z)$ (i.e., replace $\overline{\Delta}$ by $(\overline{\Delta}-Z)$ in (17)). It then updates Z and repeats the process. The optimal objective in (19) is reached when $Z = \widehat{Z}$. Thm. 3.1 also explains why `cMDS` often fails to correctly identify the noisy sources when $\overline{\Delta}$ only contains a small number of contaminated entries, for instance, caused by some outliers. The matrix $C = \overline{\Delta} - D^{\rm mds} \not= 0$ unless $\overline{\Delta}$ is already an EDM. Furthermore, the resulting matrix $\widehat{Z}$ is fully dense (because $\hat{\mathbf{z}}$ is so) unless some stringent conditions are enforced. This means that `cMDS` punishes every entry even only a small number of the entries in $\overline{\Delta}$ are contaminated. Therefore, `cMDS` is blind to the sparse situation and it punishes every entry in order to remove sparse noise. We call it over-denoising. We use a simple example to illustrate this behaviour.

Example 3.2: This is a single source localization example. Suppose there are one (unknown) sensor $\mathbf{x}_1 \in \Re^2$ and three anchors $\mathbf{a}_2 = (-1, 0)$, $\mathbf{a}_3 = (1, 0)$ and $\mathbf{a}_4 = (0, 1)$. The true location of the sensor is $(0, 0)$. However, its Euclidean distances to the three anchors are contaminated and are given as $(2, 2, 2)$ (i.e., 100% error). This is the first type of outlier considered in [9] and caused by a faulty node. Therefore, $\overline{\Delta}$ is sparsely contaminated, and the matrix $S = \overline{\Delta} - D$ is sparse, where D is the true EDM of the four nodes (one sensor and three anchors). As expected, `cMDS` used a fully dense matrix $\widehat{Z}$ to approximate this sparse matrix S. The corresponding $\hat{\mathbf{z}}$ is $[0.0090, 0.1250, 0.1250, 0.3750]$ (a dense vector). In contrast, our SSMDS model below will generate a sparse vector $\hat{\mathbf{z}} = [2.7965, 0, 0, 0]$, which not only correctly detected the faulty node, but also removed a good approximation of the true contamination in magnitude $(\delta^2_{ij} - d^2_{ij} = 2^2 - 1 = 3)$.

B. Sparse remedy with practical considerations

Example 3.2 raises the question how to properly denoise when $\overline{\Delta}$ is only sparsely contaminated. This topic has been addressed by Forero and Giannakis [9] in a different context. Our answer to this question comes from Thm. 3.1 in the sense that we can enforce sparsity on the matrix Z via ℓ_1 regularization, naturally leading to the following problem:

$$\begin{array}{ll} \min_{D,\mathbf{z}} & \frac{1}{2}\|(D + \mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T) - \overline{\Delta}\|^2 + \mu\mathcal{R}_1(\mathbf{z}) \\ \text{s.t.} & D \in \mathcal{D}^n, \ \mathbf{z} \in \Re^n, \end{array} \qquad (20)$$

where $\mu > 0$ is a parameter controlling the sparsity in $\mathbf{z}$. A particular choice is the ℓ_1 regularization: $\mathcal{R}_1(z) := \|\mathbf{z}\|_1$. If $\mu = 0$, then (20) becomes `cMDS` (19), and if $\mu = +\infty$, we have $\mathbf{z} = 0$ and (20) becomes the EDM problem (17). Therefore, `cMDS` and EDM (17) stand at the two extremes of (20) with `cMDS` tending to over-denoise and EDM (17) making no attempt at all to denoise.

However, there are three practical and important issues that have been left out so far. The first issue is the embedding dimension. The regularization term $\mathcal{R}_1(\mathbf{z})$ tends to force the EDM variable D to have higher embedding dimension so as to decrease the overall objective. Therefore, we should include the embedding dimension constraint (16): $\text{rank}(JDJ) \le r$. It follows from (9) that we can represent this constraint and $D \in \mathcal{D}^n$ by $g(D) = 0$ and $\text{diag}(D) = 0$. The second issue is about the missing values in δ_{ij}. A common practice is to apply positive weights on available δ_{ij} and 0 weights on missing δ_{ij}. For example, a weight matrix $W \in \mathcal{S}^n$ can be defined as follows: $W_{ij} = 1$ for available δ_{ij} and $W_{ij} = 0$ otherwise. The third issue is the bound constraints on certain distances and they can be generally represented by

$$L_{ij} \le D_{ij} \le U_{ij} \quad \text{for some} \quad (i, j), \qquad (21)$$

where L_{ij} and U_{ij} are lower and upper bounds for the distance D_{ij}. In Example 3.2, the distances among the three anchors are known and hence they should be fixed through $L_{ij} = U_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\|^2$. Moreover, $L_{ii} = U_{ii} = 0$ represents $\text{diag}(D) = 0$.

Consideration of those three issues leads to the *Subspace Sparse MDS* (SSMDS) model below:

$$\begin{array}{ll} \min_{D,\mathbf{z}} & \frac{1}{2}\|W \circ [(D + Z) - \overline{\Delta}]\|^2 + \mu\mathcal{R}_1(\mathbf{z}) \\ \text{s.t.} & D \in \mathcal{B}, \ g_r(D) = 0, \ Z \in \mathcal{S}^n_2, \end{array} \qquad (22)$$

where $\mathcal{B} := \{D \in \mathcal{S}^n \mid L \le D \le U\}$ and $\circ$ is the Hadamard product (elementwise multiplication: $A \circ B := (A_{ij}B_{ij})$). We will show that the model (22) works very well when the sparse noises caused by few faulty nodes (outliers) such as in the single source localization [19] have a structural pattern.

When the sparse noise does not have any structural pattern, it is more reasonable to allow Z change freely in the whole space $\mathcal{S}^n$ instead of being restricted in $\mathcal{S}^n_2$. This leads to what we call the *Full-space Sparse MDS* (FSMDS) model:

$$\begin{array}{ll} \min_{D,Z} & \frac{1}{2}\|W \circ [(D + Z) - \overline{\Delta}]\|^2 + \mu\mathcal{R}_2(Z) \\ \text{s.t.} & D \in \mathcal{B}, \ g_r(D) = 0, \ Z \in \mathcal{S}^n, \end{array} \qquad (23)$$

where $\mathcal{R}_2(Z)$ is a sparsity-induced regularization such as $\|Z\|_1$. Another choice is the ℓ_{1-2} regularization: $\mathcal{R}_2(Z) := \|Z\|_1 - \|Z\|$, also used in compressed sensing [30].

The FSMDS model (23) is also relevant to the sparsity-exploiting robust MDS method [9], where Kruskal's stress function [10] (with ℓ_1 based regularizations) was used to measure the distance between the embedding distance $\|\mathbf{y}_i - \mathbf{y}_j\|$ and δ_{ij}. Due to the nondifferentiablity and nonconvexity of the stress function, a SMACOF-style [31] majorization method was developed to solve the regularized problem. In contrast, we have a differentiable objective function (not including the regularization part, whose non-differentiability is easy to deal with) and we will be able to obtain significantly more due to the simplicity of cMDS objective. The rest of the paper is devoted to solving the two models.

In our algorithmic development, we will make use of two important techniques. One is the majorization technique (see, e.g., [32]), which aims to approximate a difficult function $\theta(\cdot) : \Re^n \mapsto \Re$ by rather a simpler function (majorization function) $\theta^m(\cdot, \cdot) : \Re^n \times \Re^n \mapsto \Re$ satisfying

$$\theta^m(\mathbf{x}, \mathbf{y}) \geq \theta(\mathbf{x}) \text{ and } \theta^m(\mathbf{y}, \mathbf{y}) = \theta(\mathbf{y}), \; \forall \; \mathbf{x}, \mathbf{y} \in \Re^n. \quad (24)$$

Thus the function $g_r^m(\cdot, \cdot)$ in (12) is a majorization of $g_r(\cdot)$. The other is the penalty technique. We will penalize the constraint $g_r(D) = 0$ in both (22) and (23) to their respective objective function. This penalty approach has been recently proposed in [22] to deal with the rank constraint $\mathrm{rank}(JDJ) \leq r$ and it has been proved very effective. We also note that penalizing the squared distance function (note our $g_r(D)$ is so) is often used in statistical learning problems [34]. We will use the two techniques in the next two sections to solve the model (22) and (23) respectively.

IV. Subspace Sparse MDS

In this section, we describe an efficient alternating majorization and minimization method for (22). For ease of description, let us define

$$\begin{aligned} f(D, \mathbf{z}) &:= \frac{1}{2}\|W \circ [(D + \mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T) - \overline{\Delta}]\|^2, \\ f_\mu(D, \mathbf{z}) &:= f(D, \mathbf{z}) + \mu \mathcal{R}_1(\mathbf{z}), \\ f_{\rho,\mu}(D, \mathbf{z}) &:= f_\mu(D, \mathbf{z}) + \rho g_r(D), \end{aligned}$$

where $\rho > 0$ is a penalty parameter. We choose $\mathcal{R}_1(\mathbf{z}) = \|\mathbf{z}\|_1$.

A. The Penalty Approach and Its Majorization

As mentioned before, we penalize the nonlinear equation $g(D) = 0$ in (22) to the objective to obtain

$$\min_{D, \mathbf{z}} \; f_{\rho,\mu}(D, \mathbf{z}), \quad \text{s.t.} \quad D \in \mathcal{B}, \quad \mathbf{z} \in \Re^n. \quad (25)$$

Below, we construct a majorization function for $f_{\rho,\mu}(D, \mathbf{z})$. Define

$$\phi(\mathbf{z}) := \frac{1}{2}\|W \circ (\mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T)\|^2.$$

We also define a few quantities. Let $t_j := \|W_{\cdot j}\|$ (the Euclidean norm of the jth column of W), $t_{\max} := \max\{t_j\}$, $\mathbf{t} := (t_1, \ldots, t_n)^T$, and $s_j := \sqrt{t_j^2 + t_{\max}^2}$, $j = 1, \ldots, n$. Since $\phi(\mathbf{z})$ is quadratic, the Taylor expansion at $\mathbf{y}$ yields

$$\begin{aligned} & \phi(\mathbf{z}) \\ = \; & \phi(\mathbf{y}) + \langle \nabla\phi(\mathbf{y}), \; \mathbf{z} - \mathbf{y} \rangle + \frac{1}{2}\langle \mathbf{z} - \mathbf{y}, \; \nabla^2\phi(\mathbf{y})(\mathbf{z} - \mathbf{y}) \rangle \\ = \; & \phi(\mathbf{y}) + \langle \nabla\phi(\mathbf{y}), \; \mathbf{z} - \mathbf{y} \rangle \\ & + \langle \mathbf{z} - \mathbf{y}, \; (W \circ W)(\mathbf{z} - \mathbf{y}) \rangle + \|\mathbf{t} \circ (\mathbf{z} - \mathbf{y})\|^2 \\ \leq \; & \phi(\mathbf{y}) + \langle \nabla\phi(\mathbf{y}), \; \mathbf{z} - \mathbf{y} \rangle \\ & + t_{\max}^2\|\mathbf{z} - \mathbf{y}\|^2 + \|\mathbf{t} \circ (\mathbf{z} - \mathbf{y})\|^2 \\ = \; & \phi(\mathbf{y}) + \langle \nabla\phi(\mathbf{y}), \; \mathbf{z} - \mathbf{y} \rangle + \langle \mathbf{z} - \mathbf{y}, \; S(\mathbf{z} - \mathbf{y}) \rangle \\ =: \; & \phi^m(\mathbf{z}, \mathbf{y}), \end{aligned}$$

where $S := \mathrm{diag}(s_1^2, \ldots, s_n^2)$. The inequality above used [33, Thm. 5.5.3], which implies

$$\langle \mathbf{x}, \; (W \circ W)\mathbf{x} \rangle \leq t_{\max}^2 \|\mathbf{x}\|^2, \qquad \forall \; \mathbf{x} \in \Re^n.$$

We can verify the conditions in (24) that $\phi^m(\mathbf{z}, \mathbf{y})$ is a majorization function of $\phi(\mathbf{z})$. Thus, a majorization function (denoted as $f_{\rho,\mu}^m$) of $f_{\rho,\mu}(D, \mathbf{z})$ can be constructed as follows.

$$\begin{aligned} f_{\rho,\mu}(D, \mathbf{z}) &= \frac{1}{2}\|W \circ (D - \overline{\Delta})\|^2 + \phi(\mathbf{z}) + \rho g(D) + \mu\|\mathbf{z}\|_1 \\ &\quad + \langle W \circ (\mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T), \; W \circ (D - \overline{\Delta}) \rangle \\ &\leq \frac{1}{2}\|W \circ (D - \overline{\Delta})\|^2 + \phi^m(\mathbf{z}, \mathbf{y}) + \rho g_r^m(D, A) \\ &\quad + \langle W \circ (\mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T), \; W \circ (D - \overline{\Delta}) \rangle + \mu\|\mathbf{z}\|_1 \\ &=: f_{\rho,\mu}^m(D, \mathbf{z}, A, \mathbf{y}), \quad \forall \; D, A \in \mathcal{S}^n, \; \mathbf{z}, \mathbf{y} \in \Re^n. \end{aligned}$$

B. Algorithm: SSMDS

Our algorithm now minimizes the majorization function $f_{\rho,\mu}^m$ instead of $f_{\rho,\mu}$. Given D^k and $\mathbf{z}^k$ (k is the index of iteration), we update

$$\left\{ \begin{array}{ll} D^{k+1} & = \arg\min_{D \in \mathcal{B}} \; f_{\rho,\mu}^m(D, \mathbf{z}^k, D^k, \mathbf{z}^k) \\ \mathbf{z}^{k+1} & = \arg\min_{\mathbf{z} \in \Re^n} \; f_{\rho,\mu}^m(D^{k+1}, \mathbf{z}, D^k, \mathbf{z}^k). \end{array} \right. \quad (26)$$

We show that (26) has a close-form solution.

(i) Computing D^{k+1}. For simplicity, define

$$Z^k := \mathbf{1}(\mathbf{z}^k)^T + \mathbf{z}^k\mathbf{1}^T, \; D_+^k := \Pi_{\mathcal{K}_+^n(r)}(-D^k), \; \overline{Z}^k := \overline{\Delta} - Z^k.$$

With some simple linear algebra, we obtain

$$\begin{aligned} & D^{k+1} \\ = \; & \arg\min_{D \in \mathcal{B}} \frac{1}{2}\|W \circ (D - \overline{Z}^k)\|^2 + \frac{\rho}{2}\|D\|^2 + \rho\langle D_+^k, \; D \rangle \\ = \; & \arg\min_{D \in \mathcal{B}} \sum_{i,j} \left(\frac{1}{2} D_{ij}^2 - \Delta_{ij}^k D_{ij} \right) \\ = \; & \arg\min_{D \in \mathcal{B}} \frac{1}{2}\|D - \Delta^k\|^2 \\ = \; & \Pi_{\mathcal{B}}(\Delta^k), \end{aligned} \quad (27)$$

where the matrix Δ^k is defined by

$$\Delta_{ij}^k := \left(W_{ij}^2 \overline{Z}_{ij}^k - \rho(D_+^k)_{ij} \right) / (W_{ij}^2 + \rho), \; i, j = 1, \ldots, n \quad (28)$$

and

$$D_{ij}^{k+1} = \left(\Pi_{\mathcal{B}}(\Delta^k) \right)_{ij} := \min\left\{ \max\{\Delta_{ij}^k, L_{ij}\}, U_{ij} \right\}. \quad (29)$$

(ii) Computing $\mathbf{z}^{k+1}$. We show that $\mathbf{z}^{k+1}$ can be computed through the soft-thresholding operator (2). Define

$$R_{k+1} := W \circ W \circ (\overline{\Delta} - D^{k+1}),\ \ \mathbf{y}^k := R_{k+1}\mathbf{1} - \frac{1}{2}\nabla\phi(\mathbf{z}^k).$$

With some simple linear algebra, we have

$$\begin{aligned} & \mathbf{z}^{k+1} = \arg\min\ f^m_{\rho,\mu}(D^{k+1}, \mathbf{z}, D^k, \mathbf{z}^k) \\ = \ & \arg\min\ \langle \mathbf{z}-\mathbf{z}^k,\ S(\mathbf{z}-\mathbf{z}^k)\rangle + \langle \nabla\phi(\mathbf{z}^k),\ \mathbf{z}-\mathbf{z}^k\rangle \\ & -\langle R_{k+1},\ \mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T\rangle + \mu\|\mathbf{z}\|_1 \\ = \ & \arg\min\ \langle \mathbf{z}-\mathbf{z}^k,\ S(\mathbf{z}-\mathbf{z}^k)\rangle - 2\langle \mathbf{y}^k,\ \mathbf{z}-\mathbf{z}^k\rangle + \mu\|\mathbf{z}\|_1 \\ = \ & \arg\min \sum_{i,j} \left[\left(s_i z_i - \underbrace{(s_i z_i^k + y_i^k/s_i)}_{:=t_i^k}\right)^2 + \mu|z_i|\right]. \end{aligned}$$

Each element of $\mathbf{z}^{k+1}$ can be computed through the soft-thresholding operator in (2):

$$z_i^{k+1} = \mathcal{S}_{\mu/(2s_i^2)}(t_i^k/s_i), \quad i = 1, \ldots, n. \tag{30}$$

We summarize the algorithm below.

Algorithm 1 SSMDS

```
1: Input data: Dissimilarity matrix $\Delta$, weight matrix $W$, penalty parameter $\rho > 0$, sparsity parameter $\mu > 0$, lower-bound matrix $L$, upper-bound matrix $U$ and the initial $D^0$, $\mathbf{z}^0$. Set $k := 0$.
2: Update $D^{k+1}$: Compute $D^{k+1} = \Pi_{\mathcal{B}}(\Delta^k)$ by (28) and (29)
3: Update $\mathbf{z}^{k+1}$: Compute $\mathbf{z}^{k+1}$ through (30).
4: Convergence check: Set $k := k + 1$ and go to Step 2 until convergence.
```

The convergence analysis of SSMDS can be similarly patterned as for the algorithm FSMDS in the next section. We omit its detail to save space.

V. Full-Space Sparse MDS

Similar to the previous section, this section develops an efficient algorithm for the full-space sparse MDS (23) with complete convergence analysis. Define

$$\begin{aligned} F(D,Z) &:= \frac{1}{2}\|W \circ [(D+Z) - \overline{\Delta}]\|^2, \\ F_\mu(D,Z) &:= F(D,Z) + \mu\mathcal{R}_2(Z), \\ F_{\rho,\mu}(D,Z) &:= F_\mu(D,Z) + \rho g_r(D). \end{aligned}$$

We choose $\mathcal{R}_2(Z) = \|Z\|_1 - \|Z\|$. The penalized problem is

$$\min\ F_{\rho,\mu}(D,Z), \quad \text{s.t.} \quad D \in \mathcal{B},\ Z \in \mathcal{S}^n. \tag{31}$$

A natural majorization function, denoted as $F^m_{\rho,\mu}$, for $F_{\rho,\mu}(D,Z)$ at a given point (D^k, Z^k) is

$$\begin{aligned} F^m_{\rho,\mu}(D, Z, D^k, Z^k) &:= \frac{1}{2}\|W \circ [(D+Z) - \overline{\Delta}]\|^2 \\ + \rho g^m_r(D, D^k) + \mu\|Z\|_1 &- \mu\Big(\underbrace{\|Z^k\| + \langle T^k,\ Z - Z^k\rangle}_{=:\psi^m(Z,Z^k)}\Big), \end{aligned}$$

where T^k is a subgradient in $\partial\|Z^k\|$:

$$\partial\|Z^k\| = \begin{cases} \{Z^k/\|Z^k\|\} & \text{if } Z^k \neq 0 \\ \{T \in \mathcal{S}^n \mid \|T\| \le 1\} & \text{otherwise.} \end{cases}$$

$F^m_{\rho,\mu}$ is a majorization of $F_{\rho,\mu}$ because g^m_r in (12) is a majorization of g and $-\psi^m$ is a majorization of $-\|Z\|$ by the convexity of $\|Z\|$. The next iterate is thus computed as follows:

$$\begin{cases} D^{k+1} &= \arg\min_{D\in\mathcal{B}}\ F^m_{\rho,\mu}(D, Z^k, D^k, Z^k) \\ Z^{k+1} &= \arg\min_{Z\in\mathcal{S}^n}\ F^m_{\rho,\mu}(D^{k+1}, Z, D^k, Z^k). \end{cases} \tag{32}$$

A. Algorithm: FSMDS

For easy reference, we call the algorithm (32) FSMDS. We first calculate D^{k+1}. Let $\overline{Z}^k := \overline{\Delta} - Z^k$ and $D^k_+ := \Pi_{\mathcal{K}^n_+(r)}(-D^k)$. With simple linear algebra, we have

$$\begin{aligned} & D^{k+1} = \arg\min_{D\in\mathcal{B}}\ F^m_{\rho,\mu}(D, Z^k, D^k, Z^k) \\ = \ & \arg\min_{D\in\mathcal{B}} \frac{1}{2}\|W \circ (D - \overline{Z}^k)\|^2 + \frac{\rho}{2}\|D\|^2 + \rho\langle D^k_+,\ D\rangle, \end{aligned}$$

which is exactly what we have obtained in (27). Hence, D^{k+1} can be computed by (28) and (29).

We now obtain the formula for computing Z^{k+1}. Let $\overline{D}^{k+1} := \overline{\Delta} - D^{k+1}$. With some linear algebra, we have

$$\begin{aligned} & Z^{k+1} = \arg\min F^m_{\rho,\mu}(D^{k+1}, Z, D^k, Z^k) \\ = \ & \arg\min \frac{1}{2}\|W \circ (Z - \overline{D}^{k+1})\|^2 + \mu\Big(\|Z\|_1 - \langle T^k,\ Z\rangle\Big) \\ = \ & \arg\min \sum_{W_{ij}\neq 0} \Big\{\frac{1}{2}\Big(Z_{ij} - (\overline{D}^{k+1} + \mu T^k_{ij}/W^2_{ij})\Big)^2 \\ & \qquad + (\mu/W^2_{ij})|Z_{ij}|\Big\}. \end{aligned}$$

Note that when $W_{ij} = 0$, the corresponding optimal $Z^{k+1}_{ij} = 0$. Once again, each element of Z^{k+1} can be computed by the soft-thresholding operator (2).

$$Z^{k+1}_{ij} = \begin{cases} \mathcal{S}_{\mu/W^2_{ij}}(\widehat{T}^k_{ij}) & \text{if } W_{ij} \neq 0 \\ 0 & \text{if } W_{ij} = 0, \end{cases} \tag{33}$$

with

$$\widehat{T}^k_{ij} := \overline{D}^{k+1}_{ij} + \mu T^k_{ij}/W^2_{ij} \quad \text{when } W_{ij} \neq 0. \tag{34}$$

We summarize FSMDS below.

Algorithm 2 FSMDS

```
1: Input data: Dissimilarity matrix $\Delta$, weight matrix $W$, penalty parameter $\rho > 0$, sparsity parameter $\mu > 0$, lower-bound matrix $L$, upper-bound matrix $U$, and the initial $D^0$, $Z^0$. Set $k := 0$.
2: Update $D^{k+1}$. Compute $\overline{Z}^k = \overline{\Delta} - Z^k$, $D^k_+ = \Pi_{\mathcal{K}^n_+(r)}(-D^k)$, and $D^{k+1} = \Pi_{\mathcal{B}}(\Delta^k)$ by (28) and (29).
3: Update $Z^{k+1}$. Compute $Z^{k+1}$ through (33) and (34).
4: Convergence check: Set $k := k + 1$ and go to Step 2 until convergence.
```

B. Convergence Analysis

Since FSMDS is an alternating majorization-minimization method, it shares the basic property that all majorization methods enjoy. That is, the functional sequence $\{F_{\rho,\mu}(D^k, Z^k)\}$ is nonincreasing:

$$\begin{aligned} F_{\rho,\mu}(D^k, Z^k) &= F^m_{\rho,\mu}(D^k, Z^k, D^k, Z^k) \quad \text{(by (24))} \\ &\ge F^m_{\rho,\mu}(D^{k+1}, Z^k, D^k, Z^k) \quad \text{(by (32))} \\ &\ge F^m_{\rho,\mu}(D^{k+1}, Z^{k+1}, D^k, Z^k) \quad \text{(by (32))} \\ &\ge F^m_{\rho,\mu}(D^{k+1}, Z^{k+1}, D^{k+1}, Z^{k+1}) \quad \text{(by (24))} \\ &\ge F_{\rho,\mu}(D^{k+1}, Z^{k+1}). \quad \text{(by (24))} \end{aligned}$$

As a matter of fact, we can prove that $\{F_{\rho,\mu}(D^k, Z^k)\}$ is strictly decreasing unless $D^{k+1} = D^k$ and $Z^{k+1} = Z^k$ for some k. Moreover, any limit (D^*, Z^*) of the iterates sequence $\{D^k, Z^k\}$ is a stationary point of (31), which satisfies the following first-order optimality condition:

$$\left\{ \begin{array}{l} \langle \nabla_D F(D^*, Z^*) + \rho(D^* + \Pi_{\Pi_{\mathcal{K}^n_+(r)}}(-D^*)), D - D^* \rangle \ge 0, \\ \forall\ D \in \mathcal{B} \ \text{ and } \ \nabla_Z F(D^*, Z^*) + \mu(\Gamma^* - T^*) = 0, \end{array} \right. \tag{35}$$

for some $\Gamma^* \in \partial\|Z^*\|_1$ and $T^* \in \partial\|Z^*\|$. Furthermore, we shall prove that the limit (D^*, Z^*) also satisfies $g_r(D^*) \le \epsilon$ for a given $\epsilon > 0$ provided that ρ is above certain threshold ρ_ϵ (to be defined below). With the condition (35), this makes (D^*, Z^*) an ϵ-approximate stationary point of (23).

We summarize those properties in the following result, whose proof is in Appendix B.

Theorem 5.1: We assume that $\mathcal{B}$ is bounded and let $\{D^k, Z^k\}$ be the sequence generated by Alg. 2. Then the following hold.

(i) $\{D^k, Z^k\}$ is bounded.

(ii) We have

$$\begin{aligned} & F_{\rho,\mu}(D^k, Z^k) - F_{\rho,\mu}(D^{k+1}, Z^{k+1}) \\ \ge\ & \frac{\rho}{2}\|D^{k+1} - D^k\|^2 \\ +\ & \frac{1}{2}\langle W \circ (Z^{k+1} - Z^k),\ W \circ (Z^{k+1} - Z^k)\rangle. \end{aligned}$$

Hence $\|D^{k+1} - D^k\| \to 0$ and $\|Z^{k+1} - Z^k\| \to 0$.

(iii) Any limit of $\{D^k, Z^k\}$ is a stationary point of (31). Moreover, for a given $\epsilon > 0$, if $D^0 \in (-\mathcal{K}^n_+(r)) \cap \mathcal{B}$, $Z^0 = 0$, and

$$\rho \ge \rho_\epsilon := \|W \circ (D^0 - \overline{\Delta})\|^2/(2\epsilon),$$

then (D^*, Z^*) is an ϵ-approximate stationary point of the original (regularized) problem (23).

Remark 1. Thm. 5.1 not only guarantees that any limit must satisfy the optimality condition of the problem (31), it also provides a practical stopping criterion for Alg. 2: When both $\|D^{k+1} - D^k\|$ and $\|Z^{k+1} - Z^k\|$ are small enough or the decrease in the objective $F_{\rho,\mu}$ is stagnant, we may terminate. Moreover, the limit is also an ϵ-approximate stationary point of the original problem under reasonable conditions. Since the penalty is not exact, producing an approximate stationary point of the original problem is probably the best result that can be expected from the algorithm.

Remark 2. We note that both SSMDS and FSMDS can be cast into the general framework of block Successive Upper-bound Minimization (SUM) proposed in [42]. It is worth noting the two important assumptions used in SUM. One is that the approximation function in SUM needs to satisfy a tighter bound [42, Condition A3] involving the directional derivative of the objective function. The second assumption is that the objective function needs to be regular in order to establish the main convergence result of SUM [42, Thm. 2]. It is therefore an interesting question whether our objective function satisfies the regularity condition or not. If so, similar convergence results could be established under the coerciveness of the objective function (instead of the boundedness assumption of $\mathcal{B}$ used in this paper).

Now we turn our attention to the benefit of using ℓ_{1-2} regularization. The next result shows that we can control the sparsity in the generated iterates by setting the sparsity control parameter μ above certain computable threshold (μ_s below). This is particularly useful if we know priori the level of outliers in the data matrix. We are not aware whether the sparsity-driven method in [9] or [11], [12] (or any of its variants) has such a useful property. As seen in Appendix C, the proof of Theorem 5.2 makes use of the differentiability of $F(D, Z)$, which is a direct consequence of cMDS objective. In contrast, the objectives in [9], [11], [12] are not differentiable.

Theorem 5.2: Suppose the initial point $Z^0 = 0$. Let $\{D^k, Z^k\}$ be the sequence generated by Alg. 2. For a given positive integer s, there exists $\mu_s > 0$ such that for any $\mu \ge \mu_s$, the number of nonzeros in Z^k is not greater than $2s$, i.e.,

$$\|Z^k\|_0 \le 2s, \quad k = 1, 2, \ldots.$$

Moreover, μ_s can be estimated as

$$\mu_s = \frac{\sqrt{2} w_{\max} \sqrt{F(D^0, 0) + \rho g(D^0)}}{\sqrt{2s} - 1}$$

where $w_{\max} := \max_{i,j}\{W_{ij}\}$.

We note that both Thm. 5.1 and Thm. 5.2 are also valid when $\mathcal{R}_2(Z) = \|Z\|_1$. Therefore, Alg. 1 also enjoys the properties stated in the two theorems.

VI. Numerical Experiments

This part is organized as follows. In Sect. VI-A, we describe the implementation issues about SSMDS, FSMDS, and three benchmark methods. We then test two types of problems modelled by multiple/single source localization respectively in Sect. VI-B and Sect. VI-C. Sect. VI-D reports the numerical performance of all methods on a real test data of Motorola facility localization [16].

A. Benchmark methods and implementations

(a): Benchmark methods and computational complexity. We will compare SSMDS and FSMDS with three other methods RMDS [9], HQMMDS [11] and TMDS [39]. They are all the latest methods for detecting outliers and both RMDS and HQMMDS also employ ℓ_1-type sparsity-driven regularizations to induce sparsity. TMDS detects violations of triangle inequalities when Δ is viewed as a weighted graph and aims to correct those violations so that the modified Δ is close to being Euclidean.

In particular, SSMDS and FSMDS are similar to RMDS and HQMMDS in the sense that they all employ the least squares to the dissimilarity data with ℓ_1-type sparse regularizers. In fact, the least square part used in SSMDS and FSMDS is known to be the S-stress function and the corresponding part in RMDS and HQMMDS is the raw stress. For more comments on both stress functions, please refer to [2, Sect. 11.2] and [2, Sect. 3.2] respectively. However, we would like to emphasize two key differences between the two settings.

(i) The objective in SSMDS and FSMDS is in terms of Euclidean distance matrix, while the objective in RMDS and HQMMDS is in terms of embedding coordinates. Therefore, the former belongs to matrix optimization involving spectral properties whereas the latter belongs to the classical multivariate optimization.

(ii) Our EDM optimization is constrained. It can include lower and upper bound constraints in (21). In contrast, The coordinate minimization formulation in RMDS and HQMMDS is unconstrained because any constraints such as (21) would render their majorization subproblem to lose its closed-form solution at each iteration. And the closed-form solution is essential for both methods.

In terms of the computational complexity, as remarked in [9, Remark 5], RMDS has an overall complexity $O(n^3)$ due to computing all pairwise distances each iteration and the generalized inverse of a Laplacian matrix. It is noted that when the weight matrix W is the all one matrix (i.e., $W_{ij} = 1$ for all (i, j)), the inverse can be computed just once, which significantly reduces the overall computational burden per step. However, the current implementation of RMDS (available from the Matlab file exchange centre) can only handle the case $W_{ij} = 1$. HQMMDS shares similar computational complexity as RMDS due to matrix multiplications and computing a generalized inverse of a matrix each iteration. TMDS relies on the number of broken triangles tested. While testing all the triangles to identify the broken ones amounts to the complexity of $O(n^3)$, it is suggested in [39] that sampling twice as many triangles as the expected number of outliers is adequate. For our methods, a major computation is computing $D^k_+ = \Pi_{\mathcal{K}^n_+(r)}(-D^k)$, which requires $O(rn^2)$, see ([22, Eq.(15)-(16)]). The computational complexity for Δ^k in (28) is about $5n^2$; for D^k in (29) is about $2n^2$; and for Z^k in (33) is about $8n^2$. The overall complexity for FSMDS is about $O((r + 15)n^2)$ each iteration.

(b) Stopping criterion. For the benchmark methods, we used their default stopping criterion. For FSMDS, we used

$$\texttt{Fprog}_k := \frac{F_{\rho,\mu}(D^{k-1}) - F_{\rho,\mu}(D^k)}{1 + F_{\rho,\mu}(D^{k-1})} \le 10^{-4},$$

and

$$\texttt{Kprog}_k := \frac{2g_r(D^k)}{\|JD^kJ\|^2} \le 10^{-3}.$$

By the identity in [20, Prop. 3.3], we can obtain that

$$\begin{aligned} \frac{2g_r(D^k)}{\|JD^kJ\|^2} &= \frac{\|D^k + \Pi_{\mathcal{K}^n_+(r)}(-D^k)\|^2}{\|JD^kJ\|^2} \\ &= 1 - \frac{\sum_{i=1}^r [\lambda_i^2 - \max(\lambda_i, 0)^2)]}{\lambda_1^2 + \cdots + \lambda_n^2} \\ &\ge \frac{\lambda_{r+1}^2 + \cdots + \lambda_n^2}{\lambda_1^2 + \cdots + \lambda_n^2}, \end{aligned}$$

where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$ are the eigenvalues of $(-JD^kJ)$. This inequality means that the proportion of the eigenvalues we cut away at the final D^k is very small. In other words, D^k is very close to a true rank-r EDM when $\texttt{Kprog}_k$ is small. Since the sequence $\{F_{\rho,\mu}(D^k)\}$ is nonincreasing and bounded from below by 0, the condition on $\texttt{Fprog}_k$ is well defined. For SSMDS, $F_{\rho,\mu}(D^k)$ should be replaced by $f_{\rho,\mu}(D^k)$.

(c) Parameter selections. All of those methods have some important parameters to set before use. In particular, RMDS has one (sparsity control parameter λ). HQMMDS has two (sparsity control parameter λ_1 and the smoothness regularization parameter λ_2). TMDS has one (number of estimated outliers). SSMDS and FSMDS have two (ρ and μ). It has always been a challenging task on choosing the best values for the parameters involved. [9, Remark 4] offered several heuristic guidelines that are also useful here. For example, a grid search could be used when test data is available (see Fig. 4 for our experiment). Based on our extensive experiments, we found that the following heuristic criteria are very effective. To choose (ρ, μ) such that (i) the number of iterations are between 20 and 60 and (ii) the values of the error $\texttt{Kprog}_k$ remains steady at the level between 10^{-3} to 10^{-4}. Furthermore, also as suggested in [9], if an expected number s of outliers is known, we can choose (ρ, μ) so that the the number of nonzeros in Z is about $2s$.

The initial point is set at $D^0 = \overline{\Delta}$, $Z = 0$ (for FSMDS) and $\mathbf{z} = 0$ for SSMDS. The lower bound matrix $L = 0$ and the upper bound matrix $U_{ij} = (n \times \max\{\delta_{ij}\})^2$. That is, each distance is bounded above by the longest path in the weighted graph defined by Δ. The starting embedding points for RMDS and HQMMDS are obtained by cMDS. We note that HQMMDS represents a family of robust methods depending on which robust M-estimator to be used. In our test, we chose the Welsch estimator and the kernel size used is $a^2 = 10^{10}$ as suggested by one referee. The inputs for other parameters were set at their default values.

Our main conclusion is that SSMDS and FSMDS are very competitive and outperform all other 3 solvers in many test instances. In particular, they are able to handle the box constraints (21), which is an effective way to improve localization accuracy. However, the box constraints may create big challenges for other methods.

B. Multiple source localization

We test a problem of the "plus" (+) sign data that was first tested in [9]. It was generated as follows

Example 6.1: (Plus sign data) We sample $n = 25$ points with equal space from the "plus" $(+)$ symbol of size 12. That is, $\mathbf{x}_i = (i-1, 6)^T$, $i = 1, \ldots, 13$, $\mathbf{x}_i = (6, i-14)^T$, $i = 14, \ldots, 19$, and $\mathbf{x}_{i-1} = (6, i-14)^T$, $i = 21, \ldots, 26$. The outlier-free, yet noisy distance is generated by

$$\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| + \epsilon_{ij}, \quad i < j = 2, \ldots, n,$$

where ϵ_{ij} follows the normal distribution with 0 mean and the variance σ^2. The indexes (i, j) of s outliers were uniformly drawn and their values were independently uniformly drawn over $[0, 20]$. These values were then added to the corresponding δ_{ij}. Finally, we set the four end-points as anchors (fixed): $\mathbf{a}_1 = \mathbf{x}_1 = (0, 6)^T$, $\mathbf{a}_2 = \mathbf{x}_{13} = (12, 6)^T$, $\mathbf{a}_3 = \mathbf{x}_{14} = (6, 12)^T$, and $\mathbf{a}_4 = \mathbf{x}_{25} = (6, 0)^T$.

The original tested data in [9] is without the four anchors being fixed. We tested the original data and then used the Procrustes (`procustes.m` Matlab built-in function) to map the output points to the true locations. Although the output of 4 methods (except SSMDS) are different, their localizations after applying the Procrustes method are surprisingly accurate with the Root-Mean-Squared-Error (RMSE):

$$\text{RMSE} = \sqrt{\sum \|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|^2 / n}$$

at an order of 10^{-14}, where $\hat{\mathbf{x}}_i$ are the final localizations. Therefore, the original data would not be able to differentiate the methods. Therefore, we add the 4 anchors as the fixed points to increase the difficulty of localizing the true positions. For this case, we cannot use Procustes method to the whole set of points. Instead, we have to map the four output points, denoted as $\widetilde{\mathbf{x}}_i$, $i = 1, 13, 14, 25$ to their anchors $\mathbf{a}_i$, $i = 1, \ldots, 4$ to obtain the linear mapping $\mathcal{T}$. We then map the rest points by $\widehat{\mathbf{x}}_i = \mathcal{T}(\widetilde{\mathbf{x}}_i)$. Finally, RMSE is computed for those $\widehat{\mathbf{x}}_i$. We refer to [35] and [36, Sect. IV] for the ways to derive such mapping $\mathcal{T}$.

(a) General performance when n is small. The following instances of Example 6.1 were tested: $\sigma^2 \in \{0.1, 0.2\}$ and the number of outliers $s \in \{15, 30, 45, 60, 75\}$, corresponding to about 5%, 10%, 15%, 20% and 25% of the total number of distances deducting the 6 fixed distances due to the 4 anchors. For SSMDS and FSMDS, we set $\rho = 1$ and $\mu = 6$. For RMDS, we used $\lambda = 1$ and for HQMMDS we used $\lambda_1 = 1$ and $\lambda_2 = 35$ for its overall best performance. For TMDS, the correct value of the outliers was used. Fig. 1 plots the embedding ($\sigma^2 = 0.1$ and $s = 60$) by the three methods: FSMDS, RMDS, and HQMMDS. We omitted the other two methods because of their poor performance and also for better visualization (there would be too many points on one graph for 5 methods). For this case, we set the random number generator `rng('default')` so that the results can be reproduced.

It can be visibly observed from Fig. 1 that FSMDS produced the best matching to the true positions of the data, with the lowest RMSE. To better understand the estimated distances, we also plotted the Shepard graph for the three methods. It is interesting to see that the estimated distances by FSMDS and RMDS are scattered almost evenly around the true diagonal line, with FSMDS having a narrow spreading region. There are quite a few points by RMDS that are far away from the diagonal line. Those few large errors resulted in a few long links in Fig. 1 and other links are very close to their true locations. In contrast, the distances by HQMMDS stay quite close to the diagonal line, but many of them are below the line, suggesting that HQMMDS tends to under-estimate the true distances.

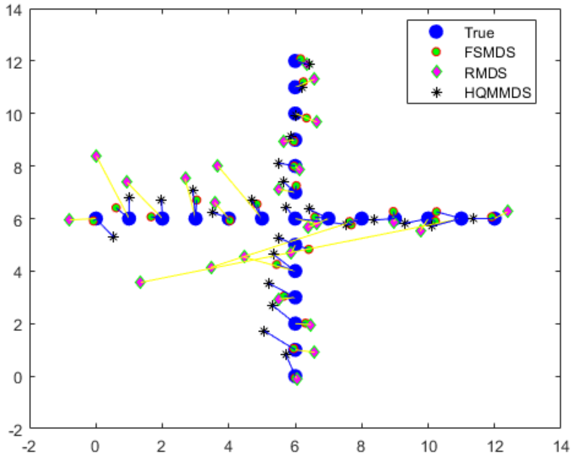

Fig. 1: Embedding for Example 6.1 ($\sigma^2 = 0.1$ and $s = 60$) by FSMDS, RMDS and HQMMDS, all linked to the corresponding true locations. The percentage of the outliers is about $60/(300-6) \approx 20\%$. The corresponding RMSE is 0.5496 for FSMDS, 2.6517 for RMDS, and 0.7245 for HQMMDS.

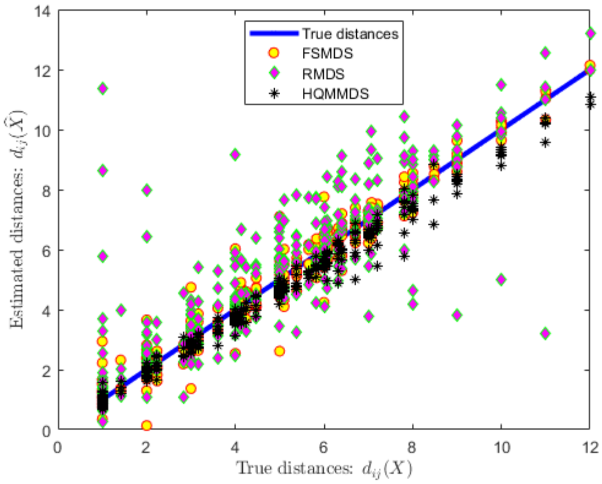

Fig. 2: Shepard graph for the embeddings in Fig. 1.

We further tested 15 instances of Example 6.1 and the corresponding average RMSE over 1000 simulations for each instance is reported in Table I. We observed that on average, FSMDS outperforms all other methods in all cases and HQMMDS works also very satisfactorily. It is worth pointing out that HQMMDS performs significantly better than RMDS despite they are closely related (see [11] for more details). The poor results by TMDS demonstrate that detecting all violated triangle inequalities in the data matrix Δ is not adequate to

TABLE I: RMSE for Example 6.1 by the five methods and RMSE is the average of 1000 simulations of each test instance where the random number generator in Matlab is set as `rng('shuffle')`. The numbers in brackets are the standard deviations. The parameters used were $\lambda = 1$ for RMDS, $(\lambda_1, \lambda_2) = (1, 35)$ for HQMMDS, s for TMDS and $(\rho, \mu) = (1, 6)$ for both SSMDS and FSMDS.

		Methods				
σ^2	s	SSMDS	FSMDS	RMDS	TMDS	HQMMDS
0	15	4.53 (1.41)	0.21 (0.38)	0.47 (1.00)	0.87 (1.01)	0.25 (0.32)
	30	5.76 (1.04)	0.39 (0.61)	0.99 (1.47)	2.25 (1.00)	0.55 (0.78)
	45	6.33 (0.94)	0.66 (0.87)	1.63 (1.76)	2.91 (1.05)	0.87 (0.99)
	60	6.62 (0.85)	1.00 (1.07)	2.09 (1.78)	3.44 (1.06)	1.20 (1.10)
	75	6.89 (0.78)	1.49 (1.25)	2.68 (1.78)	4.01 (1.05)	1.56 (1.14)
0.1	15	4.49 (1.39)	0.31 (0.30)	0.54 (0.84)	1.00 (1.04)	0.39 (0.30)
	30	5.72 (1.04)	0.45 (0.57)	0.85 (1.21)	2.31 (1.02)	0.57 (0.56)
	45	6.32 (0.90)	0.70 (0.83)	1.68 (1.66)	2.92 (1.06)	0.96 (0.95)
	60	6.65 (0.87)	1.08 (1.06)	2.09 (1.75)	3.47 (1.09)	1.29 (1.06)
	75	6.90 (0.80)	1.63 (1.30)	2.68 (1.76)	4.03 (1.04)	1.72 (1.19)
0.2	15	4.58 (1.39)	0.37 (0.26)	0.55 (0.79)	1.13 (1.13)	0.49 (0.34)
	30	5.76 (1.04)	0.51 (0.52)	1.04 (1.28)	2.26 (0.98)	0.68 (0.58)
	45	6.32 (0.91)	0.80 (0.85)	1.60 (1.58)	2.89 (1.02)	1.00 (0.85)
	60	6.63 (0.85)	1.12 (1.01)	2.18 (1.72)	3.55 (1.13)	1.35 (1.04)
	75	6.95 (0.80)	1.69 (1.26)	2.83 (1.72)	4.08 (1.07)	1.74 (1.12)

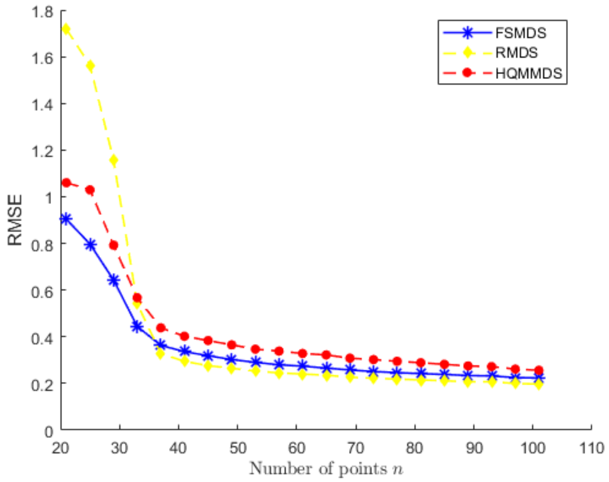

Fig. 3: RMSE performance of three method: HQMMDS, FSMDS and RMDS on Example 6.1 when the number of sampled points n gets big.

locate the true locations of the data points for most instances. We note that SSMDS completely fails for Example 6.1. This is expected because, as our theoretical result suggested, it is more suitable to single source localization problems.

(b) General performance when n gets bigger. It is interesting to see how those methods would perform when more sample points were drawn from the plus symbol. To this purpose, consider the size of the symbol of $2N$ with $N \geq 2$ being an integer and its center at (N, N). We sample $n = 4N + 1$ points with equal space on the symbol. Example 6.1 corresponds to $N = 6$. We again set the four corner points $(0, N)^T$, $(2N, N)^T$, $(N, 2N)^T$ and $(N, 0)^T$ as anchors. The variance of the normal noise added is $\sigma^2 = 0.2$ and we choose 15% of the total number of distances $n(n-1)/2$ being outliers. Their values were generated in the same way as in Example 6.1. We tested 500 instances and their average RMSE against the number of points n (i.e., N ranges from 5 to 25) were plotted in Fig. 3. We only included the three methods FSMDS, HQMMDS, and RMDS because the other two completely failed for most of the tested instances.

Two interesting observations can be made. One is that the improvement in terms of RMSE for all three methods gets better as n increases. This is reasonable and expected because there were more numbers of distances which were not outliers. The other is that the improvement becomes marginal after n is bigger enough (e.g., $n \geq 50$). The amount of improvement is significant when n is small (e.g., $n \leq 30$). In particular, RMDS improved the most over this range. Fig. 3 also suggests that one of the most challenging scenario in localization is when the network is small and is contaminated by a good number of outliers (say 15% of them).

(c) Sensitivity of FSMDS on (ρ, μ). Finally, we address another issue concerning the sensitivity of FSMDS on its two parameters ρ (penalty parameter) and μ (sparsity parameter). We tested FSMDS on a grid $[1, 40] \times [1, 40]$ for (ρ, μ) with unit step and plotted the corresponding RMSE in Fig. 4. It is interesting to see that RMSE in terms of (ρ, μ) behaves likes a step function, meaning that it performs similarly within a region and jumps to another region of similarities as the parameters vary. In other words, FSMDS is locally stable. The lowest RMSE took place when $(\rho, \mu) = (1, 6)$. We have also done this test for HQMMDS for its two parameters λ_1 and λ_2. Its lowest RMSE occurred at $(\lambda_1, \lambda_2) = (1, 35)$. We used those values in our extensive tests in Table I.

C. Single source localization

This is the hard test problem proposed in [17] with negative and positive measurement errors that lead to outliers.

Example 6.2: Suppose there are N (known) sensors that are uniformly placed on a circle with center $(0, 0)$ and radius 10:

$$\mathbf{x}_i = 10[\cos(2\pi(i-1)/N),\ \sin(2\pi(i-1)/N)]^T,\ i = 1, \ldots, N.$$

The unknown source $\mathbf{x}_n$ $(n = N + 1)$ is chosen uniformly at random from a disk centered at $(0, 0)$ with radius 15. The measurements from $\mathbf{x}_n$ to $\mathbf{x}_i$, $i = 1, \ldots, N$ are contaminated via $\delta_{in} = \|\mathbf{x}_i - \mathbf{x}_n\| + \epsilon_i + \eta_i$, where $\epsilon_i \sim N(0, \Sigma)$ with $\Sigma = 0.5\sigma^2(I_N + \mathbf{1}_N\mathbf{1}_N^T)$, and $\eta_i = U_i - U_0$ with U_i being

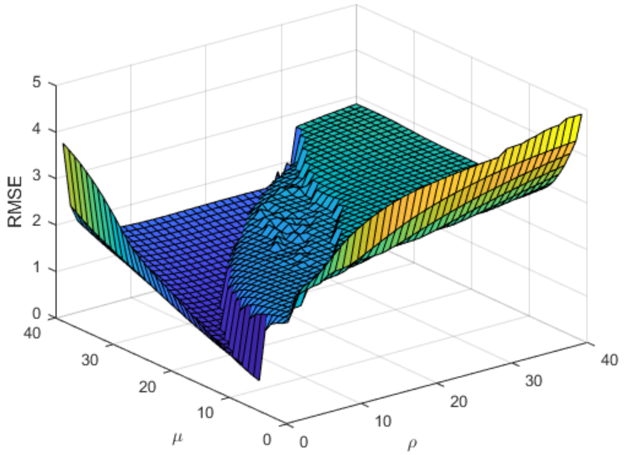

Fig. 4: RMSE of FSMDS on Example 6.1 with $\sigma^2 = 0.1$ and $s = 60$, `rng('default')`. The parameters (ρ, μ) vary on the grid of $[1, 40] \times [1, 40]$. The lowest RMSE is when $(\rho, \mu) = (1, 6)$.

TABLE II: Average error for Example 6.1 ($N = 4$) by the five methods over 1000 random simulations of each test instance. The numbers in brackets are the standard deviations. The parameters used were $\lambda = 1$ for RMDS, $(\lambda_1, \lambda_2) = (5, 0.001)$ for HQMMDS, $s = 4$ for TMDS and $(\rho, \mu) = (5, 1)$ for both SSMDS and FSMDS.

		Methods				
Cases	α	SSMDS	FSMDS	RMDS	TMDS	HQMMDS
Case 1	0.3	0.59	0.77	0.72	0.85	0.69
		(0.30)	(0.37)	(0.38)	(0.30)	(0.37)
	0.6	1.20	1.68	1.66	1.70	1.52
		(0.63)	(0.81)	(0.84)	(0.63)	(0.83)
	0.9	1.78	2.56	2.51	2.47	2.25
		(0.94)	(1.25)	(1.25)	(0.94)	(1.21)
Case 2	0.3	0.78	1.05	1.01	1.10	0.96
		(0.41)	(0.50)	(0.52)	(0.41)	(0.52)
	0.6	0.36	0.40	0.33	0.56	0.33
		(0.18)	(0.19)	(0.17)	(0.18)	(0.16)
	0.9	0.79	1.04	1.06	1.15	1.05
		(0.42)	(0.51)	(0.56)	(0.42)	(0.58)
Case 3	0.3	0.59	0.76	0.75	0.82	0.73
		(0.31)	(0.36)	(0.39)	(0.31)	(0.39)
	0.6	1.22	1.65	1.74	1.88	1.77
		(0.65)	(0.81)	(0.90)	(0.65)	(0.95)
	0.9	1.93	2.63	2.79	3.07	2.92
		(1.05)	(1.29)	(1.40)	(1.05)	(1.54)

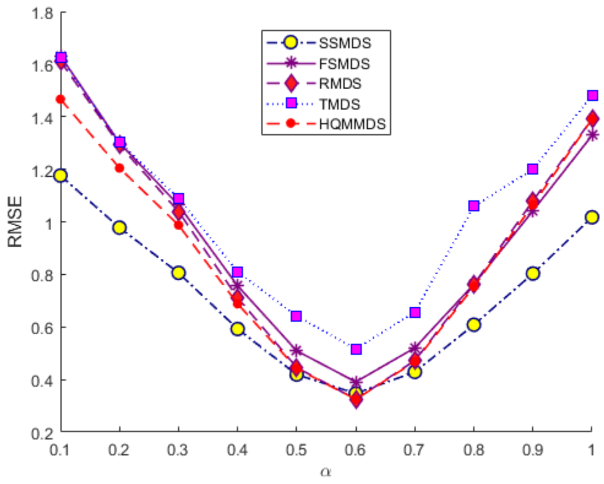

Fig. 5: Average error (RMSE) vs α varying from 0.1 to 1 for Case 2 in Example 6.2 over 1000 simulations of test data.

uniformly distributed between 0 and ω_i, $i = 0, 1, \ldots, N$. Here, ω_i can be treated as error upper bounds. We tested the first three scenarios in [17]. Case 1: $\omega_0 = 5\alpha$ and $\omega_i = 0.5$ for $i = 1, \ldots, N$. Case 2: $\omega_0 = 3$ and $\omega_i = 5\alpha$ for $i = 1, \ldots, N$. Case 3: $\omega_0 = 0.5\alpha$ and $\omega_i = 5\alpha$ for $i = 1, \ldots, N$. In all three cases, α varies from 0.1 to 1 and $\sigma = 0.3$.

This problem is designed to model distance measurements obtained by measuring the time of arrival of signals emitted from the sensors. Therefore, the large errors in η_i may be negative or positive, creating realistically diverse measurement errors. Another difficult feature of this problem is that the source has about 56% chance of lying outside of the convex hull of the known sensors. Table II reports the average localization error $\|\widehat{\mathbf{x}}_n - \mathbf{x}_n\|$ over 1000 simulations, where $\widehat{\mathbf{x}}_n$ is the estimated location and $\mathbf{x}_n$ is the true location. It can be seen that SSMDS yields the best performance in almost all cases except $\alpha = 0.6$ in Case 2, for which HQMMDS works better. We also plotted the results in Fig. 5 for Case 2 with α varying from 0.1 to 1. It is obvious that the line by SSMDS is the lowest except at $\alpha = 0.6$, where HQMMDS works slightly better. This verifies our theoretical result that SSMDS is particularly suitable to SSL problems. We also note that FSMDS, RMDS and TMDS all perform reasonably well.

D. Real data: Motorola facility localization

The real data was obtained by the channel measurement experiment conducted at the Motorola facility in Plantation, which is reported in [16]. The experiment environment is an office area which is partitioned by cubicle walls. 44 device locations are identified within a 14m × 13m area. Four of the devices labelled as $3, 11, 35, 44$ are chosen to be anchors and remaining locations are unknown. In this experiment, each node can communicate with all other nodes. We use the original time-of-arrival (TOA) to obtain the pairwise range measurements: $\delta_{ij} = c \times \mathrm{T\_TOA}_{ij}$, where c is the speed of light in terms of meters and $\mathrm{T\_TOA}_{ij}$ is the measured TOA between device i and j after removing the mean time delay error (details see [16]). This implies that all of the measurements have large errors (positive or negative). In particular, there are 37 negative pairwise distances in Δ (In our test, we replace them by $|\delta_{ij}|$). This data has been studied in [15], where a few latest state-of-art methods based on Semi-Definite Programming (SDP) were tested. The reported results there indicates that it would be challenging to achieve RMSE less than 1 meter for the unknown facilities.

We use this example to demonstrate two important strategies that are able to drive RMSE below 1m and that have not

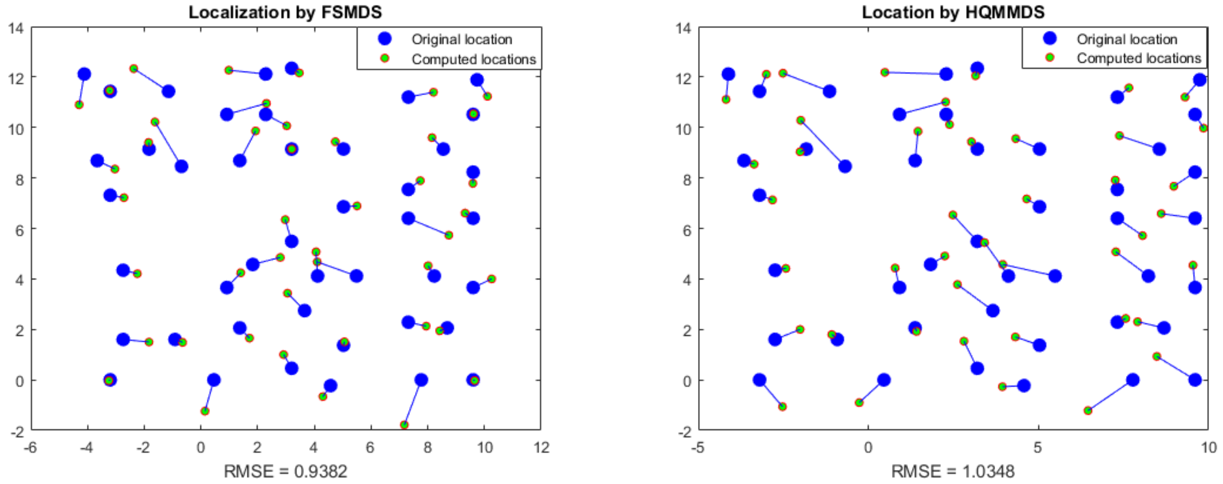

Fig. 6: Visualization difference between RMSE less than 1m (left graph by FSMDS) and RMSE above 1m (right graph HQMMDS): There appears bigger localization errors among the points near boundary for the right graph than the left.

been explored in the two previous examples. One is using the weights W_{ij} to distinguish importance of individual δ_{ij} to the objective. The other is enforcing tighter lower and upper bounds in (21).

(a) Sammon weighting scheme. It was proposed by Sammon [37], see also [2, P.255]. Each weight W_{ij} is inversely proportional to δ_{ij}. In our test, we used $W_{ij} = \alpha/\delta_{ij}$ with $\alpha = 3$ for $\delta_{ij} \neq 0$ and 0 otherwise. Here $\alpha > 0$ is balancing parameter, which actually can be factorized into the penalty and smoothing parameter ρ and μ. A generalized choice is $W_{ij} = \delta_{ij}^q$ with $q \in \Re$ being properly chosen and is proposed in [38]. We note that the standard choice $W_{ij} = 1$ when $\delta_{ij} \neq 0$ and 0 otherwise simply indicates that for the point pair (i, j) a dissimilarity δ_{ij} is available. The results are reported in Table III for both types of weights. It can be clearly seen that Sammon weights effectively drove RMSE below 1m for both SSMDS and FSMDS. All other methods are not affected by the different weighting choices. It is worth noting that RMDS and HQMMDS can also be adapted to include weights. But the implementations we obtained do not have such flexibility. The visualization of the obtained localization for the data by FSMDS and HQMMDS was plotted in Fig. 6. For TMDS, we used the half of the points for s.

TABLE III: Effect of Sammon weights on RMSE for Motorola data with $\alpha = 3$, and $\rho = 20$, $\mu = 90$ for SSMDS and FSMDS, $\lambda = 1$ for RMDS, $s = 20$ for TMDS, and $\lambda_1 = 20$, $\lambda_2 = 100$ for HQMMDS.

	Methods				
Weights	SSMDS	FSMDS	RMDS	TMDS	HQMMDS
Standard	1.24	1.17	1.09	1.22	1.04
Sammon	0.96	0.94	1.09	1.22	1.04

(b) Adding tighter lower and upper bounds. In the previous tests, we simply set the lower bound $L_{ij} = 0$ and the upper bounds U_{ij} big numbers. If we were able to increase the lower bounds and decrease the upper bounds toward their true values $d_{ij}^{\text{true}} = \|\mathbf{x}_i - \mathbf{x}_j\|$, then we expect that the resulting localization will become more accurate. For example, let

$$\ell_{ij} := \beta d_{ij}^{\text{true}} \quad \text{and} \quad u_{ij} := (2-\beta) d_{ij}^{\text{true}}.$$

As β varies from 0 to 1, the bounds in (21) with $L_{ij} = \ell_{ij}^2$ and $U_{ij} = u_{ij}^2$ become tighter. In the extreme case, $\beta = 1$, the bounds are true and should result in the true location. This is demonstrated in Fig. 7, where we considered three scenarios with FSMDS: (i) only increase the lower bounds (FSMDS-lb); (ii) only decrease the upper bounds (FSMDS-ub); and (iii) increase the lower bounds and decrease the upper bounds simultaneously.

We note that all three scenarios result in improvement in terms of RMSE accuracy and they all get better and better as the bounds get tighter. However, there were limits for both FSMDS-lb and FSMDS-ub. At the extreme $\beta = 1$ (the lower bounds or the upper bounds are true), the corresponding RMSE is between 0.4 and 0.5 and they cannot get smaller. In contrast, the best improvement occurred when the both bounds are enforced simultaneously. At the extreme, FSMDS-lu recover the true positions of the facilities. We also like to note that in practice, there would incur extra cost for obtaining tighter bounds. Fortunately there are many applications where such tighter bounds (known as interval distance geometry) are available, see a recent survey [41]. It is also important to note that while SSMDS and FSMDS have the capability of handling the lower and upper bounds without any extra cost, it is not known how other methods such as RMDS and HQMMDS can handle such constraints.

VII. Conclusion

cMDS has been a classical method for analyzing dissimilarity data and it is widely known that it spreads errors among all dissimilarities causing undesirable embeddings. This paper provides a new interpretation of cMDS and casts it as a joint optimization problem with one variable residing in the almost positive semidefinite cone $\mathcal{K}_+^n$ and the other in the subspace $\mathcal{S}_2^n$. This new reformulation also reveals why cMDS tends to overly denoise even there is just one erroneous dissimilarity

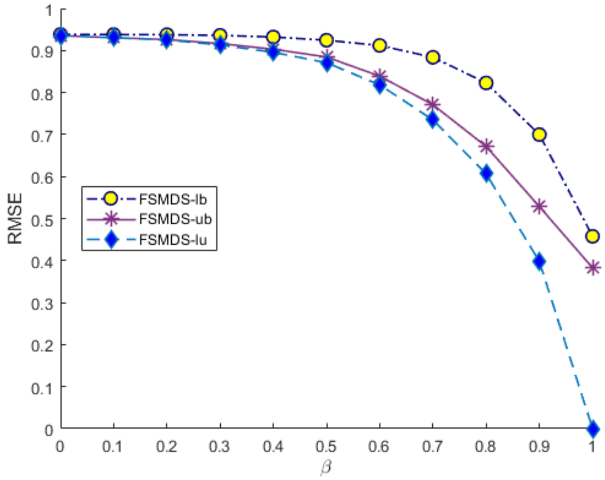

Fig. 7: Power of adding lower and upper bounds: as the bounds get tighter as β increases, FSMDS yields better localization. FSMDS-lb: adding lower bounds only; FSMDS-ub: adding upper bounds only; FSMDS-lu: adding both lower and upper bounds simultaneously.

and naturally leads us to consider a subspace MDS and its full-space variant FSMDS. We established their convergence results and compared them with several sate-of-the-art methods for outlier removal. Our numerical results on synthetic and real data demonstrate their capability of recovering high-quality embedding. In particular, we are able to handle the lower and upper bounds constraints, which could create huge challenging for other methods. For some applications such as the Motorola facility localization, enforcing quality lower and upper bounds is an effective (maybe the only way) to improve localization accuracy. This important capability of ours is due to the Euclidean distance matrix (EDM) optimization we employed.

In terms of the objectives, ours is based on the cMDS and both RMDS and HQMMDS are based on the stress function in MDS. One advantage of cMDS objective is its continuous differentiability when put in EDM optimization, which subsequently simplifies our proof analysis. It will be our next research topic to see if the proposed framework can be extended to the stress function.

Acknowledgement

We sincerely thank the associate editor and the four referees for their constructive comments, which have significantly improved the quality of the paper. In particular, we thank the referee who brought to our attention of [42], whose block successive upper-bound minimization framework is close to ours. The authors would like to thank Fotios Mandanas for sharing his code HQMMDS.

Appendix A
Proof of Theorem 3.1

We need the following result, which is a restatement of a result in [40].

Lemma A.1: [40, Cor. 2.1(a)] Let $\mathcal{K}^n_+$ be the conditionally positive semidefinite cone, $\mathcal{D}^n$ be the EDM cone and $\mathcal{S}^n_2$ be the subspace defined in (18). Then it holds

$$\mathcal{K}^n_- = \mathcal{D}^n + \mathcal{S}^n_2, \tag{36}$$

where $\mathcal{K}^n_- := -\mathcal{K}^n_+$. Moreover, the decomposition in (36) is unique in the sense that for any given matrix $A \in \mathcal{K}^n_-$, there exist unique $D \in \mathcal{D}^n$ and $Z \in \mathcal{S}^n_2$ such that $A = D + Z$ with

$$D = A - Z \quad \text{and} \quad Z := \mathbf{a}\mathbf{1}^T + \mathbf{1}\mathbf{a}^T, \quad \mathbf{a} := \frac{1}{2}\text{diag}(A).$$

Proof: (Thm. 3.1) The proof is in three parts: (i) We prove the optimization problem has a unique solution $(\widehat{D}, \widehat{Z})$; (ii) we prove $\widehat{D} = D^{\text{mds}}$; and (iii) we prove $\widehat{Z}$ takes the form given in the theorem.

(i) Lemma A.1 implies that the joint optimization problem (19) is equivalent to

$$\min \ \|\overline{\Delta} - A\|^2, \qquad \text{s.t.} \ \ A \in \mathcal{K}^n_-.$$

This is the projection problem onto the convex cone $\mathcal{K}^n_-$. Its unique optimal solution is $\widehat{A} := \Pi_{\mathcal{K}^n_-}(\overline{\Delta})$, and the corresponding unique $(\widehat{D}, \widehat{Z})$ are

$$\widehat{D} = \widehat{A} - \widehat{Z}, \quad \widehat{Z} = \widehat{\mathbf{a}}\mathbf{1}^T + \mathbf{1}\widehat{\mathbf{a}}^T, \quad \widehat{\mathbf{a}} := \frac{1}{2}\text{diag}(\widehat{A}).$$

(ii) It follows from (6) that

$$\widehat{A} = \Pi_{\mathcal{K}^n_-}(\overline{\Delta}) = -\Pi_{\mathcal{K}^n_+}(-\overline{\Delta}) = \widehat{\Delta} - \Pi_{\mathcal{S}^n_+}(J\widehat{\Delta}J). \tag{37}$$

We note (by direct verification) that

$$J = Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q, \ \text{with} \ \ Q := I - \frac{1}{n + \sqrt{n}}\mathbf{v}\mathbf{v}^T \tag{38}$$

and $\mathbf{v}^T := (1, \ldots, 1, \sqrt{n}+1) \in \Re^n$. Here, I_{n-1} is the identity matrix in $\mathcal{S}^{n-1}$ and Q is known as a Householder matrix satisfying $Q^2 = I$. Therefore,

$$\begin{aligned} J\overline{\Delta}J &= Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q\overline{\Delta}Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q \\ &= Q \begin{bmatrix} \overline{\Delta}_1 & 0 \\ 0 & 0 \end{bmatrix} Q, \end{aligned}$$

where $\overline{\Delta}_1$ is the leading $(n-1) \times (n-1)$ block of the matrix $Q\overline{\Delta}Q$. Since Q is orthogonal, we have

$$\Pi_{\mathcal{S}^n_+}(J\overline{\Delta}J) = Q \begin{bmatrix} \Pi_{\mathcal{S}^{n-1}_+}(\overline{\Delta}_1) & 0 \\ 0 & 0 \end{bmatrix} Q \tag{39}$$

and

$$Q\Pi_{\mathcal{S}^n_+}(J\overline{\Delta}J)Q = \begin{bmatrix} \Pi_{\mathcal{S}^{n-1}_+}(\overline{\Delta}_1) & 0 \\ 0 & 0 \end{bmatrix}. \tag{40}$$

Furthermore,

$$\begin{aligned} & J\Pi_{\mathcal{S}_+^n}(J\overline{\Delta}J)J \\ \overset{(38)}{=} \quad & Q\left[\begin{array}{cc} I_{n-1} & 0 \\ 0 & 0 \end{array}\right] Q\Pi_{\mathcal{S}_+^n}(J\overline{\Delta}J)Q\left[\begin{array}{cc} I_{n-1} & 0 \\ 0 & 0 \end{array}\right]Q \\ \overset{(40)}{=} \quad & Q\left[\begin{array}{cc} I_{n-1} & 0 \\ 0 & 0 \end{array}\right]\left[\begin{array}{cc} \Pi_{\mathcal{S}_+^{n-1}}(\overline{\Delta}_1) & 0 \\ 0 & 0 \end{array}\right]\left[\begin{array}{cc} I_{n-1} & 0 \\ 0 & 0 \end{array}\right]Q \\ = \quad & Q\left[\begin{array}{cc} \Pi_{\mathcal{S}_+^{n-1}}(\overline{\Delta}_1) & 0 \\ 0 & 0 \end{array}\right]Q \\ \overset{(39)}{=} \quad & \Pi_{\mathcal{S}_+^n}(J\overline{\Delta}J). \end{aligned} \tag{41}$$

Since $J\mathbf{1} = 0$, we have $J\widehat{Z}J = 0$. Putting those facts together, we have

$$\begin{aligned} J\widehat{D}J &= J(\widehat{A} - \widehat{Z})J = J\widehat{A}J \\ &\overset{(37)}{=} J\widehat{\Delta}J - J\Pi_{\mathcal{S}_+^n}(J\widehat{\Delta}J)J \\ &\overset{(41)}{=} J\widehat{\Delta}J - \Pi_{\mathcal{S}_+^n}(J\widehat{\Delta}J) \\ &= -\Pi_{\mathcal{S}_+^n}(-J\widehat{\Delta}J), \end{aligned}$$

where the last equation used the fact

$$X = \Pi_{\mathcal{S}_+^n}(X) - \Pi_{\mathcal{S}_+^n}(-X), \quad \forall\ X \in \mathcal{S}^n.$$

Consequently, we have

$$-\frac{1}{2}J\widehat{D}J = \frac{1}{2}\Pi_{\mathcal{S}_+^n}(-J\widehat{\Delta}J) \overset{(13)}{=} B_+.$$

Since $\widehat{D} \in \mathcal{D}^n$, it follows from (16) that $\widehat{D}$ can be generated by the decomposition of B_+ in (14): $\widehat{D}_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$. Hence, we must have $\widehat{D} = D^{\rm mds}$.

(iii) Since we have established $\widehat{D} = D^{\rm mds}$, the optimal $\widehat{Z}$ must be the optimal solution of the following problem:

$$\begin{aligned} \min_{Z \in \mathcal{S}_2^n} f(Z) &= \frac{1}{2}\|(D^{\rm mds} + Z) - \overline{\Delta}\|_F^2 \\ &= \frac{1}{2}\|(D^{\rm mds} + \mathbf{z}\mathbf{1}^T + \mathbf{1}\mathbf{z}^T) - \overline{\Delta}\|_F^2, \end{aligned}$$

with $\mathbf{z} \in \Re^n$. Recall $C := \overline{\Delta} - D^{\rm mds}$, we have

$$f(\mathbf{z}) = \frac{1}{2}\|C\|_F^2 - 2\langle C\mathbf{1},\ \mathbf{z}\rangle + n\|\mathbf{z}\|^2 + (\mathbf{1}^T\mathbf{z})^2.$$

Since $f(\mathbf{z})$ is convex, its gradient must vanish at its optimal solution $\hat{\mathbf{z}}$:

$$0 = \nabla f(\hat{\mathbf{z}}) = -2C\mathbf{1} + 2n\hat{\mathbf{z}} + 2(\mathbf{1}^T\hat{\mathbf{z}})\mathbf{1}.$$

Computing the inner product with $\mathbf{1}$ on both sides of the above equation yields

$$\mathbf{1}^T\hat{\mathbf{z}} = \frac{1}{2n}\mathbf{1}^TC\mathbf{1},$$

which in turn gives rise to

$$\hat{\mathbf{z}} = -\frac{1}{n}C\mathbf{1} - \frac{1}{2n^2}\mathbf{1}^TC\mathbf{1} = \mathbf{c} - \frac{1}{2}\bar{c}\mathbf{1}.$$

The optimal solution $\widehat{Z} = \hat{\mathbf{z}}\mathbf{1}^T + \mathbf{1}\hat{\mathbf{z}}^T$, which is what we stated in the theorem. ■

Appendix B
Proof of Theorem 5.1

Please refer to Sect. II and Sect. V for the definition of the functions $g(D)$, $h(D)$, $g_r^m(D, A)$ and $F(D, Z)$, $F_\mu(D, Z)$, $F_{\rho,\mu}(D, Z)$ and its majorization function $F_{\rho,\mu}^m(D, Z, D^k, Z^k)$. We further let $\varphi(Z) := \|Z\|_1 - \|Z\|$. We will need the following inequalities.

$$h(-D^{k+1}) - h(-D^k) \ge \langle \Pi_{\mathcal{K}_+^n(r)}(-D^k), D^k - D^{k+1}\rangle \tag{42}$$

due to the convexity of $h(\cdot)$ and $\Pi_{\mathcal{K}_+^n(r)}(-D^k) \in \partial h(-D^k)$ by (11).

Since $D^{k+1} = \arg\min F(D, Z^k) + \rho g_r^m(D, D^k)$, the optimality condition holds at D^{k+1}:

$$\langle \Omega_{k+1},\ D - D^{k+1}\rangle \ge 0, \quad \forall\ D \in \mathcal{B}, \tag{43}$$

where $\Omega_{k+1} := \nabla_D F(D^{k+1}, Z^k) + \rho(D^{k+1} + \Pi_{\mathcal{K}_+^n(r)}(-D^k))$. Since $Z^{k+1} = \arg\min F(D^{k+1}, Z) + \mu(\|Z\|_1 - \langle T^k, Z\rangle)$, the optimality condition holds at Z^{k+1}: There exists $\Gamma^{k+1} \in \partial\|Z^{k+1}\|_1$ such that

$$\nabla_Z F(D^{k+1}, Z^{k+1}) + \mu(\Gamma^{k+1} - T^k) = 0. \tag{44}$$

Define the quantity

$$\tau_k := \langle \nabla_Z F(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1}\rangle + \mu(\varphi(Z^k) - \varphi(Z^{k+1}).$$

We claim

Lemma B.1: $\tau_k \ge 0$.

Proof: It is known that for the one-dimensional absolute value function $|x|$, its subdifferential is defined as

$$\partial|x| = \left\{ \begin{array}{ll} 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ -1 & \text{if } x < 0. \end{array} \right.$$

One consequence is that $\xi x = |x|$ and and $|\xi| \le 1$ for any $\xi \in \partial|x|$. Applying this fact to $\Gamma^{k+1} \in \partial\|Z^{k+1}\|_1$ yields

$$\langle \Gamma^{k+1},\ Z^{k+1}\rangle = \|Z^{k+1}\|_1, \quad \langle \Gamma^{k+1}, Z\rangle \le \|Z\|_1, \forall\ Z. \tag{45}$$

Now computing the inner product with $(Z^k - Z^{k+1})$ on both sides of (44) leads to

$$\begin{aligned} & \langle \nabla_Z F(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1}\rangle \\ = \quad & \mu\langle \Gamma^{k+1} - T^k, Z^{k+1} - Z^k\rangle \\ = \quad & \mu\langle \Gamma^{k+1}, Z^{k+1}\rangle - \mu\langle \Gamma^{k+1}, Z^k\rangle - \mu\langle T^k, Z^{k+1} - Z^k\rangle \\ \overset{(45)}{=} \quad & \mu\|Z^{k+1}\|_1 - \mu\langle \Gamma^{k+1}, Z^k\rangle - \mu\langle T^k, Z^{k+1} - Z^k\rangle. \end{aligned}$$

Substituting the above into τ_k and simplifying to get

$$\begin{aligned} \tau_k \quad = \quad & \mu\Big(\underbrace{\|Z^k\|_1 - \langle Z^k, \Gamma^{k+1}\rangle}_{\ge 0 \text{ due to (45)}}\Big) \\ & + \mu\Big(\underbrace{\|Z^{k+1}\| - \|Z^k\| - \langle T^k, Z^{k+1} - Z^k\rangle}_{\ge 0 \text{ due to the convexity of } \|Z\|}\Big) \end{aligned}$$

This completes the proof. ■

The following two identities can be verified directly.

$$\begin{aligned} & \|D^{k+1}\|^2 - \|D^k\|^2 \\ = \;& 2\langle D^{k+1} - D^k, D^{k+1}\rangle - \|D^{k+1} - D^k\|^2. \qquad (46) \\ & \nabla_D F(D^{k+1}, Z^{k+1}) - \nabla_D F(D^{k+1}, Z^k) \\ = \;& (W \circ W) \circ (Z^{k+1} - Z^k). \qquad (47) \end{aligned}$$

Proof: (Thm. 5.1) (i) Since $\mathcal{B}$ is bounded, $\{D^k\}$ is so because $D^k \in \mathcal{B}$. Now suppose $\{Z^k\}$ is not bounded. There must exists a subsequence indexed by $\{k_i\}$ such that $|Z^{k_i}_{\ell j}| \to \infty$ for some fixed (ℓ, j). According to the update rule (33), we must have $W_{\ell j} > 0$ (otherwise $Z^k_{\ell,j} = 0$ for all k). The nonincreasing property of $\{F_{\rho,\mu}(D^k, Z^k)\}$ yields

$$\begin{aligned} F_{\rho,\mu}(D^0, Z^0) \;&\geq\; F_{\rho,\mu}(D^{k_i}, Z^{k_i}) \geq F(D^{k_i}, Z^{k_i}) \\ &\geq\; \frac{1}{2} W^2_{\ell j}\Big(\overline{\Delta}_{\ell j} - D^{k_i}_{\ell j} - Z^{k_i}_{\ell j}\Big)^2 \to \infty \end{aligned}$$

due to the boundedness of $\{D^k\}$. This contradiction establishes the boundedness of $\{Z^k\}$.

(ii) This part of the proof involves a considerable amount of calculation, but most of them are simple. The first fact we used (the second equality below) is the exact Taylor expansion of $F(D, Z)$ at (D^{k+1}, Z^{k+1}) since $F(D, Z)$ is quadratic.

$$\begin{aligned} & F_{\rho,\mu}(D^k, Z^k) - F_{\rho,\mu}(D^{k+1}, Z^{k+1}) \\ = \;& F(D^k, Z^k) - F(D^{k+1}, Z^{k+1}) \\ + \;& \rho(g_r(D^k) - g_r(D^{k+1}) + \mu(\varphi(Z^k) - \varphi(Z^{k+1})) \\ = \;& \langle \underbrace{\nabla_D F(D^{k+1}, Z^{k+1})}_{\text{apply (47)}}, D^k - D^{k+1}\rangle \\ + \;& \langle \nabla_Z F(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1}\rangle \\ + \;& \frac{1}{2}\underbrace{\langle W \circ (D^k - D^{k+1}),\; W \circ (D^k - D^{k+1})\rangle}_{\geq 0} \\ + \;& \frac{1}{2}\langle W \circ (Z^k - Z^{k+1}),\; W \circ (Z^k - Z^{k+1})\rangle \\ + \;& \langle W \circ (Z^k - Z^{k+1}),\; W \circ (D^k - D^{k+1})\rangle \\ + \;& \frac{\rho}{2}\Big(\underbrace{\|D^k\|^2 - \|D^{k+1}\|^2}_{\text{apply (46)}}\Big) + \rho\Big(\underbrace{h(-D^{k+1}) - h(-D^k)}_{\text{apply (42)}}\Big) \\ + \;& \mu(\varphi(Z^k) - \varphi(Z^{k+1})) \\ \geq \;& \underbrace{\langle \Omega_{k+1}, D^k - D^{k+1}\rangle}_{\geq 0 \text{ by } (43)} + \frac{\rho}{2}\|D^k - D^{k+1}\|^2 \\ + \;& \langle \nabla_Z f(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1}\rangle \\ + \;& \frac{1}{2}\langle W \circ (Z^k - Z^{k+1}),\; W \circ (Z^k - Z^{k+1})\rangle \\ + \;& \mu(\varphi(Z^k) - \varphi(Z^{k+1})) \\ \geq \;& \frac{\rho}{2}\|D^k - D^{k+1}\|^2 \;+\; \tau_k \\ & + \frac{1}{2}\langle W \circ (Z^k - Z^{k+1}),\; W \circ (Z^k - Z^{k+1})\rangle. \end{aligned}$$

Lemma B.1 ($\tau_k \geq 0$) establishes the first claim in (ii).

Since $\{F_{\rho,\mu}(D^k, Z^k)\}$ is bounded below by 0, we must have $\lim F_{\rho,\mu}(D^k, Z^k) - F_{\rho,\mu}(D^{k+1}, Z^{k+1}) \to 0$, which forces $(D^{k+1} - D^k) \to 0$ and $Z^{k+1}_{\ell j} - Z^k_{\ell j} \to 0$ when $W_{\ell j} > 0$. However, $Z^k_{\ell j} = 0$ for all k when $W_{\ell j} = 0$. Hence, we also have $(Z^k - Z^{k+1}) \to 0$.

(iii) Suppose (D^*, Z^*) is the limit of a subsequence $\{D^k, Z^k\}_{k \in K}$. It follows from (ii) that $D^{k+1} \to D^*$ and $Z^{k+1} \to Z^*$ for $k \in K$. Since the subgradient sequence $\{\Gamma^{k+1}\}_{k \in K}$ and $\{Z^{k+1}\}_{k \in K}$ are bounded, without loss of generality we may assume $\Gamma^{k+1} \to \Gamma^*$ and $T^k \to T^*$.

By the upper semicontinuity of the subdifferentials of convex functions, we have

$$\Gamma^* \in \partial\|Z^*\|_1 \quad \text{and} \quad T^* \in \partial\|Z^*\|.$$

Taking the limits on both sides of (44) for $k \in K$ to obtain

$$\nabla_Z F(D^*, Z^*) + \mu(\Gamma^* - T^*) = 0.$$

And taking the limits on both sides of (43) for $k \in K$ to obtain

$$\langle \nabla_D F(D^*, Z^*) + \rho(D^* + \Pi_{\mathcal{K}^n_+(r)}(-D^*)),\; D - D^*\rangle \geq 0$$

for all $D \in \mathcal{B}$. These two conditions are the optimality conditions in (35).

Now suppose $D^0 \in (-\mathcal{K}^n_+(r)) \cap \mathcal{B}$ and $Z^0 = 0$. We have the following chain of inequalities.

$$\begin{aligned} & \frac{1}{2}\|W \circ (D^0 - \Delta)\|^2 \\ & = \frac{1}{2}\|W \circ (D^0 + Z^0 - \Delta)\|^2 + \rho g_r(D^0) + \mu\mathcal{R}_2(Z^0) \\ & \geq F_\mu(D^1, Z^0) + \rho g^m_r(D^1, D^0) \\ & = F^m_{\rho,\mu}(D^1, Z^0, D^0, Z^0) \\ & \geq F^m_{\rho,\mu}(D^1, Z^1, D^0, Z^0) \\ & \geq F_{\rho,\mu}(D^1, Z^1) \geq \cdots \\ & \geq F_{\rho,\mu}(D^k, Z^k) \\ & = \frac{1}{2}\|W \circ (D^k + Z^k - \overline{\Delta})\|^2 + \rho g_r(D^k) + \mu\mathcal{R}_2(Z^k) \\ & \geq \rho g_r(D^k). \end{aligned}$$

The first equation in the chain used the fact $g_r(D^0) = 0$ and $Z^0 = 0$. The first inequality was due to the definition of $g^m_r(\cdot)$. The second equation was because of the definition of $F^m_{\rho,\mu}$. The second inequality was because of (32). The third inequality was due to the properties of the majorization function $F^m_{\rho,\mu}$. The rest was obtained by repeatedly using the above facts. Taking the limit on $k \in K$, we have

$$\rho g_r(D^*) \leq \frac{1}{2}\||W \circ (D^0 - \Delta)\|^2.$$

Therefore, we have

$$g_r(D^*) \leq \frac{\|W \circ (D^0 - \Delta)\|^2}{2\rho} \leq \frac{\|W \circ (D^0 - \Delta)\|^2}{2\rho_\epsilon} = \epsilon.$$

This, together with the established condition (35), has proved that (D^*, Z^*) is an ϵ-approximate stationary point of (23). ■

Appendix C
Proof of Theorem 5.2

Proof: The proof technique is taken from [30]. It follows from (44) that

$$\nabla_Z F(D^k, Z^k) + \mu(\Gamma^k - T^{k-1}) = 0,$$

where $\Gamma^k \in \partial \|Z^k\|_1$ and $T^{k-1} \in \partial \|Z^{k-1}\|$. Therefore, $\|T^{k-1}\| \le 1$ and $\|\Gamma^k\| \ge \sqrt{\|Z^k\|_0}$, which imply

$$\begin{aligned} \|\nabla_Z F(D^k, Z^k)\| &= \mu \|\Gamma^k - T^{k-1}\| \\ \ge \quad & \mu\Big(\|\Gamma^k\| - \|T^{k-1}\|\Big) \ge \mu\Big(\sqrt{\|Z^k\|_0} - 1\Big). \end{aligned}$$

On the other hand, using

$$\nabla_Z F(D^k, Z^k) = W \circ W \circ (D^k + Z^k - \overline{\Delta}),$$

we obtain

$$\|\nabla_Z F(D^k, Z^k)\| \le w_{\max} \|W \circ (D^k + Z^k - \overline{\Delta})\|,$$

where $w_{\max} := \max\{W_{ij}\}$. We further note that

$$\frac{1}{2}\|W \circ (D^k + Z^k - \overline{\Delta})\|^2 \le F_{\rho,\mu}(D^k, Z^k) \le F_{\rho,\mu}(D^0, 0)$$

Putting the two bounds on $\|W \circ (D^k + Z^k - \overline{\Delta})\|$ together yields

$$\sqrt{\|Z^k\|_0} - 1 \le \frac{w_{\max}\sqrt{2F_{\rho,\mu}(D^0, 0)}}{\mu},$$

which means that $\mu_s > 0$ can be selected as in the theorem. We note that $F_{\rho,\mu}(D^0, 0)$ does not depend on μ. ■

References

[1] T.F. Cox and M.A.A. Cox, *Multidimensional Scaling*, 2nd Ed, Chapman and Hall/CRC, 2001.

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

[3] I.J. Schoenberg, "Remarks to Maurice Frechet's article Sur la definition axiomatque d'une classe d'espaces vectoriels distancies applicbles vectoriellement sur l'espace de Hilbet", *Ann. Math.*, 36, pp. 724-732, 1935.

[4] G. Young and A.S. Householder, "Discussion of a set of points in terms of their mutual distances", *Psychometrika*, 3, pp. 19-22, 1938.

[5] J.C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis", *Biometrika*, 53, pp. 325–338, 1966.

[6] J.B. Tenenbaum, V. de Silva and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction", *Science*, 290, pp. 2319-2323, 2000.

[7] R. Sibson "Studies in the robustness of multidimensional scaling: perturbational analysis of classical scaling", *J. Royal Statistical Society*, B, 41(2), 217–219, 1979.

[8] L. Cayton and S. Dasgupta "Robust Euclidean embedding", *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA 2006, pp. 169–176.

[9] P.A. Forero and G.B. Giannakis, "Sparsity-exploiting robust multidimensional scaling", *IEEE Trans. Signal Process.*, 60(8), pp. 4118-4134, 2012.

[10] J.B. Kruskal. "Nonmetric multidimensional scaling: a numerical method", *Psychometrika*, 29, pp. 115-129, 1964.

[11] F.D. Mandanas and C.L. Kotropoulos, "Robust multidimensional scaling using a maximum correntropy criterion", *IEEE Trans. Signal Process.*, 65(4), pp. 919-932, 2017.

[12] F.D. Mandanas and C.L. Kotropoulos, "M-estimators for robust multidimensional scaling employing ℓ_{21}-norm regularization", *Pattern Recognition*, 73, 235-246, 2018.

[13] H. Chen, G. Wang, Z. Wang, H. C. So, and H. V. Poor, "Non-Line-of-Sight node localization based on semi-definite programming in wireless sensor networks", *IEEE Trans. Wireless Commun.*, 11, 108-116, 2012.

[14] R.M. Vaghefi, J. Schloemann, and R.M. Buehrer, "NLOS mitigation in TOA-based localization using semidefinite programming", Positioning Navigation and Communication (WPNC), 2013, pp. 1-6.

[15] C. Ding and H.-D. Qi, "Convex Euclidean distance embedding for collaborative position localization with NLOS mitigation", *Comput Optim Appl.*, 66, 187-218, 2017.

[16] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks", *IEEE Tran. Signal Processing*, 51, 21372148, 2003.

[17] G. Wang, A. M-C. So and Y. Li, "Robust convex approximation methods for TDOA-based localization under NLOS conditions", *IEEE Trans. Signal Process.*, 64(13), 3281-3296, 2016.

[18] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems", *IEEE Tran. Signal. Process.*, 56, 1770–1778, 2008.

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

[20] H.-D. Qi and X.M. Yuan, "Computing the Nearest Euclidean Distance Matrix with Low Embedding Dimensions", *Math. Prog.*, 147, pp. 351-389, 2014.

[21] I. Dokmanic, R. Parhizkar, J. Ranieri and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications", *IEEE Signal Process. Mag.*, 32(6), pp. 12-30, 2015.

[22] S. Zhou, N.H. Xiu and H.-D. Qi, "A fast matrix majorization-projection method for penalized stress minimization with box constraints", *IEEE Trans. Signal Process.*, 66, pp. 4331-4346, 2018.

[23] R. Tibshirani, "Regression shrinkage and selection via the Lasso", *J. Roy. Stat. Soc.* B, 58, 267-288, 1996.

[24] N. Gaffke and R. Mathar, "A cyclic projection algorithm via duality", *Metrika*, 36, pp. 29-54, 1989.

[25] W.S. Torgerson, "Multidimensional scaling: I. Theory and method", *Psychometrika*, 17(4), pp. 401-419, 1952.

[26] K.V. Mardia, "Some properties of classical mulitidimensional scaling", *Comm. Statist. A – Theory Methods*, A7:1233-1243, 1978.

[27] R. Mathar, "The best Euclidean fit to a given distance matrix in prescribed dimensions", *Linear Alge. Appli.* 67, 1-6, 1985.

[28] 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, pp. 589-600, 1990.

[29] H.-D. Qi, "A semismooth Newton method for the nearest Euclidean distance matrix problem", *SIAM J. Matrix Anal. Appl.*, 34, pp. 67-93, 2013.

[30] P. Yin, Y. Lou, Q. He, and J. Xin, "Minimization of ℓ_{1-2} for compressed sensing", *SIAM J. Sci. Comput.*, 37, A536-A563, 2015.

[31] J. de Leeuw and P. Mair, "Multidimensional scaling using majorization: Smacof in R", *J. Stat. Software*, 31, pp. 1-30, 2009.

[32] Y. Sun, P. Babu and D.P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning", *IEEE Trans. Signal Process.*, 65(3), pp. 794-816, 2017.

[33] R.A. Horn and C.R. Johnson, "Topics in Matrix Analysis", Vol. 2, Cambridge University Press, 1991.

[34] E.C. Chu, H. Zhou and K. Lange, "Distance majorization and its applications", *Math. Program.*, 146, 409-436, 2014.

[35] K.S. Arun, T.S. Huang, S.D. Blosten, "Least-squares fitting of two 3-D point sets", IEEE Trans. Pattern Anal. Machine Intell., 9 (1987), 698-700.

[36] R. Sanyal, M. Jaiswal and K.N. Chaudhury, "On a registration-based approach to sensor network localization", IEEE Trans. Signal Process., 65 (2017), 5357-5367.

[37] J.W. Sammon, "A non-linear mapping for data structure analysis", IEEE Trans. on Computers, 18 (1969), 401-409.

[38] A. Buja and D.F. Swayne, "Visualization methodology for multidimensional scaling", J. Classification, 19 (2002), 7-44.

[39] L. Blouvshtein and D. Cohen-Or, "Outlier detection for robust multidimensional scaling", IEEE Trans. Pattern Recog. Machine Intell. 2018.

[40] C.A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions", *Constr. Approx.*, 2, 11-22, 1986.

[41] D.S. Goncalves, A. Mucherino, C. Lavor, and L. Liberti, "Recent advances on the interval distance geometry problem", *J. Global Optim.*, 69 (2017), 525-545.

[42] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization", SIAM J. Optim., 23, 1126-1153, 2013.