**UNIVERSITY OF SOUTHAMPTON**

SCHOOL OF MATHEMATICS

# Classical Multidimensional Scaling: New Perspectives from EDM Optimization

by

**Chuanqi Qi**

Thesis for the degree of Doctor of Philosophy

October 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

SCHOOL OF MATHEMATICS

Mathematics

Doctor of Philosophy

CLASSICAL MULTIDIMENSIONAL SCALING: NEW PERSPECTIVES FROM EDM OPTIMIZATION

by Chuanqi 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. It originated from the classical Euclidean geometry and has found a large number of applications in various disciplines both in sciences and social sciences. Its popularity has significantly extended to machine learning community due to its modernized version ISOMAP. The purpose of this thesis is to study why cMDS works from an optimization point of view and in particular, how cMDS works hand in hand with modern optimization under the framework of Euclidean Distance Matrix (EDM) Optimization.

The EDM optimization from cMDS has three difficulties to solve. One is from the requirement of low-dimensional embedding, which often results in a low-rank constraint in EDM optimization. The second is from the Euclidean distance constraints, which often can be reformulated as a connic constraint. The third difficulty is caused by many of the constraints enforced on certain distances such as lower and upper bounds constraints. Modern matrix optimization can efficiently handle those difficulties arose from the standard MDS problems. However, our target is not any of the standard MDS problems. It is related to outlier detection among given noisy distances.

Despite its wide and successful use in various disciplines, the view on the capability of cMDS of denoising and outlier detection is largely negative. However, its reason has never been seriously investigated. 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 $\ell_1$ and $\ell_{1-2}$ regularization to induce sparsity. For the subspace sparse MDS, we developed a proximal alternating direction method of multipliers (ADMM) and established its convergence. Interestingly, although ADMM has all the promised (and nice) convergence properties, its numerical performance is not as good as we expected. It can be used to solve some less challenging problems.

Driven by the numerical weakness of the proximal ADMM, we then develop fast majorization algorithms for both the subspace model and the full space model. We establish their convergence to a stationary point, which is the best outcome in general expected of an optimization algorithm. The majorization has been based on a penalty method, which penalizes the difficult low rank constraint to the objective. Moreover, 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.

# Contents

# List of Figures

# List of Tables

# Declaration of Authorship

I, Chuanqi Qi , declare that the thesis entitled *Classical Multidimensional Scaling: New Perspectives from EDM Optimization* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as follows. Lingchen. Kong, Chuanqi Qi and Hou-Duo Qi, Classical multidimensional scaling: a subspace perspective, over-denoising and outlier detection, IEEE Transactions on Signal Processing, 67 (2009), 3842–3857. L. Kong and H-D. Qi devised the models and studied their theory. C. Qi devleoped the agorithms and test.

Signed:

Date: October 2020.

# Acknowledgements

I would like to take this opportunity to express my sincere appreciation to those who have contributed to this thesis and supported me in one way or the other during this amazing journey.

Firstly, I would like to express my sincere gratitude to my supervisor Professor Hou-Duo Qi for his continuous guidance and all the useful discussions and brainstorming sessions, especially during the difficult conceptual development stage. I also remain indebted for his patience, understanding, motivation and support during the times when I was really down and depressed due to personal issues.

I am also hugely appreciative to Professor Jörg Fliege for invaluable suggestions and encouraging my research.

Finally, Words cannot express how grateful I am to my Father, mother and my mother-in law for all of the sacrifices they have made on my behalf . I would also like to thank all of my friends who supported me mentally to strive towards my goal.

# Nomenclature

$\|X\|_F$: The Frobenius norm.

$\Re^{m\times n}$: the space of $m \times n$ real matrices.

$\mathcal{S}^n$ and $\mathcal{S}^n_+$ (covariance matrix)

$\langle A, B\rangle = \text{Tr}(AB)$, the trace of $AB$ for any $A, B \in \mathcal{S}^n$

$\|A\|$: the Frobenius norm of $A \in \mathcal{S}^n$. It is induced by the standard trace product.

$I_n$: the identity matrix in $\mathcal{S}^n$ (aften abbreviated as $I$ if the size $n$ is clear from the context).

$\mathcal{D}^n$: the cone of all EDMs in $\mathcal{S}^n$.

$\text{diag}(D)$ is the vector formed by the diagonals of $D \in \mathcal{S}^n$.

$\text{Diag}(\mathbf{x})$ is the diagonal matrix, whose diagonal vector is $\mathbf{x}$

$\mathbf{1}^\perp$: The subspace orthogonal to the vector of all ones $\mathbf{1}$.

$\Pi_{\mathcal{S}^n_+}(A)$ is the orthogonal projection of $A \in \mathcal{S}^n$ onto $\mathcal{S}^n_+$:

$\Pi_{\mathcal{S}^n_+}(A) = \arg\min \left\{ \|X - A\| \;\middle|\; X \in \mathcal{S}^n_+ \right\}$.

$\mathcal{K}^n_+$: Conditonally positive semidefinite cone.

$\mathcal{K}^n_+(r)$: the $r$-cut of the conditonally positive semidefinite cone.

$\mathcal{K}^n_- = -\mathcal{K}^n_+$, Conditonally negative semidefinite cone.

$A^T$, the transpose of matrix $A$.

$\mathcal{S}^n_h = \{A \in \mathcal{S}^n \mid \text{diag}(A) = 0\}$.

$\mathcal{S}^n_2 = \{A \in \mathcal{S}^n \mid A = \mathbf{1}\mathbf{y}^T + \mathbf{y}\mathbf{1}^T, \ \forall\ \mathbf{y} \in \Re^n\}$.

$\mathcal{K}$, a closed convex cone.

$\mathcal{K}^*$ the dual cone of $\mathcal{K}$.

For a vector $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\|\mathbf{x}\|_0$ denotes the number of nonzero elements in $\mathbf{x}$.

$\|\mathbf{x}\| = x_1 + \ldots + x_n$ (the $\ell_1$ norm).

$\|\mathbf{x}\| = x_1^2 + \ldots + x_n^2$ (the $\ell_2$ norm).

For a positive semidefinite operator $\mathcal{P}$ in an Euclidean space with inner product $\langle \cdot, \cdot \rangle$, the $\mathcal{P}$-norm of $\mathbf{x}$ is defined as $\|\mathbf{x}\|_{\mathcal{P}} = \langle \mathbf{x},\ \mathcal{P}\mathbf{x} \rangle$.

For two vector $\mathbf{x}$ and $\mathbf{y}$, $(\mathbf{x} \circ \mathbf{y}) = (x_1y_1, \ldots, x_ny_n)^T$.

# Chapter 1

# Introduction

The classical Multi-Dimensional Scaling (cMDS) has become a corner stone for many methods for analysing dissimilarity data. It originated from the classical Euclidean geometry deeply rooted in Schoenberg [81, 82] and Young and Household [99]. It was later popularized by Torgerson [91] and Gower [45]. They each have made unique contribution to the field of MDS, which has found a large number of applications. A few text books are devoted to MDS and its applications, see, e.g., Cox and Cox [26], Borg and Groenen [12], and Pękalaska and Duin [66].

We often work with dissimilarity data, which measures how dissimilar among objects. We use a small artificial example [12, Exercise 2.4] to explain it. A psychologist investigates the dissimilarity of the colors red, orange, green, and blue. In a small experiment, she asks a subject to rank the six pairs of colours on their dissimilarity (1 = most similar, 6 = most dissimilar). The resulting table of ranks is given below. We have the following observation: (i) The data is symmetric. For example, the dissimilarity from Red to Orange is 1, so is the dissimilarity from Orange to Red. (ii) Self-dissimilarity is 0. For example, the dissimilarity between Red and itself is 0. Therefore, we can view

| Item | R | O | G | B |
|---|---|---|---|---|
| Red | 0 | – | – | – |
| Orange | 1 | 0 | – | – |
| Green | 3 | 2 | 0 | – |
| Blue | 5 | 6 | 4 | 0 |

dissimilarity as one kind of distances between items. Such dissimilarity data are often analysed by the classical multidimensional scaling. [12, Chapter 4] gave three such examples with dissimilarity data from colour similarities, Morse codes confusions and facial expressions. `cMDS` is a method that represents those dissimilarity data in a Euclidean distance space. It can be symbolically put in this way:

$$\texttt{cMDS}(\Delta) \qquad \Longrightarrow \qquad \mathbf{x}_1, \quad \mathbf{x}_2, \quad \mathbf{x}_3, \quad \mathbf{x}_4$$

where $\Delta$ represents the given dissimilarity matrix and $\mathbf{x}_i$ are points in a Euclidean space. For example, the above colour data results in the following representation in $\Re^2$:

$$\mathbf{x}_1 = \begin{bmatrix} -1.3048 \\ 1.3225 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} -2.3180 \\ -0.2192 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} -0.0639 \\ -1.4090 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 3.6867 \\ 0.3058 \end{bmatrix}$$

Those four points are called embedding points. The pairwise Euclidean distance matrix among the 4 points are

$$\begin{bmatrix} 0 & -- & -- & -- \\ 1.8448 & 0 & -- & -- \\ 3.0001 & 2.5488 & 0 & -- \\ 5.0940 & 6.0276 & 4.1241 & 0 \end{bmatrix}$$

Comparing to the original data, this is a very good approximation. Once we have those embedding points, many statistical methods may be applied. However, this statistical analysis part is not a main topic in this thesis.

The popularity of `cMDS` has significantly extended to machine learning community due to its modernized version ISOMAP by Tenenbaum et.al. [89]. The essence of ISOMAP is that, as long as the underlying distances are approximately Euclidean, then cMDS appears to work brilliantly to bring out the inner structure of the data. For example, the geodesic distances used in ISOMAP on a manifold in a high dimensional space are close to be Euclidean on a low dimensional Euclidean space. The purpose of this thesis is to study why cMDS works from an optimization point of view and in particular, how cMDS works hand in hand with modern optimization under the framework of Euclidean Distance Matrix (EDM) Optimization.

In the following, we explain in very loose terms what cMDS is about, its potential applications, and the main contribution we made in this thesis.

## 1.1 cMDS: From distances to coordinates

It is interesting and even revealing to observe that a problem is very easy to solve, however, its inverse problem could be extremely difficult to answer. For example, it is easy to calculate $2^6 = 64$. If we are given the number 64 and ask what is the power form $a^b$ that gives 64. Well, we know there are a few answers. If we want to have a unique answer, we would have to fix the base $a$ first and this leads to logarithms with different bases. cMDS is just like this example but in a more complex framework of question formulation.

Given a set of points $\mathbf{x}_i \in \Re^p$, $i = 1, \ldots, n$, the original problem is to calculate the (squared) pairwise Euclidean distances:

$$D = \left( d_{ij}^2 \right)_{i,j=1}^n, \qquad d_{ij} := \|\mathbf{x}_i - \mathbf{x}_j\|, \ i, j, = 1, \ldots, n.$$

It takes about $O(pn^2)$ simple operations $(+, -, \times)$ to compute $D$. The inverse problem is to compute those coordinates $\mathbf{x}_i$ given the matrix $D$. This inverse problem has infinitely many answers.

**Example 1.1.** *(4 points) Suppose we are given four points*

$$\mathbf{x}_1 = (0, \ 0)^T, \quad \mathbf{x}_2 = (-4, \ 3)^T, \quad \mathbf{x}_3 = (4, \ -3)^T, \quad \mathbf{x}_4 = (0, \ 5)^T.$$

*The squared pairwise Euclidean distance matrix is*

$$D = \begin{pmatrix} 0 & \|\mathbf{x}_1 - \mathbf{x}_2\|^2 & \|\mathbf{x}_1 - \mathbf{x}_3\|^2 & \|\mathbf{x}_1 - \mathbf{x}_4\|^2 \\ & 0 & \|\mathbf{x}_2 - \mathbf{x}_3\|^2 & \|\mathbf{x}_2 - \mathbf{x}_4\|^2 \\ & & 0 & \|\mathbf{x}_3 - \mathbf{x}_4\|^2 \\ & & & 0 \end{pmatrix} = \begin{pmatrix} 0 & 5^2 & 5^2 & 5^2 \\ & 0 & 8^2 & 80 \\ & & 0 & 80 \\ & & & 0 \end{pmatrix}.$$

*The inverse problem asks what is a set of points that generate this $D$. We see that any set of points $\{\mathbf{x}_i + \mathbf{b}\}$ will serve the purpose. If we want to get a unique solution, we need to enforce some conditions.*

*If we use cMDS (by using the Matlab built-in function `Y = cmdscale(sqrt(D))`), it would generate the following set of points:*

$$\mathbf{y}_1 = (0.25, 0)^T, \quad \mathbf{y}_2 = (-2.75, -4)^T, \quad \mathbf{y}_3 = (-2.75, 4)^T \quad \mathbf{y}_4 = (5.25, 0)^T.$$

*Those points satisfy the centralization condition:*

$$\mathbf{y}_1 + \mathbf{y}_2 + \mathbf{y}_3 + \mathbf{y}_4 = 0.$$

*That is, they place their geometric centre at the origin.*

Another way to get a unique set of points is to fix three points, say $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ (they are often known as anchors), then the fourth point $\mathbf{x}_1$ is uniquely determined by "positioning by three circle method" as illustrated in the following figure:

The situation becomes a little bit complicated if the given distances are not accurate (i.e., there are measurement errors or even missing), what are the best solutions to the inverse problem? This is exactly what cMDS tries to answer in a well-formulated computational procedure. As before in our example, one could enforce different conditions to induce different "best" solutions. This leads to diverse MDS methods in literature, see Review in Chapter 2.

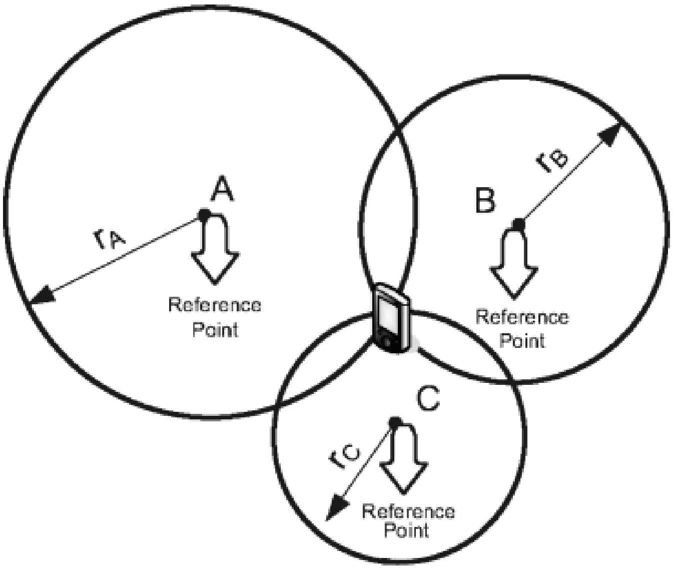


Figure 1.1: Positioning by thrilateration (source: Internet).

## 1.2 Applications

As mentioned above, in practice, distance measurements are often not accurate (noisy observations) and are even missing. To illustrate this common feature we present three such applications: Sensor Network Localization (SNL), Molecular Conformation and Image Processing.

### 1.2.1 Sensor network localization

In this application, we are typically given $m$ known sensors (they are called anchors): $\mathbf{x}_i = \mathbf{a}_i$, $i = 1, \ldots, m$; and a set of unknown sensors $\mathbf{x}_i$, $i = m+1, \ldots, n$ in $\Re^2$. Anchors and sensors can pick up the neighbouring sensors through receiving transmitted signals. The size of a neighbour is often determined by the radio range of the sensors. Distance information that we can collect from such situation can be represented as follows. We use $\delta_{ij}$ to denote the observed distance between sensor/anchor $i$ to a sensor $j$.

$$\begin{aligned}
\delta_{ij} &:= \|\mathbf{x}_i - \mathbf{x}_j\| \times |1 + \epsilon_{ij} \times \texttt{nf}|, \ \forall\ (i,j) \in \mathcal{N} \\
\mathcal{N} &:= \mathcal{N}_x \cup \mathcal{N}_a \\
\mathcal{N}_x &:= \{(i,j) \mid \|\mathbf{x}_i - \mathbf{x}_j\| \le R,\ i > j > m\} \\
\mathcal{N}_a &:= \{(i,j) \mid \|\mathbf{x}_i - \mathbf{a}_j\| \le R,\ i > m,\ 1 \le j \le m\},
\end{aligned}$$

where $R$ is known as the radio range, $\epsilon_{ij}$'s are independent standard normal random variables, and nf is the noise factor (e.g., nf $= 0.2$ corresponds to 20% noise level). In the literature (see, e. g., [8]), this type of perturbation in $\delta_{ij}$ is known to be multiplicative and follows the unit-ball rule in defining $\mathcal{N}_x$ and $\mathcal{N}_a$ (see [3, Sect. 3.1] for more detail).

Fig. 1.2 depicts such a situation. There are 4 anchors (big blue dot) and 50 sensors. When the distance between 2 sensors can be measured, there is a line (edge) to link them: they are neighbours The question is how to find those 50 sensors based on those partially observed distances (with noises) so that

$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx \delta_{ij}, \qquad \text{for those available } \delta_{ij}. \tag{1.1}$$

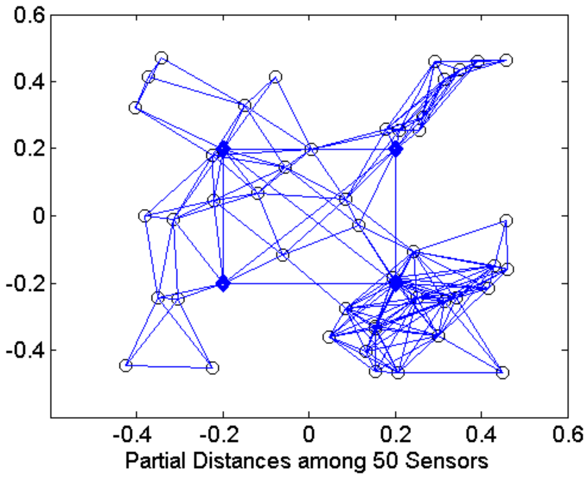


Figure 1.2: Sensor network localization (50 sensors with 4 anchors).

### 1.2.2 Molecular conformation

As emphasized in the abstract of [10]: "Magnetic resonance imaging (MRI) is a well known diagnostic tool in radiology that produces unsurpassed images of the human body, in particular of soft tissue. However, the medical community is often not aware that MRI is an important yet limited segment of magnetic resonance (MR) or nuclear magnetic resonance (NMR) as this method is called in basic science. The tremendous morphological information of MR images sometimes conceal the fact that MR signals in general contain much more information, especially on processes on the molecular level.

NMR is successfully used in physics, chemistry, and biology to explore and characterize chemical reactions, molecular conformations, biochemical pathways, solid state material, and many other applications that elucidate invisible characteristics of matter and tissue".

Molecular conformation is one of the applications of MR images and is any spatial arrangement of the atoms in a molecule which can be interconverted by rotations about formally single bonds. In this particular application, the information we can collect is approximate distances, again denoted as $\delta_{ij}$, between some atoms in a molecular. Hence, we have the same purpose as in SNL to find a set of points in $\Re^3$ that satisfy (1.1). Moreover, due to the physical properties of some of the well-studied atoms, we also have a large amount of constraints of the type:

$$\ell_{ij} \leq \|\mathbf{x}_i - \mathbf{x}_j\| \leq u_{ij},$$

where $\ell_1 ij$ and $u_{ij}$ are the lower and upper bounds that two atoms should obey. For more information of such constraints, see [42, 43].

Fig. 1.3 illustrates one of such applications to the molecular 1HOE in the protein data bank [6], based on a partially obtained distances information. The conformation is obtained by the method in [75].

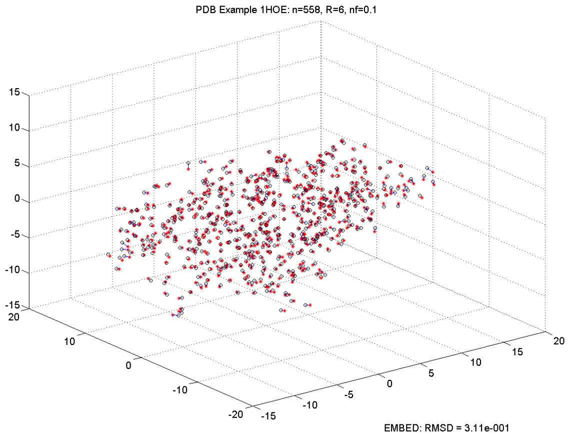
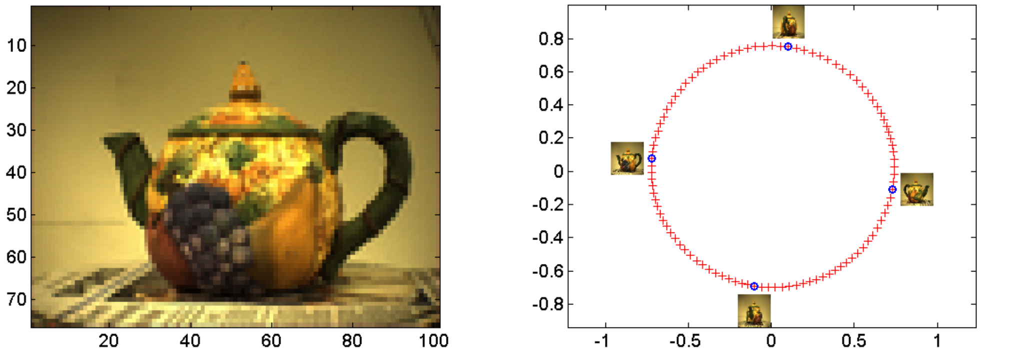


Figure 1.3: Molecular conformation: 1HOE from Protein Data Bank.

### 1.2.3 Image Processing

One of the major tasks in image processing is to represent images in a low-dimensional space so that the subsequent analysis can be quickly done. For example, suppose we took 100 photos of a teapot (see Fig. 1.4(a)). Each image has $76 \times 101$ pixes with 3 byte colour depth, hence is a vector of 23028 ($= 76 \times 101 \times 3$) dimensions. If we mixed up those photos, can we put them in order so that we can quickly know which photo was taken from what angle. If we can accomplish such a task effectively, the implication would be substantial in, say, reconstructing scenes of an event based photos taken from different sources. The information we need is being able to compute the neighbouring distances of any given photo (e.g., $K$-nearest neighbours). We then use such local distances to put the photos in order. Again, the approximation in (1.1) should be maximally preserved. The representation of this teapot example is illustrated in Fig. 1.4.



(a) Teapot image

(b) Representation of 100 teapot images on a circle

Figure 1.4: Representing teapot images in a low dimensional space.

If we are given a complete set of pairwise distances, denoted in $\Delta$, among the photos, we may apply `cMDS` to $\Delta$ to get a representation in an Euclidean space.

## 1.3 The Problem, Model and Main Contributions

`cMDS` has been well studied and widely used as will be demonstrated in the literature review (Chapter 2). It is notoriously known that it is incapable of detecting and removing outliers [28, 29]. One particular case is when we have many missing values in the dissimilarity matrix. The main purpose of this thesis is to understand why it is so and how we can improve its capability in this aspect.

### 1.3.1 The Problem and the Model

Suppose we are given a dissimilarity matrix $\Delta$, whose elements have the following characteristics:

$$\delta_{ij} = d_{ij} + z_{ij} + \epsilon_{ij}, \qquad 1 \le i < j = n, \tag{1.2}$$

where $d_{ij}$ are true Euclidean distances among unknown $n$ points, $z_{ij}$ are deterministic errors caused by physical faulty (e.g., battery of sensors weakening), and $\epsilon_{ij}$ are random errors. Our task is to find a set of embedding points $\mathbf{x}_i$, $i = 1, \ldots, n$ such that their pairwise distances are as close to their true distances $d_{ij}$ as possible. If all $z_{ij} = 0$, this case can be studied by `cMDS` provided all observations $\delta_{ij}$ are available [11].

Our innovation is to model the part $z_{ij}$. We first restrict $z_{ij}$ in a subspace

$$\mathcal{S}^n_2 := \left\{ Z \in \mathcal{S}^n \mid Z = \mathbf{1}\mathbf{y}^T + \mathbf{y}\mathbf{1}^T, \ \ \mathbf{y} \in \Re^n . \right\}$$

We then consider the case that $z_{ij}$ can take any values in the whole space $\mathcal{S}^n$ of $n \times n$ symmetric matrices. Those lead to the subspace model and the full-space model respectively:

$$\min_{D,\mathbf{y}} \ \|\Delta - (D + \mathbf{1}\mathbf{y}^T + \mathbf{y}\mathbf{1}^T))\|^2, \qquad \text{s.t.} \ \ D \text{ is Euclidean}, \ \ \mathbf{y} \in \mathcal{S}^n_2,$$

and

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

Those two models are thoroughly studied in Chapter 3 and Chapter 4. A hidden constraint is that $D$ needs to be of low-rank, which makes the two problems non-convex.

### 1.3.2 Main Contributions

This thesis makes a systematic study on this topic and has the following main contributions.

(i) The first contribution is summarized in Thm. 3.4 , which casts `cMDS` as a joint optimization of two variables: one is matrix variable $D$ being restricted to be Euclidean and a vector $\mathbf{y} \in \Re^n$ being restricted in a rank-2 subspace. This is a new characterization of `cMDS` and its consequence (Prop. 3.5) mathematically shows that why `cMDS` is incapable of outlier removal. It is because that it always tries to remove them by subtracting a dense matrix. In other words, in order to remove a small number of outliers, it punishes every distance no matter how accurate they are. This is bad and this explains its poor practical performance in outlier removal.

(ii) Based on Thm. 3.4, we develop an $\ell_1$ regularized joint optimization problem. It is a natural choice because of its extensive use in spare optimization. We enforce the sparsity on the vector variable $\mathbf{y}$ and this leads to a subspace MDS model. Since `cMDS` is nonconvex because of its low-dimensional embedding constraint, we study its convex relaxation by dropping the the embedding constraint. We then develop a proximal alternating direction method of multipliers (PADMM). This has been done in Sect. 3.4 and Sect. 3.5. We note that PADMM has been extensively studied in optimization over the past few years, we will not pursuit adaptation of its general convergence theory to our special case. Instead, we focus on how the proximal operators can be selected. We also demonstrate its numerical performance in the sensor network localization.

(iii) PADMM is mainly for structural convex optimization. However, our problem is essentially nonconvex. Our next contribution is on developing nonconvex methods. In Chapter 4, we further develop two more practical noconvex models. One is the subspace sparse MDS (SSMDS) and the full-space sparse MDS (FSMDS). Both models also employ the $\ell_1$ regularization. One advantage over existing methods is that both models can control the sparsity level provided the sparsity control parameter is above certain threshold, see Th,m. 4.3. In order to develop fast algorithms, we used some numerical techniques including penalty and majorization. Both techniques will be explained in Chapter 4. We also provide a complete set of convergence analysis for the developed algorithms (see, Thm.4.2).

(v) We have also done a good number of numerical experiments both on synthetic and real data. Our experience is that PADMM works well when the noise level is not too big. SSMDS works well when the outliers have some sparsity pattern while FSMDS works well when there is no sparsity pattern at all among the given data. Their performance has been compared with a few state-of-the-art algorithms in outlier removal. In particular, for the real data: Motorola facility localization (see Sect. 4.6), which contains a large number of outliers, FSMDS is able to achieve the best performance in terms of the root of mean-squared deviation among those tested solvers.

We conclude this part by emphasizing that outlier removal is a difficult problem, not just in the focused topic in this thesis, but also in many other disciplines. It is therefore unlikely that there exists a “universal” algorithm for all types of outlier removal. We believe our study provides an efficient approach for `cMDS` to detect as well as to remove those outliers.

## 1.4 Organization

In Chapter 2 we give a literature review, mainly focusing on those wildly used methods related to `cMDS`. In particular, we will discuss its link the principle component analysis when the Euclidean distances are computed from a given set of coordinates of points. We also discuss the two matrix approaches: Euclidean distance matrix optimization and semidefinite programming. Chapter 3 contains our main theoretical result on a new characterization of `cMDS`. We further study its convex relaxation and develop a PADMM with $\ell_1$ regularization. In Chapter 4, we study two nonconvex models (SSMDS and FSMDS) that also take into consideration of a few practical issues in embedding. We develop two fast algorithms for the two models and establish their convergence analysis.

Numerical comparison with several leading solvers are also reported in this chapter. We conclude the thesis in Chapter 5

# Chapter 2

# Literature Review

In this chapter, we conduct a review on some important theory and methods in MDS that are relevant to our research. We include formal or informal proofs for some of the facts that we think are important to our approach.

## 2.1 Multidimensional Scaling

Classical Multi-Dimensional Scaling (cMDS) originated in the 1930s when Young and Householder [99] showed how starting with a matrix of distances between points in an Euclidean space, coordinates for the points can be be found such that the pairwise distances are preserved. The same problem had also been investigated by Schoenberg [81] previously (see also [82] for a more general theory that casts this problem as a special case). However, the first major breakthrough that has led to a wide use of cMDS is due to Torgerson [91], followed by a celebrated (and independent) paper by Gower [45]. The equivalence of cMDS and PCA (Principal Component Analysis) in Section 2.2 is due to Gower.

In order to describe cMDS and its variants, we introduce the concept of Euclidean Distance Matrix (EDM). First, we let $\mathcal{S}^n$ denote the space of all $n \times n$ symmetric matrices.

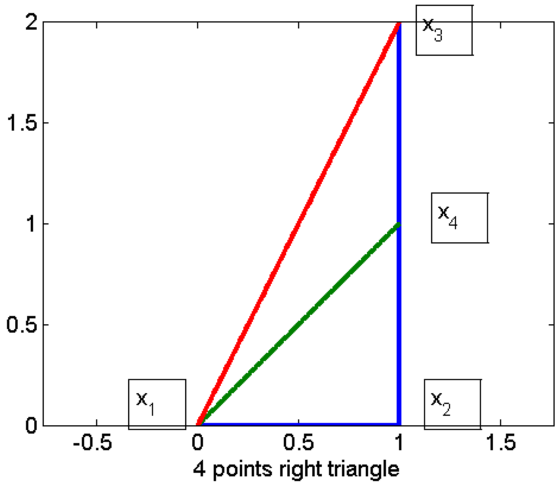


Figure 2.1: EDM for Triangle

**Definition 2.1.** A matrix $D \in \mathcal{S}^n$ is called EDM if there exists a set of $n$ points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\Re^r$ such that

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

The smallest dimension of such $r$ is called the embedding dimension of $D$. The set of points $\mathbf{x}_i$, $i = 1, \ldots, n$ are called the embedding points of $D$ in $\Re^r$.

Let us look at the simple example in Fig. 2.1, where 4 points formed a right triangle. The EDM matrix is then given by

$$D = \begin{bmatrix} 0 & 1 & 5 & 2 \\ 1 & 0 & 4 & 1 \\ 5 & 4 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix}.$$

We make following remarks.

(a) **Squared distances**. We note that the elements in $D$ are the **squared** pairwise (Euclidean) distance. For example, the Euclidean distance between point $\mathbf{x}_1$ and $\mathbf{x}_3$ is $\sqrt{5}$. The element $D_{13} = 5$, which is $(\sqrt{5})^2$.

(b) **Many sets of embedding points**. We note that one set of embedding points are

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Suppose $\mathbf{x}_0$ is another point in $\Re^2$, then $\{\mathbf{x}_i + \mathbf{x}_0\}$, $i = 1, \ldots, 4$ are another set of embedding points in $\Re^2$ because

$$D_{ij} = \|(\mathbf{x}_i + \mathbf{x}_0) - (\mathbf{x}_i + \mathbf{x}_0)\|^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad i, j = 1, \ldots, 4.$$

That is, we may have infinitely many sets of embedding points. There is a question as to which set should be used. This question is often resolved by the Procrustes analysis (see Sect. 2.5).

(c) **Emedding dimension**. Let $\mathbf{x}_i$ be the set of embedding points above. Define

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \in \Re^3, \quad i = 1, \ldots, 4.$$

Hence $\{\mathbf{z}_i\}$ is another set of embedding points in $\Re^3$ (a higher dimensional space). We note that $\Re^2$ is the smallest embedding space. Finding a lower dimensional embedding is often related to rank optimization.

The next important concept is the **centering matrix**, which aims to put the origin at the geometric centre of a set of given points. As commented above, for a given EDM $D \in \mathcal{S}^n$, there exist infinitely many embedding points. Suppose $\mathbf{x}_i$, $i = 1, \ldots, n$ are one set of such embedding points in $\Re^r$. We would like them to satisfy the **centralization** condition:

$$\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_n = 0. \tag{2.1}$$

If $\{\mathbf{x}_i\}$ does not satisfy the condition (2.1), what can we to make it satisfied? This can be done by multiplying them with the centering matrix.

**Definition 2.2.** Let $I$ be the identity matrix in $\mathcal{S}^n$ and $\mathbf{1}$ be the vector of all ones in $\Re^n$. The centering matrix $J$ is defined to be

$$J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T.$$

Now, let us assume that a set of point $\mathbf{y}_i$, $i = 1, \ldots, n$ are given in $\Re^r$. Then the set point $\{\mathbf{x}_i\}$ given by

$$[\mathbf{x}_1,\ \mathbf{x}_2,\ \cdots,\ \mathbf{x}_n] = [\mathbf{y}_1,\ \mathbf{y}_2,\ \cdots,\ \mathbf{y}_n]J,$$

must satisfy the condition (2.1). This is because $J\mathbf{1} = 0$, which is stated in the following result.

**Proposition 2.3.** *Let $J$ be the centering matrix defined in Definition 2.2. Then we must have*

$$J^2 = J.$$

*Moreover, the vector $\mathbf{1}$ is the eigenvector of $J$ with the corresponding eigenvalue being* 0. *That is*

$$J\mathbf{1} = 0.$$

**Proof.** Both results can be proved directly. We first note that

$$J\mathbf{1} = \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\mathbf{1} = \mathbf{1} - \frac{1}{n}\mathbf{1}(\mathbf{1}^T\mathbf{1}) = \mathbf{1} - \frac{1}{n} \times n \times \mathbf{1} = 0.$$

Hence, $\mathbf{1}$ is an eigenvector of $J$ with the corresponding eigenvalue 0. We proved $J\mathbf{1} = 0$.

We then note that

$$\begin{aligned} J^2 &= J \times J = J \times \left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right) \\ &= J - \frac{1}{n}(\underbrace{J\mathbf{1}}_{=0}\mathbf{1}^T) \\ &= J. \end{aligned}$$

(In linear algebra, $J$ is called a projection matrix). □

**Algorithm 1** cMDS($D$) [81, 99, 91, 45]

```
Input: A true Euclidean distance matrix $D \in \mathcal{S}^n$
Output: embedding points $\mathbf{x}_i \in \Re^r$, $i = 1, \ldots, n$.
Gram matrix: Compute the Gram matrix $B$ via the double centralization:
    $B := -\frac{1}{2}(JDJ)$
SVD: Compute the Singular Value Decomposition (SVD) of $B$ by
    $B = [\mathbf{p}_1,\ \mathbf{p}_2,\ \ldots,\ \mathbf{p}_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_n^T \end{bmatrix}$
    where $\lambda_1 \geq \ldots, \geq \lambda_n \geq 0$ with $\lambda_i$ being the eigenvalues of $B$ in nonincreasing order.
Embedding points:
    $X := [\mathbf{x}_1,\ \mathbf{x}_2, \ldots, \mathbf{x}_n] = [\sqrt{\lambda_1}\mathbf{p}_1,\ \sqrt{\lambda_2}\mathbf{p}_2,\ \ldots,\ \sqrt{\lambda_r}\mathbf{p}_r]^T, \qquad (2.2)$
    where $r$ is the number of positive eigenvalues of $B$.
```

We have the following remarks on the use of cMDS.

(i) When the input matrix $D$ is a true EDM, then the eigenvalues of the matrix $B$ must be all non-negative:

$$B \succeq 0. \tag{2.3}$$

Hence the algorithm is well defined in (2.2).

(ii) The resulting embedding points $\{\mathbf{x}_i\}$ must satisfy the centralization condition (2.1) as we see below. It follows from Prop. 2.3 that

$$B\mathbf{1} = -\frac{1}{2}JDJ\mathbf{1} = 0.$$

That is, $\mathbf{1}$ is an eigenvector of $B$ corresponding to the eigenvalue 0. Hence, the eigenvectors corresponding to its positive eigenvalues must be orthogonal to $\mathbf{1}$. In other words,

$$\langle \mathbf{p}_i,\ \mathbf{1} \rangle = 0, \qquad i = 1, \ldots, r.$$

Consequently,

$$\mathbf{x}_1 + \ldots + \mathbf{x}_n = X\mathbf{1} = \begin{bmatrix} \sqrt{\lambda_1}\langle \mathbf{p}_1,\ \mathbf{1}\rangle \\ \vdots \\ \sqrt{\lambda_r}\langle \mathbf{p}_r,\ \mathbf{1}\rangle \end{bmatrix} = 0.$$

The following result justifies the use of cMDS. It confirms that cMDS generates a set of embedding points that preserve the pairwise Euclidean distances given in an EDM $D$.

**Theorem 2.4.** *([26, Chapter 2]) Suppose $D$ is a true EDM and the embedding points $\mathbf{x}_i$ are generated by (2.2). Then we must have*

$$D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2, \qquad \textit{for all} \ \ i, j = 1, \ldots, n.$$

However, in practice, $D$ is not always a true EDM. It is often given by an another name: dissimilarity matrix $\Delta \in \mathcal{S}^n$, where each of its elements $\delta_{ij}$ represents measured dissimilarity between the point $i$ and point $j$. By convention, $\delta_{ij}$ is often a good approximation to the Euclidean distance between the point $i$ and point $j$:

$$\delta_{ij} \approx \|\mathbf{x}_i - \mathbf{x}_j\|.$$

cMDS can be adapted to this case and we describe it below in detail.

The only difference between Alg. 1 and Alg. 2 is that $\Delta^{(2)}$ is used in the place of the true EDM $D$. The consequence is that the resulting $B$ matrix is not Gram matrix any more. It may have negative eigenvalues. However, we only need its positive eigenvalues to generate a set of embedding points. If the negative eigenvalues are small, the embedding quality should be good. An example is illustrated in Fig. 2.2, whose only negative eigenvalue is is $2.0 \times 10^{-7}$. We see that the quality of the embedding is almost the same as in Fig. 2.1.

**Algorithm 2** cMDS($\Delta$) [26, Sect. 2.2.5]

```
Input: A dissimilarity matrix $\Delta \in \mathcal{S}^n$
Output: embedding points $\mathbf{x}_i \in \Re^r$, $i = 1, \dots, n$.
Squared dissimilarity matrix: Compute $\Delta^{(2)} := \Delta \circ \Delta = \left(\delta_{ij}^2\right)$.
Matrix $B$: Compute $B$ via the double centralization:
    $B := -\frac{1}{2}(J\Delta^{(2)}J)$
SVD: Compute the Singular Value Decomposition (SVD) of $B$ by
    $B = [\mathbf{p}_1,\ \mathbf{p}_2,\ \dots,\ \mathbf{p}_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_n^T \end{bmatrix}$
    where $\lambda_1 \geq \dots, \geq \lambda_n$ with $\lambda_i$ being the eigenvalues of $B$ in nonincreasing order.
Embedding points:
    $X := [\mathbf{x}_1,\ \mathbf{x}_2, \dots, \mathbf{x}_n] = [\sqrt{\lambda_1}\mathbf{p}_1,\ \sqrt{\lambda_2}\mathbf{p}_2,\ \dots,\ \sqrt{\lambda_r}\mathbf{p}_r]^T,$    (2.4)
    where $r$ is the number of positive eigenvalues of $B$.
```

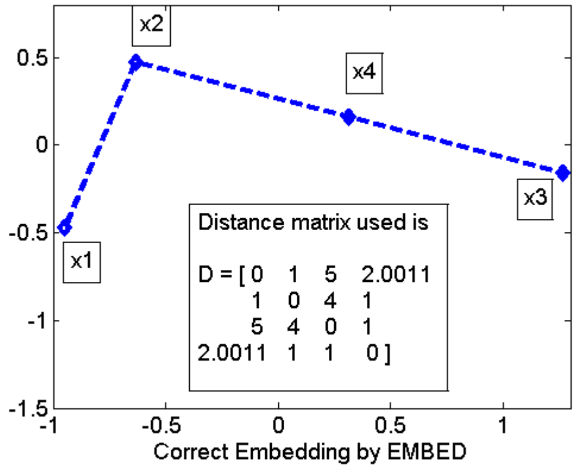


Figure 2.2: 4 points embedding.

## 2.2 cMDS vs PCA

This part aims to restate an important relationship between cMDS and the well-known principal component analysis (PCA) [51]. More detailed description can be found in [45] and [26, Chapter 2]. We first describe why we need PCA.

### 2.2.1 PCA

**(a) Basic idea of PCA.** Let us use a very simple example to show what the PCA aims to achieve. Suppose we have four points in two dimension:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}.$$

Apparently, those 4 points are on the line $y = x$ in $\Re^2$. The standard axes in $\Re^2$ are

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Now look at $\mathbf{x}_1$, it is easy to see that

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{x}_1^T \mathbf{e}_1 \\ \mathbf{x}_1^T \mathbf{e}_2 \end{bmatrix}.$$

We call the first component $\mathbf{x}_1^T\mathbf{e}_1$ is the first principal component of $\mathbf{x}_1$ along the first axis $\mathbf{e}_1$ent $\mathbf{x}_1^T\mathbf{e}_2$ is the second principal component of $\mathbf{x}_1$ along the second axis $\mathbf{e}_2$. The **main purpose of PCA** is to find a **new set of principal axes** that would give **more meaningful** interpretation of the points concerned.

For those 4 points, an obvious choice of a new axes are

$$\mathbf{v}_1 = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{v}_2 = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The first (principal) axis $\mathbf{v}_1$ now is the $y = x$ line and the second (principal) axis $\mathbf{v}_2$ is the orthogonal axis. For this new set of principal axes, the two principal components of $\mathbf{x}_1$ are

$$\mathbf{x}_1^T\mathbf{v}_1 = \sqrt{2} \qquad \text{and} \qquad \mathbf{x}_1^T\mathbf{v}_2 = 0.$$

This means that the second principal axis $\mathbf{v}_2$ does not bear any *useful* information for $\mathbf{x}_1$. And the component along $\mathbf{v}_1$ captures *all* information of $\mathbf{x}_1$. Those are also true for

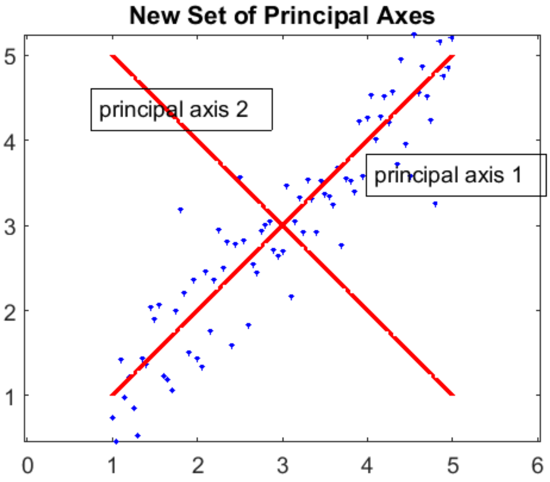


Figure 2.3: Demonstration of PCA

| Observation | Random Variables | | | | | Denoted by $\mathbf{x}$ |
|---|---|---|---|---|---|---|
| | $\xi_1$ | $\xi_2$ | $\cdots$ | $\cdots$ | $\xi_p$ | |
| $O_1$ | $x_{11}$ | $x_{12}$ | $\cdots$ | $\cdots$ | $x_{1p}$ | $= \mathbf{x}_1^T$ |
| $O_2$ | $x_{21}$ | $x_{22}$ | $\cdots$ | $\cdots$ | $x_{2p}$ | $= \mathbf{x}_2^T$ |
| $\vdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $\vdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $O_n$ | $x_{n1}$ | $x_{n2}$ | $\cdots$ | $\cdots$ | $x_{np}$ | $= \mathbf{x}_n^T$ |

Table 2.1: Representation of Data Points

other $\mathbf{x}_i$'s. In other words, the two dimensional data in $\mathbf{x}_i$ are actually represented by just one dimension in $\mathbf{v}_1$. The main idea was demonstrated in Figure 2.3 where many points along the $y = x$ line were generated (with random noises). The new principal axes are $\mathbf{v}_1$ and $\mathbf{v}_2$.

The major question is how to find those **principal axes** that would give meaningful interpretation of the known data, which is described below.

**(b) Data description.** Suppose we have $p$ random variables $\xi_i$, $i = 1, \ldots, p$. For each variable,we have $n$ observations. We include them in Table 2.1.

Let the observation matrix $X$ be

$$X := [\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_n]. \tag{2.5}$$

It has $n$ observations (columns) and each observation vector (column) has $p$ individual observations among the $p$ random variables $\xi_i$, $i = 1, \dots, p$. Without loss of generality, we can always assume that the data matrix has been mean-corrected. That is,

$$\overline{\mathbf{x}} := \frac{1}{n}\Big(\mathbf{x}_1 + \mathbf{x}_2 + \ldots + \mathbf{x}_n\Big) = 0. \tag{2.6}$$

Otherwise, we can subtract its means $\overline{\mathbf{x}}$ from each $\mathbf{x}_i$ so that the new data points satisfy (2.6).

The sample mean covariance matrix $S$ is given by

$$S := \frac{1}{n-1} XX^T \in \mathcal{S}^p. \tag{2.7}$$

We note that factor $(1/(n-1))$ above does not matter to us in our analysis. What matters to us is the Gram matrix $XX^T$, which is decomposed in SVD:

$$XX^T = V \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix} V^T \qquad \text{with} \quad V := [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p], \tag{2.8}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ are the eigenvalues of $XX^T$ and $V^T V = I_p$ (i.e., $\mathbf{v}_i$, $i = 1, \dots, p$ are the orthonormal eigenvectors). Those eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ are the **new principal axes** from the data matrix $X$.

Given a new observation vector $\mathbf{x}$, its principal components are given by

$$\begin{array}{cc} \text{1st principal component is} & \mathbf{x}^T\mathbf{v}_1 \\ \text{2nd principal component is} & \mathbf{x}^T\mathbf{v}_2 \\ \vdots & \vdots \\ \text{pth principal component is} & \mathbf{x}^T\mathbf{v}_p \end{array}$$

We may ask what the principal components are for the known observations $\mathbf{x}_i$. Let $\mathbf{s}_i$ denote the principal components of $\mathbf{x}_i$ (known as the score vector):

$$\mathbf{s}_i := [\mathbf{v}_1^T\mathbf{x}_i, \ \mathbf{v}_2^T\mathbf{x}_i, \ \cdots, \ \mathbf{v}_p^T\mathbf{x}_i]^T.$$

We collect all $\mathbf{s}_i$ in the following matrix. We call it PCA matrix, denoted by $S_{\mathtt{pca}}$, for easy reference:

$$S_{\mathtt{pca}} := \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \vdots \\ \mathbf{s}_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n \end{bmatrix} [\mathbf{v}_1,\ \mathbf{v}_2,\ \cdots,\ \mathbf{v}_p] = X^T V. \tag{2.9}$$

That is $X^T V$ contains all the principal components of the known data matrix $X$. Its first row contains the principal components of the first data point $\mathbf{x}_1$, the second row for the second data point $\mathbf{x}_2$, and so on. Moreover, we have the following important result.

**Proposition 2.5.** *(Zero scores) For any $\lambda_i = 0$, the scores of the data points $\mathbf{x}_i$ along the corresponding axis $\mathbf{v}_i$ must be zero.That is*

$$X^T\mathbf{v}_i = 0 \qquad \textit{whenever} \ \ \lambda_i = 0.$$

*Consequently, the PCA score matrix in (2.9) can be reduced to (by removing the zero columns from it)*

$$S_{pca} := \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \vdots \\ \mathbf{s}_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n \end{bmatrix} [\mathbf{v}_1,\ \mathbf{v}_2,\ \cdots,\ \mathbf{v}_r] = X^T[\mathbf{v}_1,\ \mathbf{v}_2,\ \cdots,\ \mathbf{v}_r], \tag{2.10}$$

*where $r = \mathrm{rank}(XX^T)$ (i.e., $\lambda_r > 0$ and $\lambda_{r+1} = 0$).*

**Proof.** Note that $\mathbf{v}_i$ is the eigenvector of $XX^T$ corresponding to the eigenvalue $\lambda_i$. Hence,

$$XX^T\mathbf{v}_i = \lambda_i\mathbf{v}_i.$$

Pre-multiplying $\mathbf{v}_i$ on both sides, we have

$$\|X^T\mathbf{v}_i\|^2 = \mathbf{v}_i^T XX^T\mathbf{v}_i = \lambda_i\mathbf{v}_i^T\mathbf{v}_i = \lambda_i\|\mathbf{v}_i\|^2.$$

For $\lambda_i = 0$, we must have $\|X^T\mathbf{v}_i\|^2 = 0$ and hence $X^T\mathbf{v}_i = 0$ whenever $\lambda_i = 0$. This completes our proof. □

The PCA described above is based on the data matrix $X$ given in (2.5). Its more general definition is for when a covariance matrix is given. We give a formal definition below for covariance matrices.

**Definition 2.6.** (PCA) Suppose $S \in \mathcal{S}^p$ is a given covariance matrix and it admits the SVD:

$$S = V \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix} V^T \qquad \text{with} \quad V := [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p],$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ are the eigenvalues of $S$ and $V^TV = I_p$ (i.e., $\mathbf{v}_i$, $i = 1, \ldots, p$ are the orthonormal eigenvectors). We have the following.

(i) Those eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ are the **principal axes** from the covariance matrix $S$.

(ii) For a given vector $\mathbf{x} \in \Re^p$. Its principal components are defined to be (contained in the vector $\mathbf{s}$)

$$\mathbf{s} := [\mathbf{v}_1^T\mathbf{x},\ \mathbf{v}_2^T\mathbf{x},\ \cdots,\ \mathbf{v}_p^T\mathbf{x}]^T.$$

### 2.2.2 PCoA

PCA works for covariance matrices. What can be said of when the matrix is EDM? It leads to what is now known as the **Principal Co-ordinate Analysis** (PCoA) due to Gower [45]. We now formally describe PCoA.

Let $D \in \mathcal{E}^n$ (the cone of EDMs of size $n$). $D$ can be thought of pairwise Euclidean distances among $n$ items (or objects). Define the corresponding $B$-matrix by

$$B := -\frac{1}{2}JDJ. \tag{2.11}$$

It is known from (2.3) that $B$ is positive semidefinite. Let $B$ admit the following SVD:

$$B = U \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_n \end{bmatrix} U^T \qquad \text{with} \quad U := [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n], \tag{2.12}$$

where $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n \geq 0$ are the eigenvalues of $B$ and $U^TU = I$ (i.e., $\mathbf{u}_i$, $i = 1, \ldots, n$ are the orthonormal eigenvectors).

**Definition 2.7.** (PCoA) Let $D \in \mathcal{E}^n$ and $B$ defined in (2.11). Assume that $B$ has the SVD (2.12) and $\mathrm{rank}(B) = r$ (i.e., $\mu_r > 0$ and $\mu_{r+1} = 0$). Then the PCoA of $D$ are given by

$$S_{\texttt{pcoa}} := \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \\ \vdots \\ \mathbf{t}_n \end{bmatrix} = \left[\sqrt{\mu_1}\mathbf{u}_1, \ \sqrt{\mu_2}\mathbf{u}_2, \ \cdots, \ \sqrt{\mu_r}\mathbf{u}_r\right]. \tag{2.13}$$

The vector $\mathbf{t}_1$ ($\mathbf{t}_1^T$ is the first row) contains the principal co-ordinates in $\Re^r$ of the first item, and $\mathbf{t}_2$ ($\mathbf{t}_2^T$ is the second row) contains the principal co-ordinates in $\Re^r$ of the second item; and so on.

It follows from Theorem 2.4 that

$$\|\mathbf{t}_i - \mathbf{t}_j\|^2 = D_{ij} \qquad \text{for all} \ \ i, j = 1, \ldots, n.$$

Therefore, the vectors $\mathbf{t}_i$'s are a set of embedding points, which depend on the SVD in (2.12). It is important to realize that $B$ may have many other decompositions. We state one important class of decompositions below.

**Proposition 2.8.** *(Decomposition of $B$ matrix) Suppose $D \in \mathcal{E}^n$ and the matrix $B$ is defined by (2.11). Suppose we have a set of points*

$$X := [\mathbf{x}_1, \ \mathbf{x}_2, \ \cdots, \ \mathbf{x}_n] \qquad \textit{with} \ \ \mathbf{x}_i \in \Re^N, \quad i = 1, \ldots, n,$$

*which satisfy the following two conditions*

$$\begin{array}{rll} \textit{Distance preservation condition:} & \|\mathbf{x}_i - \mathbf{x}_j\|^2 = D_{ij} & \textit{for all } \; i,j = 1,\dots,n \\ \textit{Centralization condition:} & \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_n = 0. & \end{array}$$

*Then we have*

$$B = X^T X.$$

**Proof.** Some of the identities used below are taken from [26, Chapter 2]. It follows from the distance preservation condition that

$$D_{ij} = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{x}_i^T \mathbf{x}_j. \tag{2.14}$$

We then have

$$\begin{aligned} \frac{1}{n}\sum_{j=1}^{n} D_{ij} &= \|\mathbf{x}_i\|^2 + \frac{1}{n}\sum_{j=1}^{n}\|\mathbf{x}_j\|^2 - 2\mathbf{x}_i^T \underbrace{\left(\sum_{j=1}^{n}\mathbf{x}_j\right)}_{=0} \\ &= \|\mathbf{x}_i\|^2 + \frac{1}{n}\sum_{j=1}^{n}\|\mathbf{x}_j\|^2 \qquad (2.15) \end{aligned}$$

We used the centralization condition above. Hence

$$\|\mathbf{x}_i\|^2 = \frac{1}{n}\sum_{j=1}^{n} D_{ij} - \frac{1}{n}\sum_{j=1}^{n}\|\mathbf{x}_j\|^2. \tag{2.16}$$

Similarly, we have

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} D_{ij} &= \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_j\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{x}_j^T \underbrace{\left(\sum_{i=1}^{n}\mathbf{x}_j\right)}_{=0} \\ &= \|\mathbf{x}_j\|^2 + \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2. \qquad (2.17) \end{aligned}$$

Hence,

$$\|\mathbf{x}_j\|^2 = \frac{1}{n}\sum_{i=1}^{n} D_{ij} - \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2. \tag{2.18}$$

Using (2.15) and (2.17), we have

$$\begin{aligned}
& \frac{1}{n}\left(\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{j=1}^{n} D_{ij}\right)\right) \\
= & \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij} \\
= & \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \frac{1}{n^2}\sum_{i=1}^{n}\left(\sum_{j=1}^{n}\|\mathbf{x}_j\|^2\right) \\
= & \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \frac{1}{n}\sum_{j=1}^{n}\|\mathbf{x}_j\|^2 \\
= & \frac{2}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2. \qquad (2.19)
\end{aligned}$$

Substituting (2.16) and (2.18) into (2.14) leads to

$$\begin{aligned}
\mathbf{x}_i^T\mathbf{x}_j &= -\frac{1}{2}\Big(D_{ij} - \|\mathbf{x}_i\|^2 - \|\mathbf{x}_j\|^2\Big) \\
&= -\frac{1}{2}\left(D_{ij} - \frac{1}{n}\sum_{j=1}^{n} D_{ij} - \frac{1}{n}\sum_{i=1}^{n} D_{ij} + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}\right) \\
&= -\frac{1}{2}\Big(JBJ\Big)_{ij} = B_{ij}.
\end{aligned}$$

We omit the detailed verification of the last equation above, which is due to elementary linear algebra. Hence the decomposition $B = X^T X$ is correct. □

Prop. 2.8 leads to the well-known characterization of an EDM $D \in \mathcal{E}^n$, which we have mentioned in (2.3).

**Corollary 2.9.** *(Positive semidefiniteness of $B$ matrix) Suppose $D \in \mathcal{E}^n$ and let $B$ be defined by (2.11). Then we must have*

$$B \in \mathcal{S}^n_+.$$

**Proof.** Since $D \in \mathcal{E}^n$, there exist a set of points $\mathbf{x}_i$ that satisfy the two conditions (distance preservation and centralization) in Prop. 2.8 such that

$$B = XX^T.$$

Hence, $B$ must be positive semidefinite. □

We note that an EDM $D$ must have zero diagonals. A matrix of such is called a *hollow* matrix.

**Definition 2.10.** (Subspace of hallow matrices) The hollow subspace, denoted by $\mathcal{S}^n_h$, in $\mathcal{S}^n$ is defined to be

$$\mathcal{S}^n_h := \left\{ A \in \mathcal{S}^n \;\middle|\; \mathrm{diag}(A) = 0 \right\}.$$

Corollary 2.9 leads to the following characterization of $\mathcal{E}^n$.

**Proposition 2.11.** *(PSD characterization of EDM via the centralization matrix $J$) A matrix $D \in \mathcal{S}^n$ is an EDM in $\mathcal{E}^n$ if and only if*

$$D \in \mathcal{S}^n_h \qquad \textit{and} \qquad B \in \mathcal{S}^n_+,$$

*where $B = -\frac{1}{2}JDJ$. Moreover, we have*

$$D = \mathcal{D}(B),$$

*where*

$$\mathcal{D}(B) := \mathrm{diag}(B)\mathbf{1}^T + \mathbf{1}\mathrm{diag}(B)^T - 2B. \tag{2.20}$$

**Proof.** The "Only if" part has been proved in Corollary 2.9. For the "If part", let $D \in \mathcal{S}^n_h$ and $B = -\frac{1}{2}JDJ \in \mathcal{S}^n_+$. Then $B$ must admit a decomposition of the form:

$$B = X^T X \qquad \text{with} \;\; X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \;\; \text{and} \;\; \mathbf{x}_i \in \Re^n.$$

Moreover, we have

$$X^T X\mathbf{1} = -\frac{1}{2}JDJ\mathbf{1} = 0 \qquad \text{(by Prop. 2.3)} \tag{2.21}$$

which implies

$$X\mathbf{1} = 0 \qquad \text{(i.e.,} \;\; \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_n = 0).$$

This is the centralization condition in Prop. 2.8. The proof there also applies here.

We give the details below. It follows from the factorization

$$2XX^T = -JDJ = -D + \frac{1}{n}\Big(\mathbf{1}\mathbf{1}^T D + D\mathbf{1}\mathbf{1}^T\Big) - \frac{1}{n^2}(\mathbf{1}^T D\mathbf{1}^T)\mathbf{1}\mathbf{1}^T$$

that

$$2\langle \mathbf{x}_i, \; \mathbf{x}_j\rangle = -D_{ij} + \frac{1}{n}\Big(\mathbf{1}^T D\mathbf{e}_j + \mathbf{e}_i^T D\mathbf{1}\Big) - \frac{1}{n^2}(\mathbf{1}^T D\mathbf{1}^T) \qquad i, j = 1, \ldots, n.$$

In particular, when $i = j$

$$\|\mathbf{x}_i\|^2 = \frac{1}{2n}\Big(\mathbf{1}^T D\mathbf{e}_i + \mathbf{e}_i^T D\mathbf{1}\Big) - \frac{1}{2n^2}(\mathbf{1}^T D\mathbf{1}^T).$$

Therefore, using the above identities we can verify that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\langle \mathbf{x}_i, \; \mathbf{x}_j\rangle = D_{ij} \qquad \text{for all} \;\; i, j = 1, \ldots, n. \tag{2.22}$$

Hence, $D \in \mathcal{E}^n$. This finishes the "If part" proof.

We now prove $D = \mathcal{D}(B)$. The way in (2.22) to get the distance matrix from $X$ can be written in terms of the matrix $B$ by noting

$$\|\mathbf{x}_i\|^2 = B_{ii}, \qquad \mathbf{x}_i^T \mathbf{x}_j = B_{ij}.$$

Hence

$$D_{ij} = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{x}_i^T \mathbf{x}_j = B_{ii} + B_{jj} - 2B_{ij}.$$

We can even get rid of the indices $(i, j)$ in the above equation to define the whole matrix $D$ through the $\mathcal{D}(B)$ as a mapping from $B$ to $D$. That is $D = \mathcal{D}(B)$. □

In the above proposition, the mapping $D(\cdot)$ actually applies to any matrix in $\mathcal{S}^n_+$, not necessarily the one defined by $-0.5JDJ$. This brings out a subtle fact that is worth of some appreciation.

For a given $B \in \mathcal{S}^n_+$, there are many matrices, represented as $\widetilde{D} \in \mathcal{S}^n$ such that $J\widetilde{D}J = -2B$. In fact, there are infinitely many elements in the following set:

$$\left\{ \widetilde{D} \in \mathcal{S}^n \;\middle|\; -\frac{1}{2} J\widetilde{D}J = B \right\} \tag{2.23}$$

The matrix $D$ obtained through the mapping (2.20) is the one in (2.23) that has zero diagonals. To see why (2.23) contains infinitely many elements, we note that

$$J\left(D + c\mathbf{1}\mathbf{1}^T\right)J = JDJ + cJ\mathbf{1}\mathbf{1}^T J = JDJ \qquad \text{for any } c \in \Re.$$

Hence,

$$\left\{ D + c\mathbf{1}\mathbf{1}^T \;\mid\; c \in \Re \right\}$$

is a subset of (2.23).

We finally emphasize that the embedding space in Proposition 2.8 is $\Re^N$, where $N$ can be very large. The embedding points by the PCoA (2.13) are in $\Re^r$, which is the lowest embedding space from $D$. PCoA has found important applications in ecology [56].

### 2.2.3 Equivalence of PCA and PCoA

We emphasize that PCA works when a covariance matrix $S$ is given and PCoA works when an EDM $D$ is given. In the scenario where the data matrix $X$ is given in (2.5) with $\mathbf{x}_i \in \Re^p$, we can calculate the covariance matrix $S$ by (2.7) and $D$ by

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

Consequently, we calculate the PCA matrix $S_{\texttt{pca}}$ in (2.10) and PCoA matrix $S_{\texttt{pcoa}}$ in (2.13). We have the following result.

**Proposition 2.12.** *(Equivalence between PCA and PCoA) In the setting above, we have*

$$S_{pca} = S_{pcoa}.$$

**Proof.** We note that the data points $\mathbf{x}_i$ have been mean-corrected. That is, they satisfy the centralization condition (2.6). It then follows from Prop. 2.8 that

$$B = X^T X.$$

Since $X^T X$ and $XX^T$ share the same set of (non-zero) eigenvalues, we known that

$$\lambda_i = \mu_i > 0, \qquad i = 1, \dots, r \qquad \text{and} \qquad \lambda_{r+1} = \mu_{r+1} = 0,$$

where $r$ is the rank of $B$. Therefore, the sizes of $S_{\texttt{pca}}$ and $S_{\texttt{pcoa}}$ are same.

Next, we show that the eigenvector $\mathbf{u}_i$ can be obtained through $\mathbf{v}_i$. We note that

$$X^T X\Big(X^T \mathbf{v}_i\Big) = X^T \Big(XX^T \mathbf{v}_i\Big) = X^T \Big(\lambda_i \mathbf{v}_i\Big) = \lambda_i \Big(X^T \mathbf{v}_i\Big),$$

where we used the fact that $\mathbf{v}_i$ is the eigenvector of $XX^T$. The above identity means that $(X^T \mathbf{v}_i)$ is the eigenvector of $X^T X$ corresponding to the eigenvalue $\lambda_i$. Moreover

$$\mathbf{v}_i^T XX^T \mathbf{v}_i = \lambda_i \|\mathbf{v}_i\|^2 = \lambda_i.$$

Hence, we can choose

$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}} X^T \mathbf{v}_i, \quad i = 1, \dots, r.$$

Finally, we note that the $i$th column in $S_{\texttt{pcoa}}$ is

$$\sqrt{\lambda_i}\mathbf{u}_i = \sqrt{\lambda_i}\frac{1}{\sqrt{\lambda_i}} X^T \mathbf{v}_i = X^T \mathbf{v}_i,$$

for $i = 1, \ldots, r$. Therefore, we must have $S_{\texttt{pca}} = S_{\texttt{pcoa}}$. This completes our proof. □

We must emphasize that the equivalence is just for the special case where the data matrix $X$ is given in (2.5) satisfying (2.6). In practice, we are often given a distance matrix without knowing any data points. In such a situation, we can use PCoA to generate coordinates of the data points, followed by further analysis on those generated data points. A good example for such a procedure is [56].

### 2.2.4 Further Comments

Roughly speaking, PCA works with correlation/covariance matrices and MDS works with dissimilarity matrices (e.g., Euclidean distance matrices (EDM)). A large body of work has been done involving dissimilarity matrices, see the books [26, 12, 66], and the recent papers/surveys [15, 53, 32, 44], to just name a few.

When each $\delta_{ij}$ 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 $\delta_{ij}$ contains noise, e.g., $\delta_{ij} = d_{ij} + \epsilon_{ij}$ with $\epsilon_{ij}$ 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 [84] based on a perturbation analysis. However, when some $\delta_{ij}$ takes the form: $\delta_{ij} = d_{ij} + \epsilon_{ij} + \eta_{ij}$ with $\eta_{ij}$ being big measurement error (such $\delta_{ij}$ 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 $\delta_{ij}$. This phenomenon has been highlighted in [18] and motivated Forero and Giannakis [35] to propose a sparsity-exploiting robust MDS (RMDS) for outlier removal. It makes use of the Kruskal stress function [54] as MDS criterion with $\ell_1$-based regularizations, a popular sparsity-induced technique widely used in machine learning and compressed sensing. However, it was observed in [59] that the subproblem solution of RMDS is actually the least-square (LS) solution of a residual equation and hence is "*strongly influenced by outliers*" [59, Sect. III]. The LS solution is replaced by the $M$-estimator, resulting in several robust algorithms depending on the $M$-estimator being used. We refer to [60] for further development along this line, in particular on using $\ell_{21}$ regularization. We note that the models behind those methods are non-convex optimization.

In summary (also as comprehensively demonstrated in [15]), `cMDS` works well when all $\delta_{ij}$ are available nd contain small noises. Furthermore, it is observed that larger dissimilarities dominate smaller ones, leading to scaling issues when properly using `cMDS`. Remember our case is that some of $\delta_{ij}$ are severely distorted.

## 2.3 Computing the Nearest EDM

The review so far has assumed that the true EDM $D$ is given, mainly via the data matrix $X$ in (2.5). However, $D$ is hardly an EDM in practice due to various reasons, e.g., inaccurate measurements or missing values in $X$. Such matrix $D$ is so important that we call it *pre-distance* matrix.

**Definition 2.13.** (Pre-distance matrix) A matrix $D \in \mathcal{S}^n$ is called a pre-distance matrix if

$$\mathrm{diag}(D) = 0 \qquad \text{and} \qquad D_{ij} \geq 0, \quad i, j = 1, \ldots, n.$$

The following example is motivated by an example in [56] to demonstrate a predistance matrix can not be embedded in $\Re^2$.

**Example 2.1.** *The predistance matrix $D$ is given by*

$$D = \begin{bmatrix} 0 & 4 & 4 & 1 \\ 4 & 0 & 4 & 1 \\ 4 & 4 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

*It is represented in the Figure 2.4.*

The question now becomes what we do about it if $D$ is just a pre-distance matrix. One simple answer is to make it to be Euclidean, but how? A natural question is to find the

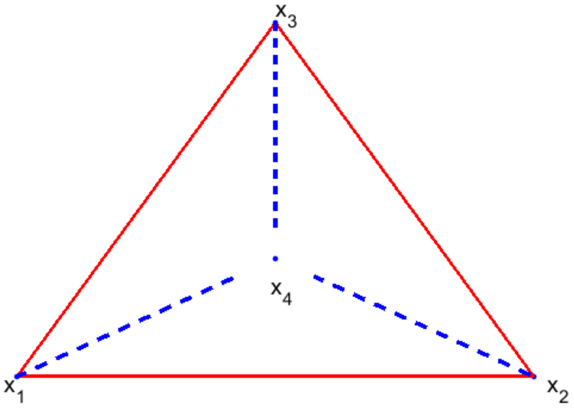


Figure 2.4: Pre-distance matrix $D$ in Example 2.1 among four points. The distances are constructed in such a way that the system cannot be represented in $\Re^2$ because the three lines going toward $x_4$ do not meet.

**nearest EDM** from $D$ and this gives the following optimization problem:

$$\min\ f(Y) := \frac{1}{2}\|Y - D\|^2 \qquad \text{s.t.}\ \ Y \in \mathcal{E}^n. \tag{2.24}$$

We will review some important methods for (2.24) and many of its variants. The first one is the famous Semi-Definite Programming (SDP) approach.

### 2.3.1 SDP approach

The basic idea of the SDP approach is to replace the Euclidean distance matrix cone $\mathcal{E}^n$ by the positive semidefinite cone $\mathcal{S}^n$, which has been well studied and is believed to be more conducive to developing efficient numerical methods, given the excellent progress made over the past thirty years. We present two reformulation of (2.24) in terms of SDP.

**(a) The first formulation.** We recall the mapping $\mathcal{D}(\cdot)$ from (2.20), which define an EDM from any positive semidefinite matrix $B \in \mathcal{S}^n_+$. And the matrix $\mathcal{D}(B)$ is the one in (2.23) that has zero diagonals. This fact follows from the proof in Prop. 2.8, where the data matrix $X$ is centralized. Recall $B = X^T X$. The centralization is equivalent to (see (2.21)):

$$B\mathbf{1} = 0.$$

Putting those facts together amounts to the following characterization of $\mathcal{E}^n$. A different proof of this characterization can be found in [50, Sect. 2].

**Proposition 2.14.** *(Positive semidefinite characterization of EDM) We have*

$$\mathcal{E}^n = \mathcal{D}^n := \left\{ \mathcal{D}(B) \;\middle|\; B \in \mathcal{S}^n_+, \quad B\mathbf{1} = 0 \right\}.$$

**Proof.** The proof is just a collection of the known results proved so far. Let $D \in \mathcal{E}^n$. Define, $B = -0.5JDJ$. It follows from Cor. 2.9 that $B \in \mathcal{S}^n_+$ and from the fact $J\mathbf{1} = 0$ in Prop. 2.3 that $B\mathbf{1} = 0$. Moreover, Prop. 2.11 implies $\mathcal{D}(B) = D$. Hence, $\mathcal{E}^n \subseteq \mathcal{D}^n$.

Conversely, consider $B \in \mathcal{S}^n_+$ and $B\mathbf{1} = 0$. Then $B$ must have the decomposition $B = X^T X$ for some data matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$. Define the EDM $D$ by

$$D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{x}_i^T \mathbf{x}_j \qquad \text{for all} \;\; i, j = 1, \ldots, n.$$

Then $D = \mathcal{D}(B)$. We proved $\mathcal{D}^n \subseteq \mathcal{E}^n$. We therefore must have $\mathcal{E}^n = \mathcal{D}^n$. □

We note that in the converse part of the proof, we did not use the fact $B\mathbf{1} = 0$. The characterization leads to the widely used SDP formulation below:

$$\min \; \frac{1}{2}\|D - \mathcal{D}(B)\|^2, \qquad \text{s.t.} \quad B\mathbf{1} = 0, \quad B \in \mathcal{S}^n_+. \tag{2.25}$$

A large number of papers have used this formulation in various applications. We refer to Biswas and Ye [7], Dattorro [28], Toh [90], and Jiang, Sun and Toh [49] and the references therein.

**(b) The second formulation.** This formulation is based on a further characterization of the feasible set in $\mathcal{D}^n$ in Prop. 2.14. Define

$$\mathcal{S}^n_c := \left\{ B \in \mathcal{S}^n \;\middle|\; B\mathbf{1} = 0 \right\} \qquad \text{(the centred subspace)}$$

Then the feasible region in $\mathcal{D}^n$ is $\mathcal{S}^n_+ \cap \mathcal{S}^n_c$. We need the following result.

**Lemma 2.15.** *(Characterization of the centred subspace) Let* $V \in \Re^{n\times(n-1)}$ *satisfy*

$$V^T V = I_{n-1} \qquad \textit{and} \qquad V^T \mathbf{1} = 0.$$

*(this basically requires that the columns of* $V$ *form an orthonormal basis for the subspace* $\mathbf{1}^\perp$*). Then*

$$\mathcal{S}^n_c = \mathcal{F}^n_c := \left\{ VAV^T \;\middle|\; A \in \mathcal{S}^{n-1} \right\}.$$

**Proof.** It is obvious that $VAV^T\mathbf{1} = 0$, and hence $VAV^T \in \mathcal{S}^n_c$ for any $A \in \mathcal{S}^{n-1}$. Therefore, $\mathcal{F}^n_c \subseteq \mathcal{S}^n_c$. We only need to prove the converse part.

Let $B \in \mathcal{S}^n_c$. Since $B\mathbf{1} = 0$, 0 is an eigenvalue of $B$. Then $B$ admits the following SVD:

$$B = P \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n-1} \end{bmatrix} P^T,$$

where $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_{n-1}$ are the eigenvalues of $B$ (we set the last eigenvalue $\lambda_n = 0$ and it does not contribute to the SVD), and $P \in \Re^{n\times(n-1)}$ satisfies $P^T P = I_{n-1}$ and $P^T\mathbf{1} = 0$ (because the vector $\mathbf{1}$ is an eigenvector of $B$ corresponding to the last eigenvalue $\lambda_n = 0$). This implies that the columns of $P$ spans the subspace $\mathbf{1}^\perp$. Hence they can be obtained through $V$:

$$P = VC,$$

where $C \in \Re^{(n-1)\times(n-1)}$. Substituting it to the SVD of $B$, we have

$$B = V \underbrace{C \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n-1} \end{bmatrix} C^T}_{\in\, \mathcal{S}^{n-1}} V^T.$$

Hence, $B \in \mathcal{F}^n_c$. This proves $\mathcal{S}^n_c \subseteq \mathcal{F}^n_c$. We establish the result. □

Consequently, the feasible set in $\mathcal{D}^n$ will become $\mathcal{S}^n_+ \cap \mathcal{F}^n_c$, which is equivalent to (using the same proof technique in Lemma 2.15):

$$\mathcal{S}^n_+ \cap \mathcal{F}^n_c = \mathcal{F}^n_{c+} := \left\{ VAV^T \;\middle|\; A \in \mathcal{S}^{n-1}_+ \right\}.$$

The second reformulation of (2.24) is

$$\min \ \frac{1}{2}\|D - \mathcal{D}(VAV^T)\|^2, \qquad \text{s.t.} \quad A \in \mathcal{S}^{n-1}_+. \tag{2.26}$$

This formulation has been first used by Alfakih et al. [1], see also [32] for a short review on it. A particular choice of $V$ is given in [1]:

$$V = \begin{bmatrix} p & p & \cdots & p \\ 1+q & q & \cdots & q \\ q & 1+q & \cdots & q \\ \vdots & \cdots & \ddots & \vdots \\ q & q & \cdots & 1+q \end{bmatrix},$$

where $p := -1/(n+\sqrt{n})$ and $q := -1/\sqrt{n}$.

### 2.3.2 EDM approach

This approach is based on a different characterization of EDM in Prop. 2.11, where a predistance matrix $D$ is an EDM if and only if $(-JDJ) \succeq 0$. We define the *almost negative semidefinite cone* by

$$\begin{aligned} \mathcal{K}^n_- &:= \left\{ Y \in \mathcal{S}^n \;\middle|\; -JYJ \in \mathcal{S}^n_+ \right\} \\ &= \left\{ Y \in \mathcal{S}^n \;\middle|\; Y \preceq 0 \ \text{ on } \ \mathbf{1}^\perp \right\}. \end{aligned} \tag{2.27}$$

This cone has been first studied Schoenberg [81] and more results can be found in Micchelli [64]. Also see Qi [71] for an elementary introduction of it with some important applications. The *conditionally positive semidefinite cone* is then defined to be

$$\mathcal{K}^n_+ := -\mathcal{K}^n_-.$$

Therefore, a matrix $Y \in \mathcal{E}^n$ if and only if

$$Y \in \mathcal{S}^n_h \qquad \text{and} \qquad Y \in \mathcal{K}^n_-.$$

Consequently, the problem (2.24) can be reformulated as

$$\min \ \frac{1}{2}\|Y - D\|^2, \qquad \text{s.t.} \qquad Y \in \mathcal{S}^n_h \ \text{ and } \ Y \in \mathcal{K}^n_-. \tag{2.28}$$

The question now comes to whether this formulation can be efficiently solved. Several major methods have been developed and they all depend on efficient computation of the orthogonal projection onto $\mathcal{K}^n_-$:

$$\Pi_{\mathcal{K}^n_-}(D) := \arg\min \left\{ \|Y - D\| \ \middle| \ Y \in \mathcal{K}^n_- \right\}.$$

There are two known formulae for computing $\Pi_{\mathcal{K}^n_-}(D)$. One is the formula by Gaffke and Mathar [37, Eq. (29)]:

$$\Pi_{\mathcal{K}^n_-}(D) = D - \Pi_{\mathcal{S}^n_+}(JDJ). \tag{2.29}$$

The other is due to Glunt et al. [40], which we omit here as we are not going to use it. Below we briefly outline two important numerical methods that will be useful to our late development.

The most famous characterization of EDM is due to [81]:

$$D \in \mathcal{D}^n \quad \text{if and only if} \quad \operatorname{diag}(D) = 0, \quad -D \in \mathcal{K}^n_+, \tag{2.30}$$

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

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

By using the centralizing matrix $J$, we have

$$\mathcal{K}_+^n = \left\{ A \in \mathcal{S}^n \mid JAJ \in \mathcal{S}_+^n \right\}. \tag{2.31}$$

The projection onto $\mathcal{K}_+^n$ can be calculated by the formula of [37, Eq.(29)]:

$$\Pi_{\mathcal{K}_+^n}(A) = A + \Pi_{\mathcal{S}_+^n}(-JAJ), \quad \forall \ A \in \mathcal{S}^n. \tag{2.32}$$

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 $r$:

$$\mathcal{K}_+^n(r) := \left\{ D \in \mathcal{K}_+^n \mid \text{rank}(JDJ) \leq r \right\}. \tag{2.33}$$

We call it the rank-$r$ cut of the conditionally positive semidefinite cone. It is extensively studied in [75, 102]. Let $A \in \mathcal{S}^n$ be given, define the distance from $A$ to $\mathcal{K}_+^n(r)$:

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

and define the squared distance function

$$g(A) := \frac{1}{2}\text{dist}^2(-A, \ \mathcal{K}_+^n(r)). \tag{2.34}$$

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

$$\begin{aligned} & D \in \mathcal{D}^n, \ \text{rank}(JDJ) \leq r \\ \overset{(2.30)}{\Longleftrightarrow} \quad & \text{diag}(D) = 0, \ -D \in \mathcal{K}_+^n, \ \text{rank}(JDJ) \leq r \\ \overset{(2.33)}{\Longleftrightarrow} \quad & \text{diag}(D) = 0, \ -D \in \mathcal{K}_+^n(r) \\ \overset{(2.34)}{\Longleftrightarrow} \quad & \text{diag}(D) = 0, \ g(D) = 0. \end{aligned} \tag{2.35}$$

Moreover, [102, Lemmas 2.1, 2.2] implies that the function

$$h(A) := \frac{1}{2}\|A\|^2 - g(-A) \tag{2.36}$$

is convex and we can calculate one of its subgradients by

$$\Pi_{\mathcal{K}_+^n(r)}(A) \in \partial h(A), \tag{2.37}$$

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 (2.36), the convexity of $h(\cdot)$ and (2.37) that

$$\begin{aligned} g(D) &= \frac{1}{2}\|D\|^2 - h(-D) \\ &\le \frac{1}{2}\|D\|^2 - h(-A) + \langle \Pi_{\mathcal{K}_+^n(r)}(-A), D - A\rangle \\ &=: g_m(D, A), \qquad \forall\ D, A \in \mathcal{S}^n. \end{aligned} \tag{2.38}$$

We call $g_m(D, A)$ a majorization of $g(D)$.

**(a) Method of Alternating Projection (MAP).** We note that the constraint set in (2.28) is the intersection of one subspace $\mathcal{S}_h^n$ and a closed convex cone $\mathcal{K}_-^n$. This method starts from any point, say $Y^0 \in \mathcal{S}^n$. It then first projects this point to $\mathcal{K}_-^n$ and then projects the resulting point to $\mathcal{S}_h^n$:

$$Y^1 = \Pi_{\mathcal{S}_h^n}\Big(\Pi_{\mathcal{K}_-^n}(Y^0)\Big).$$

The process continues until convergence is observed. More formally, the iterates can be represented as

$$Y^k = \Big(\Pi_{\mathcal{S}_h^n}\Pi_{\mathcal{K}_-^n}\Big)^k (Y^0), \quad k = 1, 2, \ldots,$$

This process is illustrated in Fig. 2.5. This method was first applied to (2.28) in Glunt et al. [40] and was recently used by Zhang et al. [100]. Based on extensive test by Qi [70], the Newton method is much faster than MAP. We describe Newton's method below.

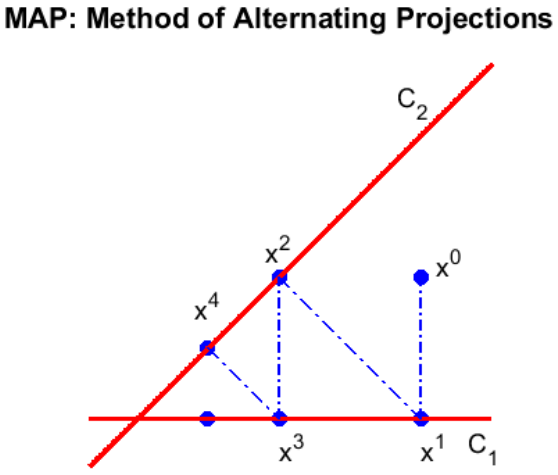


Figure 2.5: Illustration of the method of alternating projections. $C_1$ is the positive $x$ axis and $C_2$ is the subspace defined the straight line.

**(b) Newton's Method**. Newton's method was first developed by Qi [70] for the problem (2.28). It is developed for its dual problem, which is described below for the benefit of future use. We first note that the constraint $Y \in \mathcal{S}^n_h$ in (2.28) can be reformulated as

$$\mathcal{A}(Y) := \mathrm{diag}(Y) = 0,$$

where $\mathcal{A} : \mathcal{S}^n \mapsto \Re^n$ can be viewed as a mapping. Let the Lagrange function for (2.28) be defined by

$$\begin{aligned} L(Y;\mathbf{y}) &:= \frac{1}{2}\|Y - D\|^2 - \langle \mathbf{y},\ \mathrm{diag}(Y)\rangle \\ &= \frac{1}{2}\|Y - D\|^2 - \langle \mathrm{Diag}(\mathbf{y}),\ Y\rangle \\ &= \frac{1}{2}\|Y - (D + \mathrm{Diag}(\mathbf{y}))\|^2 - \frac{1}{2}\|D + \mathrm{Diag}(\mathbf{y})\|^2 + \frac{1}{2}\|D\|^2. \end{aligned}$$

The Lagrangian dual problem is defined to be (see [46, Chapter XII]):

$$\max_{\mathbf{y}\in\Re^n} \ \theta(\mathbf{y}) := \min\left\{L(Y;\mathbf{y}) \mid Y \in \mathcal{K}^n_-\right\}.$$

Although it looks complicated, $\theta(\mathbf{y})$ actually enjoys some properties, which we will see below.

$$\begin{aligned}
\theta(\mathbf{y}) &= \min_{Y \in \mathcal{K}^n_-} \ L(Y; \mathbf{y}) \\
&= \underbrace{\left\{ \min_{Y \in \mathcal{K}^n_-} \ \frac{1}{2}\|Y - (D + \mathrm{Diag}(\mathbf{y}))\|^2 \right\}}_{\text{projection}} \underbrace{- \frac{1}{2}\|D + \mathrm{Diag}(\mathbf{y})\|^2 + \frac{1}{2}\|D\|^2}_{Y \text{ is not involved}} \\
&= \underbrace{\frac{1}{2}\|\Pi_{\mathcal{K}^n_-}(D + \mathrm{Diag}(\mathbf{y})) - (D + \mathrm{Diag}(\mathbf{y}))\|^2 - \frac{1}{2}\|D + \mathrm{Diag}(\mathbf{y})\|^2}_{\text{Moreau decomposition}} + \frac{1}{2}\|D\|^2 \\
&= -\frac{1}{2}\|\Pi_{\mathcal{K}^n_-}(D + \mathrm{Diag}(\mathbf{y}))|^2 + \frac{1}{2}\|D\|^2.
\end{aligned}$$

We note that the Moreau decomposition takes the following general form:

$$\mathbf{x} = \Pi_{\mathcal{K}}(\mathbf{x}) + \Pi_{\mathcal{K}^*}(\mathbf{x}),$$

where $\mathcal{K} \in \Re^n$ is a closed convex cone and $\mathcal{K}^*$ is its dual cone

$$\mathcal{K}^* := \{\mathbf{y} \in \Re^n \mid \langle \mathbf{x}, \ \mathbf{y} \rangle \leq 0, \quad \forall \ \mathbf{x} \in \mathcal{K}\}.$$

It follows that $(\mathcal{K}^*)^* = \mathcal{K}$.

Hence the Lagrangian dual problem of (2.28) becomes

$$\max_{\mathbf{y} \in \Re^n} \ \theta(\mathbf{y}) = -\frac{1}{2}\|\Pi_{\mathcal{K}^n_-}(D + \mathrm{Diag}(\mathbf{y}))\|^2 + \frac{1}{2}\|D\|^2,$$

which only involves vector $\mathbf{y}$. We note that the number of variables in just $n$. It further becomes when put in the form of minimization:

$$\min_{\mathbf{y} \in \Re^n} \ -\theta(\mathbf{y}) = \frac{1}{2}\|\Pi_{\mathcal{K}^n_-}(D + \mathrm{Diag}(\mathbf{y}))\|^2 - \frac{1}{2}\|D\|^2. \tag{2.39}$$

The function $(-\theta(\mathbf{y}))$ is convex (see [65, Lemma 2.1(iii)] and [46, Chapter IV: Example 2.14]). The optimality condition of (2.39) is

$$F(\mathbf{y}) := -\nabla\theta(\mathbf{y}) = \mathrm{diag}\Big(\Pi_{\mathcal{K}^n_-}(D + \mathrm{Diag}(\mathbf{y}))\Big) = 0. \tag{2.40}$$

The Newton method is to solve this optimality condition. Starting from an arbitrary point $\mathbf{y}^0 \in \Re^n$, $k = 0$, compute the next iterate by

$$\mathbf{y}^{k+1} = \mathbf{y}^k - V_k^{-1} F(\mathbf{y}^k) \qquad \text{with} \;\; V_k \in \partial F(\mathbf{y}^k) \tag{2.41}$$

where $\partial F(\mathbf{y}^k)$ denotes the subdifferential of $F\cdot)$ at $\mathbf{y}^k$ ($F(\mathbf{y})$ is usually not differentiable because it involves the projection operator $\Pi_{\mathcal{K}_-^n}(\cdot)$. However, its subdifferential can be well-defined, see [24]). The convergence and its implementation of Newton's method can be found in [70]. We omit their rather technical details.

### 2.3.3 Some Comments

We make some comments on the key differences between SDP and EDM approaches.

(i) SDP appears to have been a dominating method for solving (2.24) (and its variants), probably because (a) (2.24) can be easily reformulated as SDP and (b) SDP has been well documented and gained a high reputation in that a problem can be efficiently solved if it can be put into the form of SDP. For example, Wolkowicz and his follows have made a great contribution in promoting the use of SDP. They, on the one hand, emphasize the importance of EDM as an object to describe distance information, and on the other hand, opt to reformulate their problems as SDP, see [1, 53**?** ]. Another important examples along this line of research include [92, 7, 28, 90, 69, 52] (to just name a few) and many reference therein.

(ii) Many researchers have realized the importance of EDM, or in general the importance of the almost negative semidefinite cone $\mathcal{K}_-^n$. Schoenberg [81] is probably the first to utilize it to characterize metric embedding in Hilbert space. Another importance reference on $\mathcal{K}_-^n$ is Micchelli [64] (see also [71] for more on those two references). Glunt et al. [40] and Gaffke and Mathar [37] are the first to solve (2.28) directly through the use of the method of alternating projections (MAP). The algorithm has found applications in molecular conformation [41, 42, 43]. The MAP is recently used by Zhang [100]. Newton's method has been used in a series of papers by Qi and his collaborators [70, 75, 4, 72]. In a recent survey [32], Doknabić et al. emphasized limitation of the SDP approach when the data points are of the size of thousands (e.g., it took too much time for a standard SDP solver to terminate).

(iii) In terms of constraint qualification, the so-called constraint nondegeneracy is satisfied for the EDM formulation (2.28) (see [70]), while it is hard to characterize the nondegeneracy for the SDP formulation (2.25) and (2.26). The constraint nondegeneracy has well been studied for SDP by Sun [86].

## 2.4 Sparse and Robust MDS Models

We recall the error model (1.2):

$$\delta_{ij} = d_{ij} + z_{ij} + \epsilon_{ij}, \qquad 1 \le i < j = n,$$

and the purpose is to find a set of embedding points $\mathbf{x}_i \in \Re^r$, $i = 1, \ldots, n$ such that $\|\mathbf{x}_i - \mathbf{x}_j\| = d_{ij}$. For the model to be interesting, the embedding dimension $r$ has to be small, e.g., $r = 2$ or 3.

Forero and Giannakis [35] further assumes that most of $z_{ij}$ should take the value 0. In other words, the matrix $Z = (z_{ij})$ should be sparse. They then propose the following sparsity-driven MDS model:

$$\min \sum_{i,j}^{n} \Big( \delta_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\| - z_{ij} \Big)^2 + \lambda \underbrace{\sum_{i,j}^{n} |z_{ij}|}_{=:\|Z\|_1}, \tag{2.42}$$

where $\lambda > 0$ is a penalty parameter and the $\ell_1$ norm $\|Z\|_1$ is to drive most of the elements in $Z$ to 0. A majorization algorithm is developed for the model (2.42) in [35]. We also like to note that the model is nonconvex due to the term $\|\mathbf{x}_i - \mathbf{z}_j\|$.

The model (2.42) is actually least-squares based. It is known that the least squares term can be made robust by replacing it with its robust variants:

$$\min \sum_{i,j}^{n} \phi\Big(\delta_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\| - z_{ij}\Big) + \lambda \underbrace{\sum_{i,j}^{n} |z_{ij}|}_{=:\|Z\|_1}, \tag{2.43}$$

where $\phi(\cdot)$ is a robust function. For example, $\phi(\cdot)$ can be taken as the Huber function:

$$\phi_H(x) := \begin{cases} \frac{x}{2} & \text{if } |x| \le a \\ a|x| - \frac{a^2}{2} & \text{if } |x| > a, \end{cases}$$

where $a$ is the threshold that can be chosen arbitrarily or adaptively from data. This model has been considered by Mandanas and Kotropoulos [59] with many choices for $\phi(\cdot)$.

However, the algorithm RMDS developed for the models (2.42) [35] and HQMMDS for the model (2.43) [59] cannot handle the important constraints:

$$\ell_{ij} \le \|\mathbf{x}_i - \mathbf{x}_j\| \le u_{ij}, \quad i, j = 1, \ldots, n,$$

where $\ell_{ij}$ and $u_{ij}$ are lower and upper bounds for the true distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The EDM approach developed in this thesis can handle such lower and upper bound constraints. For a good survey on other sparse and robust MDS models, see the literature review part of [59].

## 2.5 Procrustes Analysis

It is often necessary that to map one set of points to a set of another points through linear transformations. This can be done by the Procrustes analysis, which is detailed in [26, Chapter 5] and [12, Chapter 20]. Suppose we have two sets of points:

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n], \qquad Y = [\mathbf{y}_i, \mathbf{y}_2, \ldots, \mathbf{y}_n]$$

with all points in a same Euclidean space. The purpose is to find a linear transformation $T$ such that

$$\mathbf{y}_i \approx \rho T \mathbf{x}_i + \mathbf{c}, \qquad i = 1, \ldots, n$$

where $T$ is a rotation/reflection matrix, $\rho$ is a scaling constant and $\mathbf{c}$ is a translation vector. We choose the best $T$, $\rho$ and $\mathbf{c}$ through the least-squares criterion:

$$\min \sum_{i=1}^{n} \|\mathbf{y}_i - \rho T \mathbf{x}_i + \mathbf{c}\|^2.$$

MATLAB has a built-in function, `procrustes.m` to compute the best $\rho$, $T$ and $\mathbf{c}$.

## 2.6 Conclusion

In the chapter, we reviewed the classical MDS (`cMDS`) and its relationship to PCA. A key concept is the Euclidean Distance Matrix $D$, which has to satisfy the conditional negative semi-definite condition (i.e., in the conditionally negative semidefinite cone $\mathcal{K}^n$). For such $D$, `cMDS` will be able generate a set of embedding points that recover the distances in $D$. However, in practice, we are only given a dissimilarity matrix $\Delta$, which is rarely Euclidean. A commonly used approach is to find the nearest EDM from $\Delta$. There are two major ways in doing this. One is the SDP approach and the other is EDM approach. We reviewed two popular methods for the latter, namely the method of alternating projection and the Newton method. It is worth noting that

(i) Computing the nearest EDM is a convex problem because the constraints are convex and the objective is of least squares.

(ii) The dissimilarity matrix $\Delta$ does not have missing values.

(iii) Practical applications require the embedding points sitting in a low dimensional Euclidean space. Hence, those methods reviewed are not applicable, but they can provide good initial points for the low-dimensional case.

We also reviewed some sparse and robust models that generalize `cMDS`. Those models are nonconvex and can handle the case of missing values. But they are incapable of handling the lower and upper bound constraints:

$$\ell_{ij} \leq \|\mathbf{x}_i - \mathbf{x}_j\| \leq u_{ij}, \quad i, j = 1, \ldots, n.$$

The methods developed in the next chapters are based on EDM and can handle those lower and upper bounds.

# Chapter 3

# A New Reformulation of cMDS and Its Implications

## 3.1 Introduction

In this part, we will first describe the model. We will focus on one particular instance, where a sparse constraint is enforced. We will use a small example to motivate this case. Following the standard approach to dealing with sparsity, we propose an $\ell_1$-regularized model. This has led to a convex optimization that can be efficiently solved. The rest of the paper aims to substantiate those claims.

### 3.1.1 A General Model

We propose a general model of the following form:

$$\begin{array}{ll} \min_{D,\mathbf{y}} & \|\Delta - (D + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T)\|^2 \\ \text{s.t.} & D \in \mathcal{D}^n, \quad \mathbf{y} \in V, \end{array} \tag{3.1}$$

where $\mathcal{D}^n$ is the cone of Euclidean distance matrices (EDM), $V \subseteq \Re^n$ is a closed subset in $\Re^n$, $\Delta \in \mathcal{S}^n$ (the space of all $n \times n$ symmetric matrices) is a given pre-distance matrix,

$\mathbf{e}$ is the (column) vector of all ones in $\Re^n$, and the norm $\|\cdot\|$ is the Frobenius norm in $\mathcal{S}^n$.

The model (3.1) is very general and includes several known models as its special cases. For instance, when $V = \{0\}$, it is the well-known nearest EDM problem studied in [40, 70]. When $V = \Re^n$, (3.1) is the classical MDS documented in [26, 12] (see Theorem 3.4). When

$$V = \{\mathbf{y} \in \Re^n \mid y_1 = y_2 = \cdots = y_n\},$$

it is the additive constant problem studied in [72]. We summarize them in Table 3.1.

Table 3.1: Methods covered by the general model (3.1)

| Model (3.1) | Known Methods |
|---|---|
| $V = \Re^n$ | `cMDS` [12, 26] |
| $V = \{0\}$ | `NEDM` [40, 70] |
| $V = \{\mathbf{y} \in \Re^n \mid y_1 = \cdots = y_n\}$ | Additive constant problem [58, 72] |

In this chapter, we will focus on the following choice of $V$

$$V := \{\mathbf{y} \in \Re^n \mid \|\mathbf{y}\|_0 \le s\}, \tag{3.2}$$

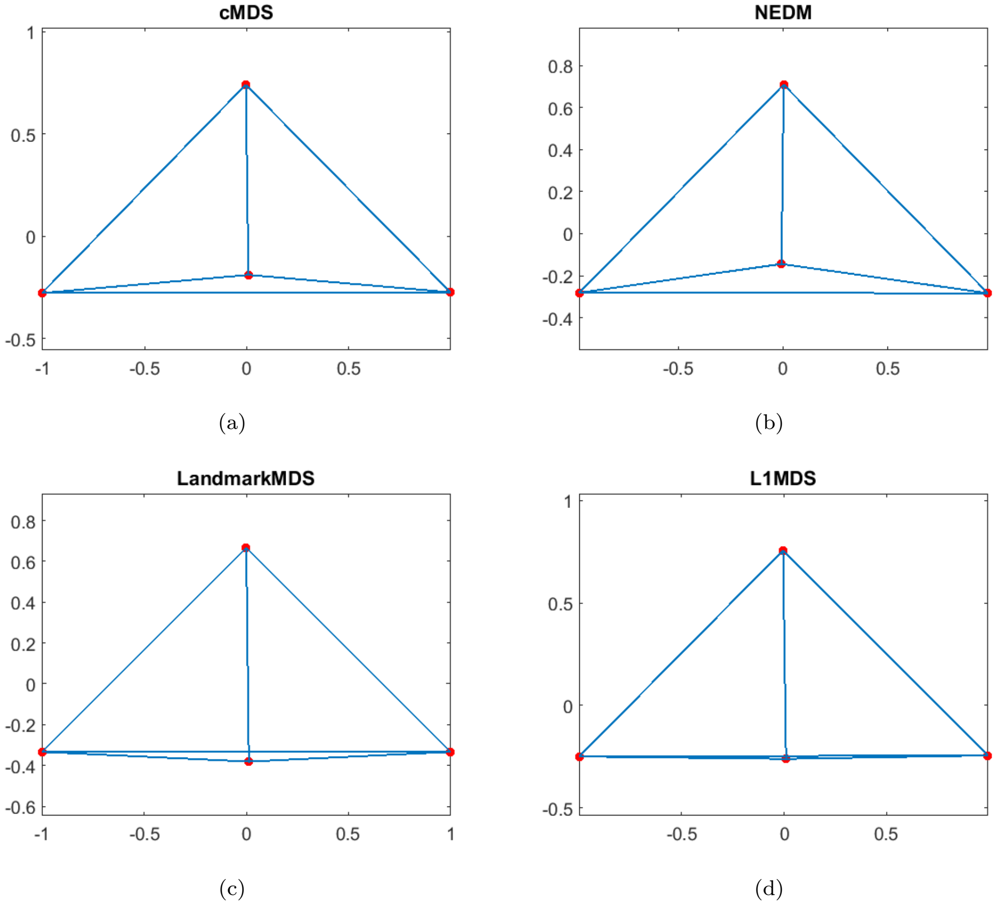
where $\|\mathbf{y}\|_0$ is the number of nonzero elements in $\mathbf{y}$ (i.e., the zero-norm of $\mathbf{y}$) and $s$ is a given positive integer. This constraint in (3.2) is known as the sparse constraint in optimization and compressed sensing. The main reason why this choice of $V$ is important is well illustrated by the following example.

### 3.1.2 A motivating example

We use a small example to illustrate a situation where existing MDS-based methods fail to work. Suppose we have 4 points $\mathbf{x}_1^T = (0,0)$, $\mathbf{x}_2^T = (-1,0)$, $\mathbf{x}_3^T = (1,0)$, and $\mathbf{x}_4^T = (0,1)$ forming a right triangle with $\mathbf{x}_1$ sitting in the middle of the hypotenuse. Let $D := \left(\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$ be the squared Euclidean distance matrix (EDM) by those four points. Suppose $\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ are fixed (known as anchors/landmarks) and we assume that the distances from $\mathbf{x}_1$ to the anchors contain large errors (in bold-faced numbers below). The node $\mathbf{x}_1$ is referred to as a faulty node. The corresponding distance matrix, denoted by $\Delta$, is given by

$$\Delta = \begin{pmatrix} 0 & \mathbf{0.45} & \mathbf{0.5} & \mathbf{0.57} \\ \mathbf{0.45} & 0 & 4 & 2 \\ \mathbf{0.5} & 4 & 0 & 2 \\ \mathbf{0.57} & 2 & 2 & 0 \end{pmatrix} \quad \text{and} \quad S := \Delta - D = - \begin{pmatrix} 0 & 0.55 & 0.5 & 0.43 \\ 0.55 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.43 & 0 & 0 & 0 \end{pmatrix},$$

where the matrix $S$ is called the discrepancy matrix between the observed distance matrix $\Delta$ and the true EDM $D$.



(a)

(b)

(c)

(d)

Figure 3.1: Reconstruction of the right triangle by `cMDS` (a), `NEDM` (b), `LandmarkMDS` (c) and `L1MDS` (d). All except `LL1MDS` failed to recover the correct structure of the triangle, which has two important features: the right angle and the middle point on the hypotenuse. All methods are implemented in MATLAB.

We applied three popular MDS methods to the data matrix $\Delta$. They are `cMDS` [26, 12], `NEDM` (the nearest EDM method [40, 70]), and `LL1MDS` proposed in this chapter. The reconstruction of the triangle by those methods are reported in Fig. 3.1.

What have gone wrong with those methods was that `cMDS` used a dense matrix to approximate the sparse discrepancy matrix $S$ (see Prop. 3.5), while `NEDM` and `LandmarkMDS` did not even approximate this sparse matrix. If we set $s = 1$ in (3.2), then

$$Z := \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T$$

will be a a rank-one sparse matrix that may well approximate $S$. We will show this is the case.

### 3.1.3 The Main Methodology: $\ell_1$ Regularization

Therefore, the problem that we are going to solve is

$$\begin{array}{ll} \min_{D,\mathbf{y}} & \|\Delta - (D + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T)\| \\ \text{s.t.} & D \in \mathcal{D}^n, \quad \|\mathbf{y}\|_0 \le s. \end{array}$$

This problem is nonconvex. A widely used approach is to use $\ell_1$ norm of $\mathbf{y}$ to approximate the 0-norm, leading to the following convex optimization problem:

$$\begin{array}{ll} \min_{D,\mathbf{y}} & \|\Delta - (D + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T)\| + \mu\|\mathbf{y}\|_1 \\ \text{s.t.} & D \in \mathcal{D}^n, \qquad \mathbf{y} \in \Re^n, \end{array} \tag{3.3}$$

where $\mu > 0$ is a given regularization parameter and

$$\|\mathbf{y}\|_1 := |y_1| + |y_2| + \cdots + |y_n|.$$

One of the purposes of this chapter is to develop a fast algorithm for (3.3). The following are established. (i) `cMDS` tends to use a dense matrix to approximate a sparse noise matrix. This means that if there exists few outliers, `cMDS` would fail to detect them. (ii) We develop a proximal ADMM method for the $\ell_1$-regularized model (3.3).

(iii) A small numerical example was used to demonstrate the effectiveness of the proposed approach.

The chapter is organized as follows. In Section 3.2, we will review the soft thresholding operator related to $\ell_1$ norm. In Section 3.3, we characterize MDS in terms of a joint optimization, which will lead to the general problem (3.1). In particular, we will show that `cMDS` always attempts to use a dense matrix to approximate the discrepancy matrix $S$, no matter whether it is spare or not. In Section 3.4, we will refine the $\ell_1$ problem (3.3) into a more efficient form and develop a fast semi-proximal method for it in Section 3.5. We report our preliminary results in Section 3.6.

## 3.2 Soft Thresholding Operator

When it comes to solve our $\ell_1$-regularized problem, we are going to face the following one-dimensional minimization problem:

$$y_* = \arg\min_{y \in \Re} \ \frac{c}{2}(\beta y - s)^2 + \mu |y|, \tag{3.4}$$

where $\beta > 0$, $c > 0$, $\mu > 0$ and $s \in \Re$ are given. This problem has a closed form solution. We consider two cases.

Case 1: Suppose $y \geq 0$. Then the problem (3.4) becomes

$$\begin{aligned}
& \arg\min_{y \geq 0} \ \frac{c}{2}(\beta y - s)^2 + \mu y \\
= \ & \arg\min_{y \geq 0} \ \frac{c}{2}\left[(\beta y - s)^2 + \frac{2\mu}{c} y\right]^2 \\
= \ & \arg\min_{y \geq 0} \ \frac{c}{2}\left[\beta y - s + \frac{\mu}{\beta c}\right]^2 \\
= \ & \arg\min_{y \geq 0} \ \left[\beta y - \left(s - \frac{\mu}{\beta c}\right)\right]^2 \\
= \ & \begin{cases} \frac{1}{\beta}\left(s - \frac{\mu}{\beta c}\right) & \text{if } s \geq \frac{\mu}{\beta c} \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Case 2: Suppose $y \leq 0$. Then the problem (3.4) becomes

$$\begin{aligned}
& \arg\min_{y\leq 0} \ \frac{c}{2}(\beta y - s)^2 - \mu y \\
= \ & \arg\min_{y\leq 0} \ \frac{c}{2}\left[\beta y - s)^2 - \frac{2\mu}{c} y\right]^2 \\
= \ & \arg\min_{y\leq 0} \ \frac{c}{2}\left[\beta y - s - \frac{\mu}{\beta c}\right]^2 \\
= \ & \arg\min_{y\leq 0} \ \left[\beta y - \left(s + \frac{\mu}{\beta c}\right)\right]^2 \\
= \ & \begin{cases} \frac{1}{\beta}\left(s + \frac{\mu}{\beta c}\right) & \text{if } s \leq -\frac{\mu}{\beta c} \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

Combining Case 1 and Case 2 above, we see that the solution $y_*$ of (3.4) is

$$y_* = \text{Shrink}_\beta(s, \mu/c) := \frac{1}{\beta} \max\left\{|s| - \frac{1}{\beta}\frac{\mu}{c}, \ 0\right\} \text{sign}(s), \tag{3.5}$$

where sign$(\cdot)$ is the sign function

$$\text{sign}(s) := \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0. \end{cases}$$

This type of formula (3.5) has been widely used in compressed sensing and is call the soft thresholding operator (see, e.g., [97]).

Application of (3.5) to the 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. Application of (3.5) to the above problem leads to its optimal solution denoted by

$$\mathcal{S}_\beta(t) := \max\{|t| - \beta, \ 0\}\text{sign}(t). \tag{3.6}$$

## 3.3 Joint Optimization of MDS

In this section, we cast cMDS as a joint optimization over the cone of EDMs and the subspace $\mathcal{S}_2^n$. This new viewpoint of cMDS leads us to consider its $\ell_1$ regularization.

We now describe `cMDS` from an optimization perspective. Suppose we have a set $n$ unknown points $\mathbf{x}_i \in \Re^r$, $i = 1, \ldots, n$ and a known observation matrix $\Delta$ with $\Delta_{ij} \approx \|\mathbf{x}_i - \mathbf{x}_j\|^2$ for all $i, j$. It is known that `cMDS` seeks the best EDM through the following optimization:

$$\min f_1(D) := \frac{1}{2}\|J(D - \Delta)J\|^2 \qquad \text{s.t.} \quad -JDJ \in \mathcal{S}_+^n \quad \text{and} \quad D \in \mathcal{S}^n. \tag{3.7}$$

The double centering matrix $J$ in $f_1(Y)$ is first introduced by Torgerson [91] to MDS. It is important to note that the objective function $f_1(D)$ in (3.7) is a semi-norm of $D$ [63] and hence the problem may have many solutions. In fact, it has infinitely many solutions. However, all solutions have a same $JDJ$. It is this same $JDJ$ that has a closed-form representation. There exists an EDM among those many solution. We denote this EDM by $D^{\mathtt{mds}}$. It can be computed by Alg. 2.

We note that `cMDS` measures the closeness between $D$ and $\Delta$ in terms of $\|JDJ - J\Delta J\|$, which is not a strongly convex function in $D$ [63] due to it being a semi-norm of $D$. A more direct measurement is $\|D - \Delta\|$. This results in the nearest EDM problem [40, 70, 32]:

$$\min f_2(D) := \frac{1}{2}\|D - \Delta\|^2 \qquad \text{s.t.} \quad D \in \mathcal{D}^n. \tag{3.8}$$

Let $D^{\mathtt{edm}}$ be its unique solution. The embedding points are obtained by applying Alg. 1 to $D^{\mathtt{edm}}$ instead of $D^{\mathtt{mds}}$. Although (3.8) is a more accurate approximation to $\Delta$ from $D^{\mathtt{edm}}$ than from $D^{\mathtt{mds}}$, it does not have a closed-form solution. Fortunately, there exist fast numerical methods for it, see [40, 70], see Section 2.3.2 for some algorithms . Despite both methods having been widely used when $D$ has random errors, our main interest here is on how they behave when $D$ contains few large measurement errors in addition to random errors.

### 3.3.1 A Subspace Perspective

We establish a few technical results before stating our new interpretation of cMDS. The first one is about the decomposition of the almost negative semidefinite cone.

**Lemma 3.1.** *The almost negative semidefinite cone $\mathcal{K}^n_-$ and the EDM cone $\mathcal{D}^n$ have the following relationship:*

$$\mathcal{K}^n_- = \mathcal{D}^n \ + \ \mathcal{S}^n_2. \tag{3.9}$$

*Moreover, given a matrix $D \in \mathcal{K}^n_-$, there exists a unique decomposition*

$$Y = D + \left(\mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T\right), \tag{3.10}$$

*with $D \in \mathcal{D}^n$ and $\mathbf{y} \in \Re^n$. In this case, $\mathbf{y}$ and $D$ are respectively determined by*

$$\mathbf{y} = \frac{1}{2}\mathrm{diag}(Y) \qquad \textit{and} \qquad D = Y - \left(\mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T\right). \tag{3.11}$$

**Proof.** For a given $Y \in \mathcal{K}^n_-$, let $\mathbf{y}$ and $D$ be defined by (3.11). We have $JYJ \in -\mathcal{S}^n_+$ by the definition of $\mathcal{K}^n_-$. It follows that $\mathrm{diag}(D) = 0$ and

$$JDJ = JYJ - J\left(\mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T\right)J = JYJ \in -\mathcal{S}^n_+,$$

where we used the fact $J\mathbf{e} = 0$. By the characterization of $\mathcal{D}^n$ in (2.30), we must have $D \in \mathcal{D}^n$. This establishes the inclusion

$$\mathcal{K}^n_- \subseteq \mathcal{D}^n + \mathcal{S}^n_2$$

and the uniqueness of the decomposition in (3.10). The reverse inclusion is obvious by noticing that

$$J(D+Z)J = JDJ \in -\mathcal{S}^n_+ \qquad \forall \ \ D \in \mathcal{D}^n \text{ and } Z \in \mathcal{S}^n_2.$$

Therefore, $D + Z \in \mathcal{K}^n_-$ for any $D \in \mathcal{D}^n$ and $Z \in \mathcal{S}^n_2$. This completes the proof. □

The following result is about an invariance property among all matrices in $\mathcal{K}^n_-$ satisfying certain property.

**Lemma 3.2.** *For any two matrices $Y_1, Y_2 \in \mathcal{K}^n_-$ satisfying $JY_1J = JY_2J$, there exists a unique EDM $D \in \mathcal{D}^n$ such that*

$$JDJ = JY_1J = JY_2J. \tag{3.12}$$

**Proof.** Since $Y_1 \in \mathcal{K}^n_-$, we have $-JY_1J \in \mathcal{S}^n_+$. Let it have the decomposition:

$$-\frac{1}{2}JY_1J = XX^T \qquad \text{with} \quad X := [\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_n].$$

Then the EDM $D \in \mathcal{D}^n$ defined by $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ for all $i, j$ satisfies the property (3.12) (see [26, Sect. 2.2]). Moreover, any EDM $D$ satisfying the property (3.12) must be generated by the embedding points $\{\mathbf{x}_i\}$. Therefore, such EDM must be unique. □

Let $\mathcal{S}^n_- := -\mathcal{S}^n_+$. $\mathcal{S}^n_-$ is known as the negative semidefinite cone in $\mathcal{S}^n$. It is easy to see that

$$A = \Pi_{\mathcal{S}^n_-}(A) + \Pi_{\mathcal{S}^n_+}(A) \qquad \forall\ \ A \in \mathcal{S}^n. \tag{3.13}$$

The centering matrix $J$ has the following decomposition:

$$J = Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q \qquad \text{with} \ \ Q := I - \frac{2}{n+\sqrt{n}}\mathbf{v}\mathbf{v}^T, \tag{3.14}$$

where $\mathbf{v}^T := (1, \ldots, 1, \sqrt{n}+1) \in \Re^n$ and $I_{n-1}$ is the identity matrix in $\mathcal{S}^{n-1}$. It is easy to verify $QQ^T = Q^2 = I$.

**Lemma 3.3.** *Le $Y_1$ denote any optimal solution of (3.7) and let $Y_2$ denote the unique optimal solution of the following problem:*

$$\min \ \ f_3(Y) := \frac{1}{2}\|Y - D\|^2 \qquad s.t. \ \ Y \in \mathcal{K}^n_-. \tag{3.15}$$

*Then we must have $JY_1J = JY_2J$. In the view of Lemma 3.2, $Y_1$ and $Y_2$ determines a same EDM, which is $D^{cmds}$.*

**Proof.** Since $f_1(Y)$ is a semi-norm of $Y$, problem (3.7) may have multiple solutions (in fact, it has infinitely many solutions). But they all have common $JY_1J$, where $Y_1$ is just one of the optimal solution. Moreover

$$JY_1J = \Pi_{\mathcal{S}^n_-}(J\Delta J). \tag{3.16}$$

Problem (3.15) is the orthogonal projection onto the cone $\mathcal{K}^n_-$. Its optimal solution is given by

$$Y_2 = \Pi_{\mathcal{K}^n_-}(\Delta).$$

It follows from the projection formula (2.29) for $\Pi_{\mathcal{K}^n_-}(\cdot)$ that

$$Y_2 = \Delta - \Pi_{\mathcal{S}^n_+}(J\Delta J). \tag{3.17}$$

Using (3.14), we have

$$\begin{aligned} J\Delta J &= Q\begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q\Delta Q \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q \\ &= Q\begin{bmatrix} \Delta_1 & 0 \\ 0 & 0 \end{bmatrix} Q, \end{aligned}$$

where $\Delta_1$ is the leading $(n-1)\times(n-1)$ block of the matrix $(Q\Delta Q)$. Therefore, by using the facts $Q^2 = I$ and $Q \in \mathcal{S}^n$, we have

$$\begin{aligned} \Pi_{\mathcal{S}^n_+}(J\Delta J) &= Q\begin{bmatrix} \Pi_{\mathcal{S}^{n-1}_+}(\Delta_1) & 0 \\ 0 & 0 \end{bmatrix} Q \\ &= Q\begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} QQ \begin{bmatrix} \Pi_{\mathcal{S}^{n-1}_+}(\Delta_1) & 0 \\ 0 & 0 \end{bmatrix} QQ \begin{bmatrix} I_{n-1} & 0 \\ 0 & 0 \end{bmatrix} Q \\ &= JQ\begin{bmatrix} \Pi_{\mathcal{S}^{n-1}_+}(\Delta_1) & 0 \\ 0 & 0 \end{bmatrix} QJ = J\Pi_{\mathcal{S}^n_+}(J\Delta J)J. \end{aligned} \tag{3.18}$$

Therefore, we have from (3.17) that

$$\begin{aligned} JY_2J &= J\Delta J - J\Pi_{\mathcal{S}^n_+}(J\Delta J)J \\ &= J\Delta J - \Pi_{\mathcal{S}^n_+}(J\Delta J) && \text{(by (3.18))} \\ &= \Pi_{\mathcal{S}^n_-}(J\Delta J) && \text{(by (3.13))} \\ &= JY_1J. && \text{(by (3.16))} \end{aligned}$$

Since both $Y_1$ and $Y_2$ are in $\mathcal{K}^n_-$, Lemma 3.2 guarantees that they generate a same EDM matrix, which must be $D^{\texttt{mds}}$. □

Combination of all those technical results together leads to the following main result.

**Theorem 3.4.** *It holds that cMDS (3.7) determines the unique EDM $D^{\texttt{mds}}$, which is also completely determined by the joint optimization problem:*

$$\min_{D,Z} \ f_4(D,Z) := \frac{1}{2}\|\Delta - (D+Z)\|^2, \qquad \textit{s.t.} \ \ D \in \mathcal{D}^n \quad \textit{and} \quad Z \in \mathcal{S}^n_2. \tag{3.19}$$

**Proof.** We note that all optimal solutions of (3.7) share the same $JY_1J$, where $Y_1$ is any one of the optimal solutions and $Y_1 \in \mathcal{K}^n_-$. Lemma 3.2 implies that there exists a unique EDM, denoted as $D^{\texttt{mds}}$, that satisfies $JD^{\texttt{mds}}J = JY_1J$. Lemma 3.3 further implies that $D^{\texttt{mds}}$ can be obtained through the optimization problem (3.15), which by Lemma 3.1 is equivalent o the joint optimization problem (3.19). □

**Remark** (on a new connection between `cMDS` and `NEDM`). The joint optimization (3.19) clearly explains what `cMDS` is trying to achieve. Given a noisy distance matrix $\Delta$, `cMDS` tries to find the best EDM corrected by an optimal rank-2 matrix from the subspace $\mathcal{S}^n_2$, corresponding to $\mathbf{y} \in \Re^n$ (i.e., $\mathbf{y}$ is unconstrained). In contrast, if we restrict $\mathbf{y} = 0$, then the joint optimization (3.19) becomes the `NEDM` problem (3.8). In other words, `cMDS` (corresponding to $\mathbf{y} \in \mathcal{S}^n_2$) and `NEDM` (corresponding to $\mathbf{y} = 0$) are at the both ends of the spectrum of the joint optimization (3.19). The `L1MDS` proposed in this paper actually sits in between.

Our last result in this subsection is to explain why `cMDS` always yields a dense matrix $Z \in \mathcal{S}^n_2$. Let $(D^{\tt mds}, Z^{\tt mds})$ be the optimal solution of the joint optimization (3.19), which can be rewritten in terms of vector $\mathbf{y} \in \Re^n$:

$$\min_{D,\mathbf{y}} \ f_5(D, \mathbf{y}) := \frac{1}{2}\|\Delta - \left(D + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T\right)\|^2, \qquad \text{s.t.} \ \ D \in \mathcal{D}^n \ \ \text{and} \ \ \mathbf{y} \in \Re^n. \quad (3.20)$$

Define

$$S^{\tt mds} := \Delta - D^{\tt mds}, \qquad \boldsymbol{\delta} := \frac{1}{n} S^{\tt mds}\mathbf{e} \qquad \text{and} \qquad \bar{\delta} := \frac{1}{n}\mathbf{e}^T\boldsymbol{\delta} = \frac{1}{n^2}\mathbf{e}^T S^{\tt mds}\mathbf{e}.$$

If we see $S^{\tt mds}$ as the discrepancy matrix between $D^{\tt mds}$ and $\Delta$, the $i$th row of $S^{\tt mds}$ can be seen as the approximation error vector associated with the $i$th embedding point obtained by `cMDS`. We have the following result.

**Proposition 3.5.** *Let $(D^{\tt mds}, \mathbf{y}_{\tt mds})$ be the optimal solution of (3.20). Then we have*

$$\mathbf{y}_{\tt mds} = \boldsymbol{\delta} - \frac{1}{2}\bar{\delta}\mathbf{e} \qquad \text{and} \qquad Z^{\tt mds} = \left(\boldsymbol{\delta}\mathbf{e}^T + \mathbf{e}\boldsymbol{\delta}^T\right) - \bar{\delta}\mathbf{e}\mathbf{e}^T.$$

**Proof.** Obviously, $\mathbf{y}_{\tt mds}$ is the optimal solution of following unconstrained optimization:

$$\min f_5(D^{\tt mds}, \mathbf{y}) = \frac{1}{2}\|\Delta - (D^{\tt mds} + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T\|^2, \qquad \text{s.t.} \ \ \mathbf{y} \in \Re^n.$$

We note that

$$f_5(D^{\tt mds}, \mathbf{y}) = \frac{1}{2}\|S^{\tt mds}\|^2 - 2\langle S^{\tt mds}\mathbf{e}, \ \mathbf{y}\rangle + n\|\mathbf{y}\|^2 + \langle \mathbf{y}, \ \mathbf{e}\rangle^2.$$

Since $f_5(Y, \mathbf{y})$ is strongly convex in $\mathbf{y}$, the optimal $\mathbf{y}$ exists and satisfies

$$\frac{\partial}{\partial \mathbf{y}} f_5(D^{\tt mds}, \mathbf{y}) = 0,$$

which leads to (after some simplification)

$$n\mathbf{y} + \langle \mathbf{y}, \ \mathbf{e}\rangle\mathbf{e} = S^{\tt mds}\mathbf{e}. \qquad (3.21)$$

Computing the inner product with $\mathbf{e}$ on both sides of (3.21) yields

$$\langle \mathbf{y},\ \mathbf{e} \rangle = \frac{1}{2n} \mathbf{e}^T S^{\texttt{mds}} \mathbf{e}.$$

The optimal solution $\mathbf{y}$ follows (3.21):

$$\mathbf{y}_{\texttt{mds}} = \frac{1}{n} \left( S^{\texttt{mds}} \mathbf{e} - \frac{\mathbf{e}^T S^{\texttt{mds}} \mathbf{e}}{2n} \mathbf{e} \right) = \boldsymbol{\delta} - \frac{1}{2} \bar{\delta} \mathbf{e}.$$

This means that the optimal solution corresponding to the discrepancy matrix $S^{\texttt{mds}}$ is the average error vector $\boldsymbol{\delta}$ shifted by half of the overall error $\bar{\delta}$. Hence, the optimal $Z$ is given by

$$Z^{\texttt{mds}} = \mathbf{y}_{\texttt{mds}} \mathbf{e}^T + \mathbf{e} \mathbf{y}_{\texttt{mds}}^T = \left( \boldsymbol{\delta} \mathbf{e}^T + \mathbf{e} \boldsymbol{\delta}^T \right) - \bar{\delta} \mathbf{e} \mathbf{e}^T.$$

□

We note that $\mathbf{y}_{\texttt{mds}}$ is a dense vector (i.e., most of its components are non-zeros) unless some stringent conditions are enforced. This further implies that the optimal matrix $Z$ is dense. Now suppose $\Delta$ only contains few inaccurate measurements of a true distance matrix $D$. This means that the discrepancy matrix $S$ is sparse. However, when we start from the optimal solution $D$ and $\mathbf{y} = 0$ for the optimization problem (3.20) by a numerical method, the next step of the method would try to use a dense matrix $Z$ to approximate $S$, leading to large objective values. In order to decrease the objective function, the next iterates would move away from the optimal $D$, leading to `cMDS` solution. Practical experience shows that it is fine as long as the errors in the measurements are not too big. However, as seen from the motivating example in Introduction, `cMDS` would never work when the discrepancy matrix $S$ has a structural sparsity pattern and has large elements in it. This is because `cMDS` always results in a dense matrix $Z$ to approximate the discrepancy matrix, regardless it is sparse or not.

## 3.4 $\ell_1$ Regularization

In this part, we will address three more particular issues, which will lead to our ultimate model. In the view of the joint optimization (3.20) and motivated by the above discuss on sparsity, it is natural to enforce sparsity on $\mathbf{y}$ in order to get a sparse $Z$. Hence, $\ell_1$ regularization comes as a straightforward choice. However, there are three potential issues, the resolution of which will lead to an enhanced model. Those issues are double penalization in $\mathbf{y}$, non-separability of $D$ and $\mathbf{y}$ in the objective in (3.20), and including landmarks/anchors in the model.

**(a) Double penalization in $\mathbf{y}$.** On one hand, we note that the diagonal of the matrix $Z$ is $2\mathbf{y}$, which has been penalized through $\|\mathbf{y}\|_1$. On the other hand, we note that the diagonals of both $\Delta$ and $D \in \mathcal{D}^n$ are all zero. Hence, the quadratic term in the objective of (3.20) again penalizes $\mathbf{y}$ through $2\|\mathbf{y}\|^2$ (the diagonal part of the quadratic term). Hence, the vector $\mathbf{y}$ has been penalized twice. Therefore, it is more reasonable to remove the diagonal part of the quadratic term from the objective, which becomes:

$$f_6(D, \mathbf{y}) = \frac{1}{2} \sum_{i \neq j} \frac{1}{2} \Big( \Delta_{ij} - (D_{ij} + y_i + y_j) \Big)^2 + \mu \|\mathbf{y}\|_1.$$

We further introduce a symmetric matrix $E_0$, whose elements are all one except its diagonal being zero (i.e., $E_0 = \mathbf{e}\mathbf{e}^T - I$). Using the Hadamard product between matrices, $f(D, \mathbf{y})$ can be put in the following form:

$$f_6(D, \mathbf{y}) = \frac{1}{2} \|E_0 \circ (\Delta - (D + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T))\|^2 + \mu \|\mathbf{y}\|_1.$$

**(b) Non-separability.** The second issue is that the $y$ term is not separable in the objective function $f(D, \mathbf{y})$ in the sense that $\mathbf{y}$ is involved in both parts of the objective. This would create some difficulties in applying the popular alternating direction method of multipliers (ADMM) to $\ell_1$ minimization problem, see [14]. A relevant issue is that the orthogonal projection on the EDM cone $\mathcal{D}^n$ does not have a closed-form solution. The difficulties can be resolved by introducing a new variable:

$$Y := D + \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T, \quad D \in \mathcal{D}^n \ \text{ and } \ \mathbf{y} \in \Re^n.$$

According to Lemma 3.1, such $Y$ is equivalent to

$$Y \in \mathcal{K}^n_- \qquad \text{and} \qquad \mathbf{y} = \frac{1}{2}\mathrm{diag}(Y).$$

To summarize, our model should take the following form:

$$\begin{aligned} \min \quad & f(Y, \mathbf{y}) = \tfrac{1}{2}\|E_0 \circ (\Delta - Y)\|^2 + \mu\|\mathbf{y}\|_1 \\ \text{s.t.} \quad & Y \in \mathcal{K}^n_-, \quad \mathbf{y} = \tfrac{1}{2}\mathrm{diag}(Y). \end{aligned} \tag{3.22}$$

**(c) Including landmarks**. Suppose we have $m$ landmarks (known as anchors in sensor network localization), whose coordinates $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are known. Their pairwise Euclidean distance should be preserved in the model. This can be easily done in terms of $Y$ according to Lemma 3.1:

$$Y_{ij} - (y_i + y_j) = d_{ij}^2 := \|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad \text{for } i < j = 2, \ldots, m.$$

We put such constraints in a more general setting, which will facilitate the description of our ADMM algorithm later on. We use the index set $\mathcal{J}$ to record those known distances:

$$\mathcal{J} := \left\{(i,j) \mid i < j, \quad d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| \text{ is known}\right\}.$$

After those preparations, our final $\ell_1$ regularized MDS model that includes the landmark information is given by

$$\begin{aligned} \min \quad & f(Y, \mathbf{y}) = \tfrac{1}{2}\|E_0 \circ (\Delta - Y)\|^2 + \mu\|\mathbf{y}\|_1 \\ \text{s.t.} \quad & Y \in \mathcal{K}^n_-, \quad \tfrac{1}{2}\mathrm{diag}(Y) - \mathbf{y} = 0, \\ & Y_{ij} - (y_i + y_j) = d_{ij}^2, \qquad (i,j) \in \mathcal{J}. \end{aligned} \tag{3.23}$$

We now put (3.23) in a more compact form. Define the linear operator $\mathcal{A}_1 : \mathcal{S}^n \mapsto \Re^n$ and $\mathcal{B}_1 : \Re^n \mapsto \Re^n$ respectively by

$$\mathcal{A}_1(Y) := \frac{1}{2}\mathrm{diag}(Y), \qquad \mathcal{B}_1(\mathbf{y}) := \mathbf{y}, \qquad \text{for } Y \in \mathcal{S}^n \text{ and } \mathbf{y} \in \Re^n.$$

Let $p := |\mathcal{J}|$ (the cardinality of $\mathcal{J}$). We will label a vector in $\Re^k$ by $(i,j) \in \mathcal{J}$. For example, define the vector $\mathbf{c} \in \Re^p$ by

$$c_{ij} := d_{ij}^2, \quad (i,j) \in \mathcal{J}.$$

We further define two linear operators $\mathcal{A}_2 : \mathcal{S}^n \mapsto \Re^p$ and $\mathcal{B}_2 : \Re^n \mapsto \Re^p$ respectively by

$$\Big(\mathcal{A}_2(Y)\Big)_{ij} := Y_{ij} \quad \text{and} \quad \Big(\mathcal{B}_2(\mathbf{y})\Big)_{ij} := y_i + y_j \ \text{ for } (i,j) \in \mathcal{J}.$$

We finally define $\mathcal{A} : \mathcal{S}^n \mapsto \Re^{n+p}$ and $\mathcal{B} : \Re^n \mapsto \Re^{n+p}$ by

$$\mathcal{A}(Y) := \begin{pmatrix} \mathcal{A}_1(Y) \\ \mathcal{A}_2(Y) \end{pmatrix}, \quad \mathcal{B}(\mathbf{y}) := \begin{pmatrix} \mathcal{B}_1(\mathbf{y}) \\ \mathcal{B}_2(\mathbf{y}) \end{pmatrix}, \quad \text{and} \quad \mathbf{b} := \begin{pmatrix} \mathbf{0}_n \\ \mathbf{c} \end{pmatrix}.$$

The model (3.23) becomes

$$\begin{aligned} \min \quad & f(Y,\mathbf{y}) = \tfrac{1}{2}\|E_0 \circ (\Delta - Y)\|^2 + \mu\|\mathbf{y}\|_1 \\ \text{s.t.} \quad & \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) = \mathbf{b} \\ & Y \in \mathcal{K}_-^n. \end{aligned} \tag{3.24}$$

We note that although $f(Y,\mathbf{y})$ is not strongly convex, its level set must be bounded and closed. Therefore, problem (3.24) must have an optimal solution. In the next subsection, we propose a PADMM to solve it.

## 3.5 A Proximal Alternating Direction Method of Multipliers

Problem (3.24) has a 2-block structure involving two variables $Y$ and $\mathbf{y}$, which are separable. This type of problems can often be solved by alternating direction method of multipliers with proximal terms [14]. We describe such a method below. Let $c > 0$ be a given constant and $\mathbf{z} \in \Re^{n+k}$ be the Lagrange multipliers corresponding to the equality constraints in (3.24). Define the augmented Lagrange function as

$$L_c(Y, \mathbf{y}; \mathbf{z}) \quad := \quad f(Y, \mathbf{y}) - \langle \mathbf{z},\ \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b} \rangle + \frac{c}{2} \| \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b} \|^2.$$

Suppose we have two positive semidefinite operators $\mathcal{P} : \mathcal{S}^n \mapsto \mathcal{S}^n$ and $\mathcal{Q} : \Re^n \mapsto \Re^n$. Define the associated norms by:

$$\|X\|^2_{\mathcal{P}} := \langle X,\ \mathcal{P}X \rangle, \qquad \|X\|^2_{\mathcal{Q}} := \langle X,\ \mathcal{Q}X \rangle.$$

Given an initial point $(Y^0, \mathbf{y}^0; \mathbf{z}^0)$ and $k := 0$, the proximal ADMM generates the following update from $(Y^k, \mathbf{y}^k; \mathbf{z}^k)$

$$\begin{cases} Y^{k+1} & := & \arg\min_{Y \in \mathcal{K}^n_-} \ L_c(Y, \mathbf{y}^k; \mathbf{z}^k) + \frac{1}{2} \|Y - Y^k\|^2_{\mathcal{P}} \\ \mathbf{y}^{k+1} & := & \arg\min_{\mathbf{y} \in \Re^n} \ L_c(Y^{k+1}, \mathbf{y}; \mathbf{z}^k) + \frac{1}{2} \|\mathbf{y} - \mathbf{y}^k\|^2_{\mathcal{Q}} \\ \mathbf{z}^{k+1} & := & \mathbf{z}^k - \tau c \left( \mathcal{A}(Y^{k+1}) - \mathcal{B}(\mathbf{y}^{k+1}) - \mathbf{b} \right), \end{cases} \tag{3.25}$$

where $\tau \in (0, (1 + \sqrt{5})/2)$ is known as the step-length.

The choice of $\mathcal{P}$ and $\mathcal{Q}$ is crucial to the success of the method (3.25) and is governed by three criteria. One is to ensure the subproblems in (3.25) easy to obtain (e.g., they admit closed-form solutions). The second is to ensure the convergence of the method. And the third is that they should be as "small" as possible in order for the proximal solutions to stay close to that of the augmented Lagrangian function. We show that we have nice choices.

### 3.5.1 Choice of $\mathcal{P}$ and computation of $Y^{k+1}$

The choice and the computation are related to the undirected graph $\mathcal{G} := (V, \mathcal{E})$, where $V$ consists of $n$ vertices $\{1, 2, \ldots, \ldots, n\}$ and the edge set $\mathcal{E}$ consists of the edges $(i, j)$ whenever $(i, j) \in \mathcal{J}$. Let $A \in \mathcal{S}^n$ be the adjacency matrix of $\mathcal{G}$:

$$A_{ij} := \begin{cases} 1 & \text{if } (i,j) \text{ or } (j,i) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

We will also need to compute $\mathcal{A}$ and its adjoint $\mathcal{A}^*$. We detail them below.

(i) Computation of $\mathcal{A}_1$ and its adjoint $\mathcal{A}_1^*$. We have

$$\mathcal{A}_1(Y) = \frac{1}{2}\text{diag}(Y) \qquad \text{and} \qquad \mathcal{A}_1^*(\mathbf{y}) = \frac{1}{2}\text{Diag}(\mathbf{y}).$$

Hence, we have

$$\mathcal{A}_1^*\mathcal{A}_1(Y) = \frac{1}{4}\Big(I \circ Y\Big) \qquad \text{for } Y \in \mathcal{S}^n.$$

(ii) Computation of $A_2$ and its adjoint $\mathcal{A}_2^*$. We have

$$\Big(\mathcal{A}_2(Y)\Big)_{ij} = Y_{ij} = \frac{1}{2}\langle \mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T, Y\rangle$$

and for $\mathbf{t} \in \Re^p$,

$$\mathcal{A}_2^*(\mathbf{t}) = \frac{1}{2}\sum_{(i,j)\in\mathcal{J}} t_{ij}(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T).$$

Hence,

$$\mathcal{A}_2^*\mathcal{A}_2(Y) = \frac{1}{2}\sum_{(i,j)\in\mathcal{J}} Y_{ij}(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T) = \frac{1}{2}\Big(A \circ Y\Big).$$

(iii) Computation of $\mathcal{A}^*\mathcal{A}$. We have

$$\mathcal{A}^*\mathcal{A}(Y) = \mathcal{A}_1^*\mathcal{A}_1(Y) + \mathcal{A}_2^*\mathcal{A}_2(Y) = \frac{1}{4}(I + 2A) \circ Y.$$

(iv) The choice of $\mathcal{P} : \mathcal{S}^n \mapsto \mathcal{S}^n$ is given by

$$\mathcal{P} := \left(\frac{4-c}{4}I + \frac{c}{2}(\mathbf{e}\mathbf{e}^T - A)\right) \circ .$$

It is straightforward to versify that $\mathcal{P}$ is a positive semidefinite operator from $\mathcal{S}^n$ to $\mathcal{S}^n$.

The computation of $Y^{k+1}$ is summarized in the following result.

**Proposition 3.6.** *The optimal solution $Y^{k+1}$ in (3.25) is given by*

$$Y^{k+1} = \frac{2}{2+c}\Pi_{\mathcal{K}_-^n}(\Delta^k), \tag{3.26}$$

*where*

$$\Delta^k := \Delta + \mathcal{P}(Y^k) + \mathcal{A}^*\Big(\mathbf{z}^k + c(\mathcal{B}(\mathbf{y}^k) + b)\Big).$$

**Proof.** The formula for $Y^{k+1}$ is derived below, where we omitted constant terms and some simplifications, and used the fact that $\mathrm{diag}(\Delta) = 0$.

$$\begin{aligned}
Y^{k+1} &= \arg\min_{Y \in \mathcal{K}^n_-} L_c(Y, \mathbf{y}^k; \mathbf{z}^k) + \frac{1}{2}\|Y - Y^k\|^2_{\mathcal{P}} \\
&= \arg\min_{Y \in \mathcal{K}^n_-} \Big\{ \frac{1}{2}\|E_0 \circ (Y - \Delta)\|^2 - \langle \mathcal{A}^*(\mathbf{z}^k),\ Y\rangle + \frac{c}{2}\|\mathcal{A}(Y)\|^2 \\
&\qquad\qquad -c\langle Y,\ \mathcal{A}^*(\mathcal{B}(\mathbf{y}^k) + \mathbf{b})\rangle + \frac{1}{2}\|Y - Y^k\|^2_{\mathcal{P}} \Big\} \\
&= \arg\min_{Y \in \mathcal{K}^n_-} \Big\{ \frac{1}{2}\|E_0 \circ Y\|^2 - \langle Y,\ \Delta\rangle + \frac{c}{8}\langle Y,\ (I + 2A) \circ Y\rangle \\
&\qquad\qquad + \frac{1}{2}\langle Y,\ \mathcal{P}(Y)\rangle - \langle Y,\ +\mathcal{P}(Y^k) + \mathcal{A}^*(\mathbf{z}^k + c(\mathcal{B}(\mathbf{y}^k) + \mathbf{b}))\rangle \Big\} \\
&= \arg\min_{Y \in \mathcal{K}^n_-} \frac{2+c}{4}\|Y\|^2 - \langle Y,\ \underbrace{\Delta + \mathcal{P}(Y^k) + \mathcal{A}^*(\mathbf{z}^k + c(\mathcal{B}(\mathbf{y}^k) + \mathbf{b}}_{=:\Delta^k}))\rangle \\
&= \arg\min_{Y \in \mathcal{K}^n_-} \frac{2+c}{4}\|Y - 2\Delta^k/(2+c)\|^2 \\
&= \frac{2}{2+c}\Pi_{\mathcal{K}^n_-}(\Delta^k).
\end{aligned}$$

### 3.5.2 Choice of $\mathcal{Q}$ and computation of $\mathbf{y}^{k+1}$

The choice of $\mathcal{Q}$ is related to the Laplacian of the graph $\mathcal{G}$ and to the formula of $\mathcal{B}^*\mathcal{B}$. We compute them below.

(i) Computation of $\mathcal{B}_2^*$ and $\mathcal{B}_2^*\mathcal{B}_2$. Recall that a vector $\mathbf{t} \in \Re^p$ is labelled by the indices $(i, j) \in \mathcal{J}$. For such vector $\mathbf{t}$, let us define its symmetric matrix, denoted by $\texttt{sym}(\mathbf{t}) \in \mathcal{S}^n$ given by

$$\Big(\texttt{sym}(\mathbf{t})\Big)_{ij} = \begin{cases} t_{ij} & \text{if } (i, j) \text{ or } (j, i) \in \mathcal{J} \\ 0 & \text{otherwise.} \end{cases}$$

We then have for any $\mathbf{t} \in \Re^p$ and $\mathbf{y} \in \Re^n$,

$$\begin{aligned}\langle \mathcal{B}_2^*(\mathbf{t}),\ \mathbf{y}\rangle &= \langle \mathbf{t},\ \mathcal{B}_2(\mathbf{y})\rangle = \sum_{(i,j)\in\mathcal{J}} t_{ij}(y_i + y_j) \\ &= \sum_{(i,j)\in\mathcal{J}} t_{ij}y_i + \sum_{(i,j)\in\mathcal{J}} t_{ij}y_j \\ &= \langle \mathtt{sym}(\mathbf{t})\mathbf{e},\ \mathbf{y}\rangle.\end{aligned}$$

Hence

$$\mathcal{B}_2^*(\mathbf{t}) = \mathtt{sym}(\mathbf{t})\mathbf{e}.$$

Moreover,

$$\langle \mathbf{y},\ \mathcal{B}_2^*\mathcal{B}_2(\mathbf{y})\rangle = \langle \mathcal{B}_2(\mathbf{y}),\ \mathcal{B}_2(\mathbf{y})\rangle = \sum_{(i,j)\in\mathcal{J}} (y_i + y_j)^2 = \mathbf{y}^T\left(\mathrm{Diag}(\mathbf{d}) + A\right)\mathbf{y},$$

where $\mathbf{d} \in \Re^n$ is the degree vector of the graph $G$ satisfying $\mathbf{d} = A\mathbf{e}$ (the column sum of the adjacency matrix). Therefore,

$$\mathcal{B}_2^*\mathcal{B}_2 = \mathrm{Diag}(\mathbf{d}) + A,$$

(ii) Choice of $\mathcal{Q}$ is given by

$$\mathcal{Q} := cL \qquad \text{with} \ \ L := \mathrm{Diag}(\mathbf{d}) - A.$$

It is known that $L$ is the Laplacian matrix of graph $\mathcal{G}$ and hence is positive semidefinite [48].

(iii) The weight vector $\boldsymbol{\beta} \in \Re^n$ to be used in computing $\mathbf{y}^{k+1}$ is defined by

$$\boldsymbol{\beta} := \sqrt{\mathbf{e} + 2\mathbf{d}},$$

where the square root is taken componentwise. Similarly, $\boldsymbol{\beta}^{-1}$ is the inverse of $\boldsymbol{\beta}$ componentwise.

The computation of $\mathbf{y}^{k+1}$ is given by a weighted shrinkage (soft thresholding in compressed sensing) operator and is summarized in the following result.

**Proposition 3.7.** *The optimal solution* $\mathbf{y}^{k+1}$ *in (3.25) is given by*

$$\mathbf{y}^{k+1} = \mathit{Shrink}_{\boldsymbol{\beta}}(\widehat{\mathbf{s}}^k,\ \frac{\mu}{c}) = \frac{1}{c}(\boldsymbol{\beta} \circ \boldsymbol{\beta})^{-1} \circ \max\Big\{|\mathbf{s}^k| - \mu,\ 0\Big\} \circ \mathit{sign}(\mathbf{s}^k).$$

*where*

$$\widehat{\mathbf{s}}^k := \frac{1}{c}\boldsymbol{\beta}^{-1} \circ \mathbf{s}^k \quad \mathit{and} \quad \mathbf{s}^k := \mathcal{B}^*\Big(c(\mathcal{A}(Y^{k+1}) - \mathbf{b}) - \mathbf{z}^k\Big) + \mathcal{Q}(\mathbf{y}^k).$$

**Proof.** We first note that

$$\mathcal{B}^*\mathcal{B} = \mathcal{B}_1^*\mathcal{B}_1 + \mathcal{B}_2^*\mathcal{B}_2 = I + (\mathrm{Diag}(\mathbf{d}) + A).$$

The choice of $\mathcal{Q}$ allows us to cancel some terms in computing $\mathbf{y}^{k+1}$ in (3.25). We outline the computation below by omitting the terms that are not relevant to $\mathbf{y}$.

$$\begin{aligned}
\mathbf{y}^{k+1} &= \arg\min L_c(Y^{k+1}, \mathbf{y}; \mathbf{z}^k) + \frac{1}{2}\|\mathbf{y} - \mathbf{y}^k\|_{\mathcal{Q}}^2 \\
&= \arg\min \mu\|\mathbf{y}\|_1 + \langle \mathbf{z}^k,\ \mathcal{B}(\mathbf{y})\rangle + \frac{c}{2}\|\mathcal{B}(\mathbf{y})\|^2 - c\langle \mathcal{B}(\mathbf{y}),\ \mathcal{A}(Y^{k+1}) - \mathbf{b}\rangle + \frac{1}{2}\|\mathbf{y} - \mathbf{y}^k\|_{\mathcal{Q}}^2 \\
&= \arg\min \mu\|\mathbf{y}\|_1 - \langle \mathcal{B}^*(c(\mathcal{A}(Y^{k+1}) - \mathbf{b}) - \mathbf{z}^k),\ \mathbf{y}\rangle + \frac{c}{2}\|\mathbf{y}\|^2 \\
&\quad + \frac{c}{2}\mathbf{y}^T(\mathrm{Diag}(\mathbf{d}) + A)\mathbf{y} + \frac{c}{2}(\mathbf{y} - \mathbf{y}^k)^T L(\mathbf{y} - \mathbf{y}^k) \\
&= \arg\min\ \mu\|\mathbf{y}\|_1 - \langle \mathbf{y},\ \underbrace{Q\mathbf{y}^k + \mathcal{B}^*(c(\mathcal{A}(Y^{k+1}) - \mathbf{b}) - \mathbf{z}^k}_{=:\mathbf{s}^k}\rangle + c\mathbf{y}^T\mathrm{Diag}(\mathbf{d})\mathbf{y} + \frac{c}{2}\|\mathbf{y}\|^2 \\
&= \arg\min \frac{c}{2}\mathbf{y}^T\Big(I + 2\mathrm{Diag}(\mathbf{d})\Big)\mathbf{y} - \langle \mathbf{y},\ \mathbf{s}^k\rangle + \mu\|\mathbf{y}\|_1 \\
&= \arg\min \frac{c}{2}\|\boldsymbol{\beta} \circ \mathbf{y}\|^2 - \langle \mathbf{y},\ \mathbf{s}^k\rangle + \mu\|\mathbf{y}\|_1 \\
&= \arg\min \frac{c}{2}\left\|\boldsymbol{\beta} \circ \mathbf{y} - \frac{1}{c}\boldsymbol{\beta}^{-1} \circ \mathbf{s}^k\right\|^2 + \mu\|\mathbf{y}\|_1 \\
&= \mathrm{Shrink}_{\boldsymbol{\beta}}(\widehat{\mathbf{s}}^k, \mu/c).
\end{aligned}$$

Applying the formula (3.5), we get the computational formula for $\mathbf{y}^{k+1}$ as stated in the result. □

With such choice $\mathcal{P}$ and $\mathcal{Q}$ in the algorithm (3.25) and with $Y^{k+1}$ and $\mathbf{y}^{k+1}$ being computed respectively as above, the sequence $\{Y^k, \mathbf{y}^k\}$ generated by the PADMM (3.25) converges to an optimal solution of (3.22). This is ensured by the general convergence theorem of Fazel et al. [34, Theorem B.1(c)] for any steplength choice $\tau \in (0, (1+\sqrt{5})/2)$.

## 3.6 Numerical Results

In this part, we will first address the stopping criterion for terminating Algorithm (3.25). We then report our preliminary results on a small set of test problems. We will tackle more complicated test problems in future research.

### 3.6.1 Stopping criterion

We will use the Karush-Kuhn-Tucker (KKT) condition of problem (3.24) as our stopping criterion. We follow a standard procedure to derive the KKT condition. We note that the problem (3.24) is equivalent to the following problem:

$$\begin{aligned}
&\min\nolimits_{Y,\mathbf{y}} \quad \Big\{ \max \Big( f(Y,\mathbf{y}) - \langle \mathbf{z}, \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b} \rangle - \langle Z, Y \rangle \mid Z \in \mathcal{K}^n_-, \quad \mathbf{z} \in \Re^q \Big) \Big\} \\
&= \min\nolimits_{Y,\mathbf{y}} \begin{cases} f(Y,\mathbf{y}) & \text{if } \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b} = 0 \\ & \quad Y \in \mathcal{K}^n_- \text{ and } \langle Z, Y \rangle = 0 \\ +\infty & \text{otherwise} \end{cases}
\end{aligned} \tag{3.27}$$

The Lagrange dual problem is (swap the max and min in (3.27)):

$$\max_{\substack{Z \in \mathcal{K}^n_- \\ \mathbf{z} \in \Re^q}} \left\{ \underbrace{\min_{Y,\mathbf{y}} \; f(Y,\mathbf{y}) - \langle \mathbf{z}, \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b} \rangle - \langle Z, Y \rangle}_{\text{inside optimization}} \right\}$$

The optimality condition of the inside optimization (together with the condition in (3.27)) consists of the KKT condition:

$$((\mathrm{KKT})\quad \left\{ \begin{array}{rcl} E_0 \circ (Y - \Delta) - \mathcal{A}^*(\mathbf{z}) - Z & = & 0 \\ \mu \partial \|\mathbf{y}\|_1 + \mathcal{B}^*(\mathbf{z}) & \ni & 0 \\ \mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b} & = & 0 \\ Y \in \mathcal{K}^n_-, \ Z \in \mathcal{K}^n_-, \ \langle Z, \ Y \rangle & = & 0, \end{array} \right. \tag{3.28}$$

where $\partial\|\mathbf{y}\|_1$ is the subdifferential of the $\ell_1$ norm.

We will measure the violation of the KKT condition at the computed point $(Y^{k+1}, \mathbf{y}^{k+1}, \mathbf{z}^{k+1})$. The corresponding $Z^{k+1}$ is computed via the first equation in the KKT condition (3.28):

$$Z^{k+1} = E_0 \circ (Y^k - \Delta) - \mathcal{A}^*(\mathbf{z}^k).$$

Hence, the first equation in (3.28) is satisfied. The second condition can be quantified as follows. It follows from the optimization of $\mathbf{y}^{k+1}$ in (3.25), we have

$$0 \in \mathcal{Q}(\mathbf{y}^{k+1} - \mathbf{y}^k) + \mathcal{B}^*(\mathbf{z}^k) + \mu\partial\|\mathbf{y}^{k+1}\|_1.$$

Hence,

$$-\mathcal{Q}(\mathbf{y}^{k+1} - \mathbf{y}^k) \in \mathcal{B}^*(\mathbf{z}^k) + \mu\partial\|\mathbf{y}^{k+1}\|_1.$$

Therefore,

$$-\mathcal{Q}(\mathbf{y}^{k+1} - \mathbf{y}^k) + \mathcal{B}^*(\mathbf{z}^{k+1} - \mathbf{z}^k) \in \mathcal{B}^*(\mathbf{z}^{k+1}) + \mu\partial\|\mathbf{y}^{k+1}\|_1.$$

Hence, the second condition in (3.28) at $(\mathbf{y}^{k+1}, \mathbf{z}^{k+1})$ can be measured by

$$\eta_1^k := \|\mathcal{Q}(\mathbf{y}^{k+1} - \mathbf{y}^k) - \mathcal{B}^*(\mathbf{z}^{k+1} - \mathbf{z}^k)\|.$$

We define

$$\eta_2(Y, \mathbf{y}) := \|\mathcal{A}(Y) - \mathcal{B}(\mathbf{y}) - \mathbf{b}\|.$$

Then $\eta_2^k := \eta_2(Y^{k+1}, \mathbf{y}^{k+1})$ measures the violation of the third equation in (3.28). The condition $Y \in \mathcal{K}_-^n$ is automatically satisfied at $Y^{k+1}$ because of the formula (3.26). Define

$$\eta_3(Z) := \|Z - \Pi_{\mathcal{K}_-^n}(Z)\| \qquad \text{and} \qquad \eta_4(Y, Z) := |\langle Z, Y\rangle|$$

Then $\eta_3^k := \eta_3(Z^{k+1})$ measures the violation of $Z \in \mathcal{K}_-^n$ at $Z^{k+1}$ and $\eta_4^k := \eta_4(Y^{k+1}, Z^{k+1})$ measures the complementary condition $\langle Z, Y\rangle = 0$ at $(Y^{k+1}, Z^{k+1})$. We further define

$$\eta^k := \max\{\eta_1^k,\ \eta_2^k,\ \eta_3^k,\ \eta_4^k\}.$$

We terminate Algorithm (3.25) whenever

$$\eta^k \le \texttt{tol},$$

where `tol` is a given tolerance (e.g., $10^{-3}$).

The other two important parameters are set by

$$c = 1.618 \approx (1+\sqrt{5})/2 \qquad \text{and} \qquad \mu = 1.$$

### 3.6.2 Test problems

This part only contains some preliminary numerical experiments. More complex problem remains to be investigated. We first test the triangle problem studied in Subsection 3.1.2. We found that the sparse vector is

$$\mathbf{y}^T = -(0.5677, 0, 0, 0)$$

and the corresponding sparse matrix $Z$ is

$$Z = -\begin{pmatrix} 0 & 0.4677 & 0.4677 & 0.4677 \\ 0.4677 & 0 & 0 & 0 \\ 0.4677 & 0 & 0 & 0 \\ 0.4677 & 0 & 0 & 0 \end{pmatrix},$$

which approximates the original sparse discrepancy matrix $S$ well

$$S = -\begin{pmatrix} 0 & 0.55 & 0.5 & 0.43 \\ 0.55 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \\ 0.43 & 0 & 0 & 0 \end{pmatrix}.$$

And the recovered figure by our method is Fig. 3.1(d).

Our second test problem is generated as follows. We uniformly randomly generate $n_0$ points $\{\mathbf{x}_i\}$ over the square $[-2,2]\times[-2,2]$ with additional $m$ points being the landmarks. We let $m = 4$ and they are

$$\begin{bmatrix} -1.5 \\ 1.5 \end{bmatrix}, \quad \begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \quad \begin{bmatrix} 1.5 \\ -1.5 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}.$$

The distances among the generated points have the following additive noises:

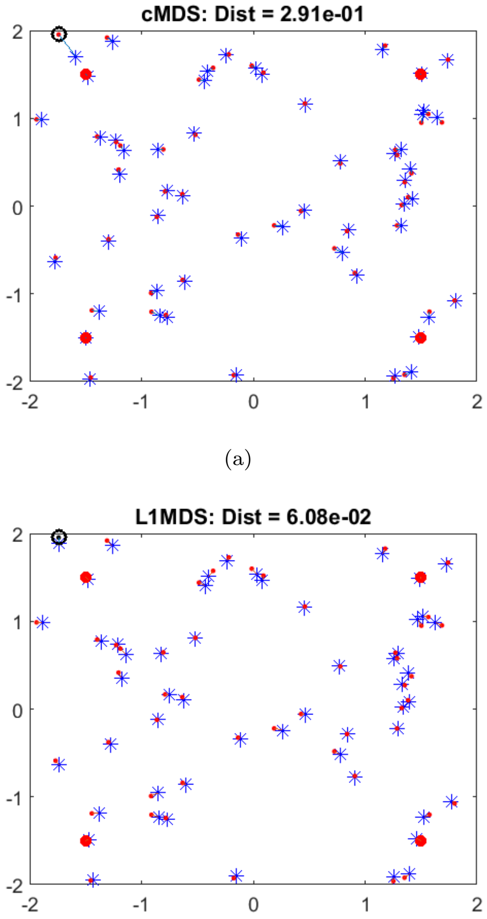
$$\Delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| + \mathtt{nf} \times \epsilon_{ij},$$

where $\mathtt{nf}$ is the noise factor (we let $\mathtt{nf} = 0.2$) and $\epsilon_{ij}$ has the normal distribution with mean 0 and variance $\sigma^2 = 1$.

We randomly pick one point as a faulty node and its distances to other points contain large errors (randomly generated from the interval $[-4,4]$). A typical recovery is illustrated in Fig. 3.2, where cMDS is also tested. Our L1MDS is able to locate the faulty node, while there is no way for cMDS to know which node is faulty. The recovery error ($6.08 \times 10^{-2}$) from L1MDS is also much smaller compared to $2.91 \times 10^{-1}$ by cMDS.

## 3.7 Conclusion

We summarize what we have achieved in this chapter.



(a)

(b)

Figure 3.2: 50 random placed sensors (in blue) and 4 fixed anchors (in red). There is one faulty sensor (circled), which is correctly identified by `L1MDS` and is also accurately allocated. Distances used all have noises with the faulty sensor having large distance errors.

(i) We gave a new interpretation about `cMDS`, which tends to use a dense matrix to approximate sparse outliers, see Fig. 3.1 and Prop.3.5.

(ii) We proposed a sparse model that restrict the sparse matrix $Z$ to the following set

$$\mathcal{S}_2^n = \{Y \mid Y = \mathbf{e}\mathbf{y}^T + \mathbf{y}\mathbf{e}^T\}$$

with $\mathbf{y}$ being a sparse vector. We then used the $\ell_1$ norm on $\mathbf{y}$ to promote the sparsity in $\mathbf{y}$, see the model (3.3).

(iii) Finally, we proposed an alternating direction methods of multipliers with proximal terms (PADMM) to solve the model, which is capable of including more constraints such as those on landmarks.

(iv) Preliminary numerical experiments were conducted on small problems. It seems that PADMM works fine with small problems due to it large number of iterations. More complex problems remain to be investigated.

(v) In the next chapter, we continue to consider the sparse model, but with $\mathcal{S}_2^n$ being replaced by a full space $\mathcal{S}^n$ and the following lower and upper bounds are included:

$$\ell_{ij} \le \|\mathbf{x}_i - \mathbf{x}_j\| \le u_{ij}.$$

Such constraints often appear in practice and are hard to deal with.

# Chapter 4

# A Subspace Model and Its Full Space Variant

## 4.1 Introduction

The starting point of this chapter is [Theorem 3.4, Chapter 3] and the resulting proximal ADMM [Section 3.5, Chapter 3], where it seems that the proximal ADMM can only solve less challenging problems. In this chapter, we develop two more practical models based on [Theorem 3.4, Chapter 3]. They are Subspace Sparse MDS model and Full Space Sparse MDS model. We will develop a fast algorithm for each of the model, study their convergence and conduct numerical test on some of very challenging problems.

Suppose there are $n$ items and their pairwise Euclidean distances $d_{ij}$ can be measured through the pairwise dissimilarities $\delta_{ij}$, 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

$$d_{ij}^2 := \|\mathbf{y}_i - \mathbf{y}_j\|^2 \approx \ \delta_{ij}^2, \quad i, j = 1, \ldots, n, \tag{4.1}$$

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).

This chapter consider a more general model than (3.1) considered in the previous chapter:

$$\begin{array}{ll} \min_{D,Z} & \|\Delta - (D+Z)\|^2 \\ \text{s.t.} & D \in \mathcal{D}^n, \quad \text{the embedding dimension of } D \leq r, \\ & Z \in V, \end{array} \tag{4.2}$$

where $r$ is a given embedding dimension and $V$ is the subspace $\mathcal{S}^n_2$ or the full space $\mathcal{S}^n$. We refer the corresponding models respectively as the subspace model and the full space model. Due to the embedding constraint, both models are nonconvex.

In this chapter, 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 widely observed phenomenon [18]. The reason is that `cMDS` always subtracts a dense matrix from the squared dissimilarity matrix $\overline{\Delta} := (\delta^2_{ij})$ before computing a set of embedding points (see Thm. 4.1). This result reveals the true mechanism behind the popular computational formula of `cMDS` [26, 12]. This detour to the desired purpose in (4.1) does not work because `cMDS` would punish every $\delta_{ij}$ 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 [5, 74, 93]. 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 $\ell_1$-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. 4.2.

(ii) We are able to control the sparsity level in every step of our calculation, thanks to the $\ell_1$-based regularization coupled with the objective function of `cMDS`, see Thm. 4.3. This is in contrast to the $\ell_1$-regularized methods in [35, 59, 60] 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.

## 4.2 cMDS and Noise Spreading

We describe how `cMDS` computes a set of embedding points $\mathbf{y}_i$ trying to satisfy the approximation in (4.1) 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 $\mathcal{S}^n_+$:

$$B := -\frac{1}{2} J \overline{\Delta} J, \qquad B_+ := \Pi_{\mathcal{S}^n_+}(B) \tag{4.3}$$

Note that $J$ is the centering matrix. The double-centering in $B$ was introduced to `cMDS` by Torgerson [91]. It further decomposes $B_+$ as a Gram matrix

$$B_+ = Y^T Y \quad \text{with} \quad Y := [\mathbf{y}_1, \ldots, \mathbf{y}_n] \tag{4.4}$$

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

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

Due to its simplicity, low-computational complexity and its mathematical interpretation via PCA, `cMDS` has become a popular method [12].

The main drawback that `cMDS` suffers is its noise spreading, which was highlighted in [18]. For example, if there is just one $\delta_{ij}$ containing noise $\epsilon$ and a measurement error $\eta$ (i.e., $\delta_{ij} = d_{ij} + \epsilon + \eta$) (all other $\delta_{ij}$ are true Euclidean distances), the double-centering operation in $B$ (4.3) spreads the error ($\epsilon + \eta$ to every entry. This would result in poor approximation, particularly when $\eta$ is caused by an outlier ($\eta$ 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 $D^{\texttt{mds}}$ is the solution of the optimization problem [61]:

$$D^{\texttt{mds}} = \arg\min \|J(D - \overline{\Delta})J\|, \quad \text{s.t.} \quad D \in \mathcal{D}^n, \tag{4.5}$$

We can obtain the matrix $B_+$ by

$$B_+ = -\frac{1}{2} J D^{\texttt{mds}} J \qquad (\text{also} \quad r = \text{rank}(J D^{\texttt{mds}} J). \tag{4.6}$$

Decomposing $B_+$ as in (4.4) to get the embedding points $\mathbf{y}_i$. As done in [63], if we define the semi-norm $\|A\|_J := \|JAJ\|$, then

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

However, a semi-norm is not a true norm. Therefore, a more natural matrix nearness problem is the so-called the nearest EDM problem (under the true norm $\|\cdot\|$):

$$D^{\texttt{edm}} = \arg\min \|D - \overline{\Delta}\|^2, \quad \text{s.t.} \quad D \in \mathcal{D}^n. \tag{4.7}$$

We refer to [37, 40, 70] for more reading on this problem and its applications. We will see that the problems (4.5) and (4.7) sit at the each end of a class of optimization problems over a subspace.

## 4.3 Revisiting [Theorem 3.4, Chapter 3]

We recall from [Theorem 3.4, Chapter 3] the following fact:

**Theorem 4.1.** *It holds that cMDS [3.7, Chapter 3] determines the unique EDM $D^{\texttt{mds}}$, which is also completely determined by the joint optimization problem:*

$$\min_{D,Z}\ f_4(D,Z) := \frac{1}{2}\|\Delta - (D+Z)\|^2, \qquad \textit{s.t.}\ \ D \in \mathcal{D}^n \quad \textit{and} \quad Z \in \mathcal{S}_2^n.$$

We further showed in [Prop. 3.5, Chapter 3] that `cMDS` is incapable of noise removal and outlier detection: It always to use a dense matrix trying to remove sparse noises. It then developed in [Sect. 3.4, Chapter 3] an $\ell_1$ regularized problem to achieve the purpose. Below we develop two more practical reformulations based on the $\ell_1$ regularization.

An elementary formulation is described below:

$$\begin{array}{ll} \min_{D,\mathbf{z}} & \frac{1}{2}\|(D + \mathbf{1}\mathbf{z}^T + \mathbf{z}\mathbf{1}^T) - \overline{\Delta}\|^2 + \mu \mathcal{R}_1(\mathbf{z}) \\ \text{s.t.} & D \in \mathcal{D}^n, \ \mathbf{z} \in \Re^n, \end{array} \tag{4.8}$$

where $\mu > 0$ is a parameter controlling the sparsity in $\mathbf{z}$. A particular choice is the $\ell_1$ regularization: $\mathcal{R}_1(z) := \|\mathbf{z}\|_1$. If $\mu = 0$, then (4.8) becomes `cMDS` (3.19), and if $\mu = +\infty$, we have $\mathbf{z} = 0$ and (4.8) becomes the EDM problem (4.7). Therefore, `cMDS` and EDM (4.7) stand at the two extremes of (4.8) with `cMDS` tending to over-denoise and EDM (4.7) 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 in (4.2), which is equivalent to $\text{rank}(JDJ) \le r$. It follows from (2.35) 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 $\delta_{ij}$. A common practice is to apply positive weights on available $\delta_{ij}$ and 0 weights on missing $\delta_{ij}$. For example, a weight matrix $W \in \mathcal{S}^n$ can be defined as follows: $W_{ij} = 1$ for available $\delta_{ij}$ and $W_{ij} = 0$ otherwise. The third issue is the bound constraints on certain distances and they can be generally represented by

$$L_{ij} \le D_{ij} \le U_{ij} \quad \text{for some} \ \ (i,j), \tag{4.9}$$

where $L_{ij}$ and $U_{ij}$ are lower and upper bounds for the distance $D_{ij}$. In the case there are anchors, $\mathbf{a}_i$, $i = 1, \ldots, m$, they should be fixed through $L_{ij} = U_{ij} = \|\mathbf{a}_i - \mathbf{a}_j\|^2$, $i, j = 1, \ldots, m$. Moreover, $L_{ii} = U_{ii} = 0$ represents $\mathrm{diag}(D) = 0$.

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

$$\begin{array}{ll} \min_{D,\mathbf{z}} & \frac{1}{2}\|W \circ [(D+Z) - \overline{\Delta}]\|^2 + \mu \mathcal{R}_1(\mathbf{z}) \\ \text{s.t.} & D \in \mathcal{B},\ g(D) = 0,\ Z \in \mathcal{S}^n_2, \end{array} \tag{4.10}$$

where $\mathcal{B} := \{D \in \mathcal{S}^n \mid L \leq D \leq U\}$ and $\circ$ is the Hadamard product (elementwise multiplication: $A \circ B := (A_{ij}B_{ij})$). We will show that the model (4.10) works very well when the sparse noises caused by few faulty nodes (outliers) such as in the single source localization [74] have a structural pattern, which refers to the fact that only the distances in the row corresponding to the single source are distorted.

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:

$$\begin{array}{ll} \min_{D,Z} & \frac{1}{2}\|W \circ [(D+Z) - \overline{\Delta}]\|^2 + \mu \mathcal{R}_2(Z) \\ \text{s.t.} & D \in \mathcal{B},\ g(D) = 0,\ Z \in \mathcal{S}^n, \end{array} \tag{4.11}$$

where $\mathcal{R}_2(Z)$ is a sparsity-induced regularization such as $\|Z\|_1$. Another choice is the $\ell_{1-2}$ regularization: $\mathcal{R}_2(Z) := \|Z\|_1 - \|Z\|$, also a popular choice in compressed sensing [98].

The FSMDS model (4.11) is also relevant to the sparsity-exploiting robust MDS method [35], where Kruskal's stress function [54] (with $\ell_1$ based regularizations) was used to measure the distance between the embedding distance $\|\mathbf{y}_i - \mathbf{y}_j\|$ and $\delta_{ij}$. Due to the nondifferentiablity and nonconvexity of the stress function, a SMACOF-style [30] majorization method was developed to solve the regularized problem. In contrast, we do not have non-differentiability issue 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 popular majorization technique (see, e.g., [88]), 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

$$\theta_m(\mathbf{x},\mathbf{y}) \geq \theta(\mathbf{x}) \text{ and } \theta_m(\mathbf{y},\mathbf{y}) = \theta(\mathbf{y}), \ \forall\ \mathbf{x},\mathbf{y} \in \Re^n. \tag{4.12}$$

Thus the function $g_m(\cdot,\cdot)$ in (2.38) is a majorization of $g(\cdot)$. Figure 4.1 demonstrates how majorization technique works. The one-dimensional function $f(x)$ is hard to minimize. At the current iterate $x^k$, a majorization function $m(x,x^k)$ is constructed and it is convex and quadratic. The function $m(;x^k)$ is easier to minimize and its optimal solution $x^{k+1}$ leads to a decrease of the original function. That is

$$f(x^{k+1} \leq f(x^k).$$

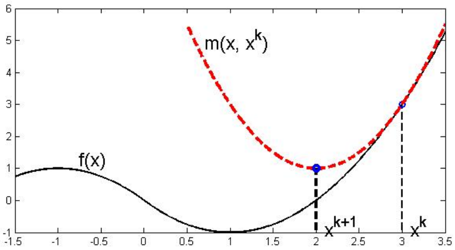


Figure 4.1: Illustration of majorization technique.

The other is the penalty technique. We will penalize the constraint $g(D) = 0$ in both (4.10) and (4.11) to their respective objective function. This penalty approach has been recently proposed in [102] to deal with the rank constraint $\text{rank}(JDJ) \leq r$ and it has been proved very effective. We also note that penalizing the squared distance function (note our $g(D)$ is so) is a widely adopted approach in statistical learning problems [23]. We will use the two techniques in the next two sections to solve the model (4.10) and (4.11) respectively.

## 4.4 Subspace Sparse MDS

In this section, we describe an efficient alternating majorization and minimization method for (4.10). For ease of description, let us define

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

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

### 4.4.1 The Penalty Approach and Its Majorization

As mentioned before, we penalize the nonlinear equation $g(D) = 0$ in (4.10) to the objective to obtain

$$\min_{D,\mathbf{z}} \ f_{\rho,\mu}(D,\mathbf{z}), \quad \text{s.t.} \quad D \in \mathcal{B}, \quad \mathbf{z} \in \Re^n. \tag{4.13}$$

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 $j$th 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 the fact

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

We just verified the conditions in (4.12) 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.

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

### 4.4.2 Algorithm: SSMDS

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

$$\left\{\begin{array}{ll} D^{k+1} & = \arg\min_{D\in\mathcal{B}}\ f^m_{\rho,\mu}(D,\mathbf{z}^k,D^k,\mathbf{z}^k) \\ \mathbf{z}^{k+1} & = \arg\min_{\mathbf{z}\in\Re^n}\ f^m_{\rho,\mu}(D^{k+1},\mathbf{z},D^k,\mathbf{z}^k). \end{array}\right. \tag{4.14}$$

We show that (4.14) 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

$$\begin{aligned} & D^{k+1} \\ = \ & \arg\min_{D\in\mathcal{B}} \frac{1}{2}\|W\circ(D-\overline{Z}^k)\|^2 + \frac{\rho}{2}\|D\|^2 + \rho\langle D^k_+,\ D\rangle \\ = \ & \arg\min_{D\in\mathcal{B}} \sum_{i,j}\left(\frac{1}{2}D^2_{ij} - \Delta^k_{ij}D_{ij}\right) \\ = \ & \arg\min_{D\in\mathcal{B}} \frac{1}{2}\|D-\Delta^k\|^2 \\ = \ & \Pi_{\mathcal{B}}(\Delta^k), \end{aligned} \tag{4.15}$$

where the matrix $\Delta^k$ is defined by

$$\Delta^k_{ij} := \left(W^2_{ij}\overline{Z}^k_{ij} - \rho(D^k_+)_{ij}\right)/(W^2_{ij}+\rho), \quad i = 1,\ldots,n \tag{4.16}$$

and

$$D^{k+1}_{ij} = \left(\Pi_{\mathcal{B}}(\Delta^k)\right)_{ij} := \min\left\{\max\{\Delta^k_{ij}, L_{ij}\}, U_{ij}\right\}. \tag{4.17}$$

(ii) Computing $\mathbf{z}^{k+1}$. We show that $\mathbf{z}^{k+1}$ can be computed through the soft-thresholding operator (3.6). 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 (3.6):

$$z_i^{k+1} = \mathcal{S}_{\mu/(2s_i^2)}(t_i^k / s_i), \quad i = 1, \ldots, n. \tag{4.18}$$

We summarize the algorithm below.

**Algorithm 3** 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 (4.16) and (4.17)
3: Update $\mathbf{z}^{k+1}$: Compute $\mathbf{z}^{k+1}$ through (4.18).
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.

## 4.5 Full-Space Sparse MDS

Similar to the previous section, this section develops an efficient algorithm for the full-space sparse MDS (4.11) with complete convergence analysis. Define

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

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

$$\min \ F_{\rho,\mu}(D, Z), \quad \text{s.t.} \quad D \in \mathcal{B}, \ Z \in \mathcal{S}^n. \tag{4.19}$$

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

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

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

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

$F^m_{\rho,\mu}$ is a majorization of $F_{\rho,\mu}$ because $g_m$ in (2.38) 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:

$$\begin{cases} D^{k+1} & = \arg\min_{D\in\mathcal{B}} \ F^m_{\rho,\mu}(D, Z^k, D^k, Z^k) \\ Z^{k+1} & = \arg\min_{Z\in\mathcal{S}^n} \ F^m_{\rho,\mu}(D^{k+1}, Z, D^k, Z^k). \end{cases} \tag{4.20}$$

### 4.5.1 Algorithm: FSMDS

For easy reference, we call the algorithm (4.20) 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

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

which is exactly what we have obtained in (4.15). Hence, $D^{k+1}$ can be computed by (4.16) and (4.17).

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

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

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 (3.6).

$$Z^{k+1}_{ij} = \begin{cases} \mathcal{S}_{\mu/W^2_{ij}}(\widehat{T}^k_{ij}) & \text{if } W_{ij} \neq 0 \\ 0 & \text{if } W_{ij} = 0, \end{cases} \tag{4.21}$$

with

$$\widehat{T}^k_{ij} := \overline{D}^{k+1}_{ij} + \mu T^k_{ij}/W^2_{ij} \quad \text{when } W_{ij} \neq 0. \tag{4.22}$$

We summarize FSMDS below.

**Algorithm 4** 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 (4.16) and (4.17).
3: Update $Z^{k+1}$. Compute $Z^{k+1}$ through (4.21) and (4.22).
4: Convergence check: Set $k := k + 1$ and go to Step 2 until convergence.
```

### 4.5.2 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:

$$\begin{aligned} F_{\rho,\mu}(D^k, Z^k) &= F^m_{\rho,\mu}(D^k, Z^k, D^k, Z^k) \quad \text{(by (4.12))} \\ &\geq F^m_{\rho,\mu}(D^{k+1}, Z^k, D^k, Z^k) \quad \text{(by (4.20))} \\ &\geq F^m_{\rho,\mu}(D^{k+1}, Z^{k+1}, D^k, Z^k) \quad \text{(by (4.20))} \\ &\geq F^m_{\rho,\mu}(D^{k+1}, Z^{k+1}, D^{k+1}, Z^{k+1}) \quad \text{(by (4.12))} \\ &\geq F_{\rho,\mu}(D^{k+1}, Z^{k+1}) \quad \text{(by (4.12))} \end{aligned}$$

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 (4.19), which satisfies the following first-order optimality condition:

$$\begin{cases} \langle \nabla_D F(D^*, Z^*) + \rho(D^* + \Pi_{\Pi_{\mathcal{K}^n_+(r)}}(-D^*)), D - D^* \rangle \geq 0, \\ \forall\ D \in \mathcal{B} \ \text{ and } \ \nabla_Z F(D^*, Z^*) + \mu(\Gamma^* - T^*) = 0, \end{cases} \tag{4.23}$$

for some $\Gamma^* \in \partial\|Z^*\|_1$ and $T^* \in \partial\|Z^*\|$. We summarize those properties in the following result, whose proof is in Appendix 4.8.1.

**Theorem 4.2.** *We assume that $\mathcal{B}$ is bounded and let $\{D^k, Z^k\}$ be the sequence generated by Alg. 4. Then the following hold.*

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

*(ii)* *We have*

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

*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 (4.19).*

Theorem 4.2 not only guarantees that any limit must satisfy the optimality condition of the problem (4.19), it also provides a practical stopping criterion for Alg. 4: 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. Now we turn our attention to the benefit of using $\ell_{1-2}$ regularization. The next result shows that we can control the sparsity in the generated iterates by setting the sparsity control parameter $\mu$ above certain computable threshold ($\mu_s$ 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 [35] or [59, 60] (or any of its variants) has such a useful property. As seen in Appendix 4.8.2, the proof of Theorem 4.3 makes use of the differentiability of $F(D, Z)$, which is a direct consequence of `cMDS` objective. In contrast, the objectives in [35, 59, 60] are not differentiable.

**Theorem 4.3.** *Suppose the initial point $Z^0 = 0$. Let $\{D^k, Z^k\}$ be the sequence generated by Alg. 4. For a given positive integer $s$, there exists $\mu_s > 0$ such that for any $\mu \geq \mu_s$, the number of nonzeros in $Z^k$ is not greater than $2s$, i.e.,*

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

*Moreover, $\mu_s$ 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. 4.2 and Thm. 4.3 are also valid when $\mathcal{R}_2(Z) = \|Z\|_1$. Therefore, Alg. 3 also enjoys the properties stated in the two theorems.

## 4.6 Numerical Experiments

In this part, we will test SSMDS and FSMDS on a few challenging localization problems, benchmarked against three other methods RMSD [35], HQMMDS [59] and TMDS [9]. They are all the latest methods for detecting outliers and both RMDS and HQMMDS also employ $\ell_1$-type sparsity-driven regularizations to induce sparsity. TMDS detects violations of triangle inequalities when $\Delta$ is viewed as a weighted graph and aims to correct those violations so that the modified $\Delta$ is close to being Euclidean. In terms of free parameters in those methods, SMMDS and FSMDS have two ($\rho$ and $\mu$), RMDS has one (sparsity control parameter $\lambda$), HQMMDS has two (sparsity control parameter $\lambda_1$ and the smoothness regularization parameter $\lambda_2$), TMDS has one (number of estimated outliers).

All tests were run in Matlab 2017a. A key computational task in our implementation is to compute $D_+^k = \Pi_{\mathcal{K}_+^n(r)}(-D^k)$, which has been well addressed in [102, Eq.(15)-(16)]. The important message is that it can be cheaply computed by just computing the first $r$ leading eigenvalues and the eigenvectors of a matrix (use the build-in `eigs` in Matlab). The stopping criterion for FSMDS is

$$\texttt{Fprog}_k := \frac{F_{\rho,\mu}(D^{k-1}, Z^{k-1}) - F_{\rho,\mu}(D^k, Z^k)}{1 + F_{\rho,\mu}(D^{k-1}, Z^{k-1})} \leq 10^{-5}.$$

Since the sequence $\{F_{\rho,\mu}(D^k, Z^k)\}$ is nonincreasing and is bounded from below by 0, the criterion is well defined. For SSMDS, $F_{\rho,\mu}(D^k, Z^k)$ should be replaced by $f_{\rho,\mu}(D^k, z^k)$. 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 $\Delta$. The inputs for other methods are 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 (4.9), which is an effective way to improve localization accuracy. However, the box constraints may create big challenges for other methods. The section is organized according to the types of testing problems. The first subsection is for problems that come under the framework of multiple source localization, followed by the single source location problem. The final subsection is on a real test data.

### 4.6.1 Multiple source localization

We test a problem of the "plus" (+) sign data that was first tested in [35]. It was generated as follows

**Example 4.1.** *(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 $\epsilon_{ij}$ follows the normal distribution with 0 mean and the variance $\sigma^2$. 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 $\delta_{ij}$. 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 [35] is without the four anchors being fixed. We tested the original data and then used the Procrustes (`procustes.m` Matlab build-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 surprising 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 [2] and [79, Sect. IV] for the ways to derive such mapping $\mathcal{T}$.

The following instances of Example 4.1 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 its default values 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. 4.2 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.

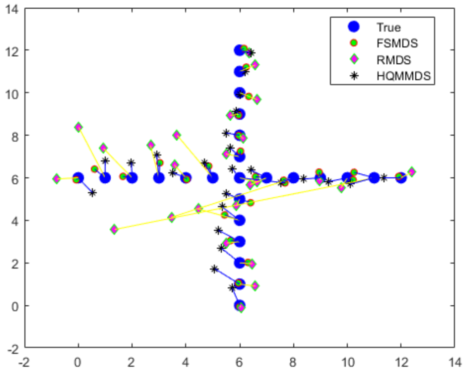


Figure 4.2: Embedding for Example 4.1 ($\sigma^2 = 0.1$ and $s = 60$) by FSMDS, RMDS and HQMMDS, all linked to the corresponding true locations. The percentage of the outliers is about $60/(300 - 6) \approx 20\%$. The corresponding RMSE is 0.5496 for FSMDS, 2.6517 for RMDS, and 0.7245 for HQMMDS.

It can be visibly observed from Fig. 4.2 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. 4.2 and other links are very close to their true locations. In contrast, the distances by HQMMDS stay quite close to the diagonal line,

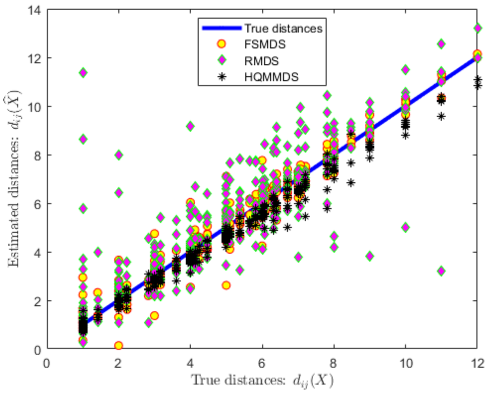


Figure 4.3: Shepard graph for the embeddings in Fig. 4.2. Shepard graph plots the true distance ($x$ axis) against its estimated distance ($y$ axis). If the plot is close the to diagonal line, the estimated distances are more accurate.

but many of them are blow the line, suggesting that HQMMDS tends to under-estimate the true distances.

Table 4.1: RMSE for Example 4.1 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')`.

| | | Methods | | | | |
|---|---|---|---|---|---|---|
| $\sigma^2$ | s | SSMDS | FSMDS | RMDS | TMDS | HQMMDS |
| 0.1 | 15 | 4.24 | 0.32 | 0.38 | 0.97 | 0.40 |
| | 30 | 5.36 | 0.52 | 0.77 | 2.24 | 0.63 |
| | 45 | 5.92 | 0.71 | 1.29 | 2.90 | 0.98 |
| | 60 | 6.24 | 1.10 | 1.77 | 3.45 | 1.23 |
| | 75 | 6.52 | 1.66 | 2.39 | 4.04 | 1.66 |
| 0.2 | 15 | 4.28 | 0.38 | 0.46 | 1.11 | 0.48 |
| | 30 | 5.40 | 0.58 | 0.80 | 2.25 | 0.70 |
| | 45 | 5.95 | 0.83 | 1.36 | 2.93 | 1.05 |
| | 60 | 6.27 | 1.14 | 1.86 | 3.47 | 1.39 |
| | 75 | 6.53 | 1.66 | 2.42 | 4.07 | 1.71 |

We further tested 10 instances of Example 4.1 and the corresponding average RMSE over 1000 simulations for each instance is reported in Table 4.1. We observe 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 [59] for more details). The poor results by TMDS demonstrate that detecting all violated triangle inequalities in the data matrix

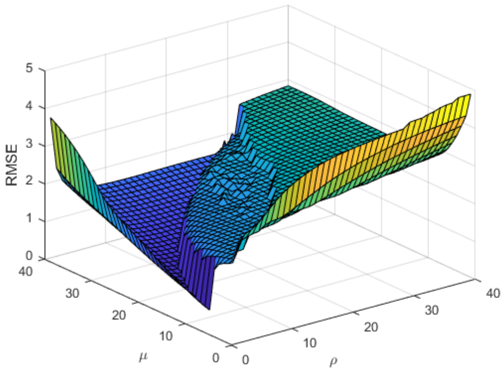


Figure 4.4: RMSE of FSMDS on Example 4.1 with $\sigma^2 = 0.1$ and $s = 60$, `rng('default')`. The parameters $(\rho, \mu)$ vary on the grid of $[1, 40] \times [1, 40]$. The lowest RMSE is when $(\rho, \mu) = (1, 6)$.

$\Delta$ is not adequate to locate the true locations of the data points for most instances. We note that it is a multiple sources localization problem and SSMDS completely failed. This is expected because, as our theoretical result suggested, it is more suitable to single source localization problems. Finally, we address another issue concerning the sensibility of FSMDS on its two parameters $\rho$ (penalty parameter) and $\mu$ (sparsity parameter). We tested FSMDS on a grid $[1, 40] \times [1, 40]$ for $(\rho, \mu)$ with unit step and plotted the corresponding RMSE in Fig. 4.4. It is interesting to see that RMSE in terms of $(\rho, \mu)$ 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 $\lambda_1$ and $\lambda_2$. Its lowest RMSE occurred at $(\lambda_1, \lambda_2) = (1, 35)$. We used those values in our extensive tests in Table 4.1.

### 4.6.2 Single source localization

This is the hard test problem proposed in [96] with negative and positive measurement errors that lead to outliers.

**Example 4.2.** *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* $\mathbf{x}_i$*,* $i = 1,\dots,N$ *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 uniformly distributed between* 0 *and* $\omega_i$*,* $i = 0,1,\dots,N$*. Here,* $\omega_i$ *can be treated as error upper bounds. We tested the first three scenarios in [96]. Case 1:* $\omega_0 = 5\alpha$ *and* $\omega_i = 0.5$ *for* $i = 1,\dots,N$*. Case 2:* $\omega_0 = 3$ *and* $\omega_i = 5\alpha$ *for* $i = 1,\dots,N$*. Case 3:* $\omega_0 = 0.5\alpha$ *and* $\omega_i = 5\alpha$ *for* $i = 1,\dots,N$*. In all three cases,* $\alpha$ *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 $\eta_i$ 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 4.2 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 $\alpha = 0.9$ in Case 2 and Case 3, for which HQMMDS works better than FSMDS. We also plotted the results in Fig. 4.5 for Case 2 for $\alpha$ varying from 0.1 to 1. It is obvious that the line by SSMDS is the lowest up to $\alpha = 0.9$ and HQMMDS works slightly better than SSMDS when $\alpha \geq 0.9$. 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.

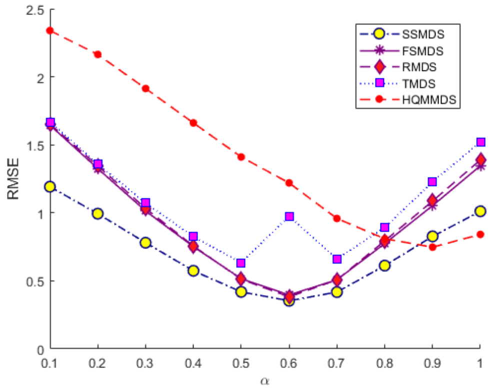


Figure 4.5: Average error (RMSE) vs $\alpha$ varying from 0.1 to 1 for Case 2 in Example 4.2 over 1000 simulations of test data.

Table 4.2: Average error for Example 4.1 ($N = 4$) by the five methods over 1000 random simulations of each test instance.

| | | Methods | | | | |
|---|---|---|---|---|---|---|
| Cases | $\alpha$ | SSMDS | FSMDS | RMDS | TMDS | HQMMDS |
| Case 1 | 0.3 | 0.59 | 0.76 | 0.77 | 0.87 | 1.66 |
| | 0.6 | 1.23 | 1.66 | 1.68 | 1.69 | 2.42 |
| | 0.9 | 1.81 | 2.54 | 2.56 | 2.46 | 3.08 |
| Case 2 | 0.3 | 0.79 | 1.04 | 1.06 | 1.08 | 1.93 |
| | 0.6 | 0.36 | 0.40 | 0.39 | 0.54 | 1.19 |
| | 0.9 | 0.83 | 1.06 | 1.09 | 1.21 | 0.75 |
| Case 3 | 0.3 | 0.62 | 0.78 | 0.80 | 0.89 | 0.80 |
| | 0.6 | 1.24 | 1.66 | 1.73 | 1.83 | 0.98 |
| | 0.9 | 1.95 | 2.59 | 2.71 | 3.05 | 1.82 |

### 4.6.3 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 [68]. 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 [68]). This implies that all of the measurements have large errors (positive or negative). In particular, there are 37 negative pairwise distances in $\Delta$ (In our test, we replace them by $|\delta_{ij}|$). This data has been studied in [29], 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 been explored in the two previous examples. One is using the weights $W_{ij}$ to distinguish importance of individual $\delta_{ij}$ to the objective. The other is enforcing tighter lower and upper bounds in (4.9).

**(a) Sammon weighting scheme**. It is a popular choice in scaling dissimilarity data proposed by Sammon [78], see also [12, P.255]. Each weight $W_{ij}$ is inversely proportional to $\delta_{ij}$. 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 $\rho$ and $\mu$. A generalized choice is $W_{ij} = \delta_{ij}^q$ with $q \in \Re$ being properly chosen and is proposed in [15]. 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 $\delta_{ij}$ is available. The results are reported in Table 4.3 for both types of weights. It can be clearly seen that Sammon weights effectively drove RMSE below 1m for both SSMDS and FSMDS. In particular, despite being designed for single source localization, SSMDS works well for this multiple source localization problem. 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 from their authors do not have such flexibility. The visualization of the obtained localization for the data by FSMDS and HQMMDS was plotted in Fig. 4.6. It is not surprising to note that TMDS completely failed this data because there are too many outliers and TMDS is mainly designed for situations with sparse outliers.

Table 4.3: Effect of Sammon weights on RMSE for Motorola data with $\alpha = 3$ and $\rho = 20$, $\mu = 90$ for SSMDS and FSMDS, and $\lambda_1 = 20$, $\lambda_2 = 100$ for HQMMDS.

| | Methods | | | | |
|---|---|---|---|---|---|
| Weights | SSMDS | FSMDS | RMDS | TMDS | HQMMDS |
| Standard | 1.24 | 1.17 | 1.09 | 9.94 | 1.04 |
| Sammon | 0.96 | 0.94 | 1.09 | 9.94 | 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

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

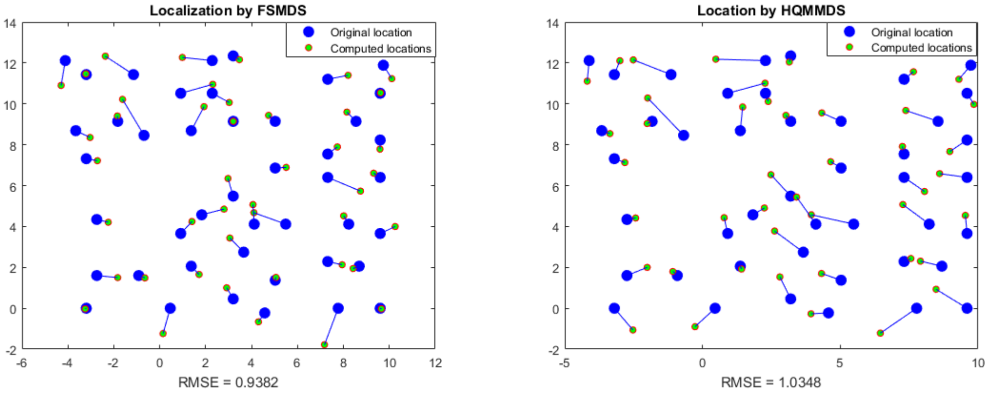


Figure 4.6: Visualization difference between RMSE less than 1m (left graph by FSMDS) and RMSE above 1m (right graph HQMMDS): There appears bigger localization errors among the points near boundary for the right graph than the left.

As $\beta$ varies from 0 to 1, the bounds in (4.9) 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. 4.7, 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 [44]. 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.

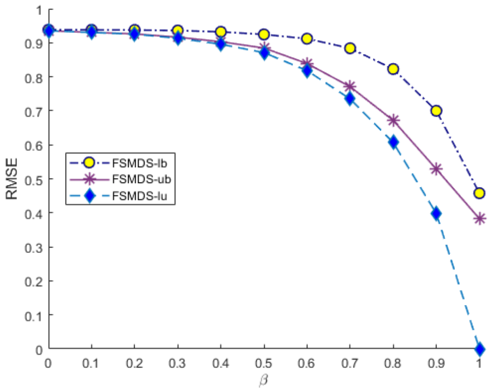


Figure 4.7: Power of adding lower and upper bounds: as the bounds get tighter as $\beta$ increases, FSMDS yields better localization. FSMDS-lb: adding lower bounds only; FSMDS-ub: adding upper bounds only; FSMDS-lu: adding both lower and upper bounds simultaneously.

## 4.7 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 chapter provides a brand new interpretation of cMDS and casts it as a joint optimization problem with one variable residing in the conditionally positive semidefinite cone $\mathcal{K}^n_+$ and the other in the subspace $\mathcal{S}^n_2$. This new reformulation also reveals why cMDS tends to overly denoise even there is just one erroneous dissimilarity. Continuing from Chapter 3, we considered 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.

## 4.8 Technical Proofs

### 4.8.1 Proof of Theorem 4.2

Please refer to Sect. 4.2 and Sect. 4.5 for the definition of the functions $g(D)$, $h(D)$, $g_m(D, A)$ and $F(D, Z)$, $F_\mu(D, Z)$, $F_{\rho,\mu}(D, Z)$ and its majorization function $F^m_{\rho,\mu}(D, Z, D^k, Z^k)$. We further let $\varphi(Z) := \|Z\|_1 - \|Z\|$. We will need the following inequalities.

$$h(-D^{k+1}) - h(-D^k) \geq \langle \Pi_{\mathcal{K}^n_+(r)}(-D^k), D^k - D^{k+1} \rangle \tag{4.24}$$

due to the convexity of $h(\cdot)$ and $\Pi_{\mathcal{K}^n_+(r)}(-D^k) \in \partial h(-D^k)$ by (2.37).

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

$$\langle \Omega_{k+1},\ D - D^{k+1} \rangle \geq 0, \quad \forall\ D \in \mathcal{B}, \tag{4.25}$$

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

$$\nabla_Z F(D^{k+1}, Z^{k+1}) + \mu(\Gamma^{k+1} - T^k) = 0. \tag{4.26}$$

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 4.4.** $\tau_k \geq 0$.

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

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

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

$$\langle\Gamma^{k+1},\ Z^{k+1}\rangle = \|Z^{k+1}\|_1, \quad \langle\Gamma^{k+1}, Z\rangle \leq \|Z\|_1, \forall\ Z. \tag{4.27}$$

Now computing the inner product with $(Z^k - Z^{k+1})$ on both sides of (4.26) leads to

$$\begin{aligned} & \langle\nabla_Z F(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1}\rangle \\ = \quad & \mu\langle\Gamma^{k+1} - T^k, Z^{k+1} - Z^k\rangle \\ = \quad & \mu\langle\Gamma^{k+1}, Z^{k+1}\rangle - \mu\langle\Gamma^{k+1}, Z^k\rangle - \mu\langle T^k, Z^{k+1} - Z^k\rangle \\ \overset{(4.27)}{=} \quad & \mu\|Z^{k+1}\|_1 - \mu\langle\Gamma^{k+1}, Z^k\rangle - \mu\langle T^k, Z^{k+1} - Z^k\rangle. \end{aligned}$$

Substituting the above into $\tau_k$ and simplifying to get

$$\begin{aligned} \tau_k \quad = \quad & \mu\Big(\underbrace{\|Z^k\|_1 - \langle Z^k, \Gamma^{k+1}\rangle}_{\geq 0 \text{ due to } (4.27)}\Big) \\ & +\mu\Big(\underbrace{\|Z^{k+1}\| - \|Z^k\| - \langle T^k, Z^{k+1} - Z^k\rangle}_{\geq 0 \text{ due to the convexity of } \|Z\|}\Big) \end{aligned}$$

This completes the proof. □

The following two identities can be verified directly.

$$\begin{aligned} & \|D^{k+1}\|^2 - \|D^k\|^2 \\ = \quad & 2\langle D^{k+1} - D^k, D^{k+1}\rangle - \|D^{k+1} - D^k\|^2. \end{aligned} \tag{4.28}$$

$$\begin{aligned} & \nabla_D F(D^{k+1}, Z^{k+1}) - \nabla_D F(D^{k+1}, Z^k) \\ = \quad & (W \circ W) \circ (Z^{k+1} - Z^k). \end{aligned} \tag{4.29}$$

**Proof.** (Thm. 4.2) (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 $(\ell, j)$. According to the update rule (4.21), 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

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

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.

$$
\begin{aligned}
& F_{\rho,\mu}(D^k, Z^k) - F_{\rho,\mu}(D^{k+1}, Z^{k+1}) \\
= \;& F(D^k, Z^k) - F(D^{k+1}, Z^{k+1}) \\
+ \;& \rho(g(D^k) - g(D^{k+1}) + \mu(\varphi(Z^k) - \varphi(Z^{k+1})) \\
= \;& \langle \underbrace{\nabla_D F(D^{k+1}, Z^{k+1})}_{\text{apply (4.29)}}, D^k - D^{k+1} \rangle \\
+ \;& \langle \nabla_Z F(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1} \rangle \\
+ \;& \frac{1}{2} \underbrace{\langle W \circ (D^k - D^{k+1}),\; W \circ (D^k - D^{k+1}) \rangle}_{\geq 0} \\
+ \;& \frac{1}{2} \langle W \circ (Z^k - Z^{k+1}),\; W \circ (Z^k - Z^{k+1}) \rangle \\
+ \;& \langle W \circ (Z^k - Z^{k+1}),\; W \circ (D^k - D^{k+1}) \rangle \\
+ \;& \frac{\rho}{2} \Big( \underbrace{\|D^k\|^2 - \|D^{k+1}\|^2}_{\text{apply (4.28)}} \Big) + \rho \Big( \underbrace{h(-D^{k+1}) - h(-D^k)}_{\text{apply (4.24)}} \Big) \\
+ \;& \mu(\varphi(Z^k) - \varphi(Z^{k+1})) \\
\geq \;& \underbrace{\langle \Omega_{k+1}, D^k - D^{k+1} \rangle}_{\geq 0 \text{ by (4.25)}} + \frac{\rho}{2} \|D^k - D^{k+1}\|^2 \\
+ \;& \langle \nabla_Z f(D^{k+1}, Z^{k+1}), Z^k - Z^{k+1} \rangle \\
+ \;& \frac{1}{2} \langle W \circ (Z^k - Z^{k+1}),\; W \circ (Z^k - Z^{k+1}) \rangle \\
+ \;& \mu(\varphi(Z^k) - \varphi(Z^{k+1})) \\
\geq \;& \frac{\rho}{2} \|D^k - D^{k+1}\|^2 \; + \; \tau_k \\
& + \frac{1}{2} \langle W \circ (Z^k - Z^{k+1}),\; W \circ (Z^k - Z^{k+1}) \rangle.
\end{aligned}
$$

Lemma 4.4 ($\tau_k \geq 0$) 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

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

Taking the limits on both sides of (4.26) for $k \in K$ to obtain

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

And taking the limits on both sides of (4.25) 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 (4.23). This completes our proof. □

### 4.8.2 Proof of Theorem 4.3

**Proof.** The proof technique is taken from [98]. It follows from (4.26) 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, $\|T^{k-1}\| \leq 1$ and $\|\Gamma^k\| \geq \sqrt{\|Z^k\|_0}$, which imply

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

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 $\mu$. □

# Chapter 5

# Conclusions

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. However, when a small number of dissimilarity data contains large errors, `cMDS` is known to work poorly. Our new interpretation of `cMDS` as a joint optimization of a matrix variable and a vector clearly shows why `cMDS` would not work in this case. It is simply because that it tries to denoise through a dense matrix. This new interpretation opens a venue for us to mend its capability of outlier removal.

`cMDS` is essentially a nonconvex optimization and can be reformulated as the following joint optimization:

$$\begin{array}{ll} \min_{D,Z} & \|\Delta - (D+Z)\|^2 \\ \text{s.t.} & D \in \mathcal{D}^n, \quad \text{rank}(JDJ) \leq r \\ & Z \in V, \end{array}$$

where $V$ is a space. When $V = \mathcal{S}_2^n$, the above problem becomes `cMDS`.

Due the rank constraint, the problem is nonconvex. We first study a convex relaxation approach and we develop the algorithm PADMM. It works when the number of outlier is not too big and its noisy level is not too high. We further proposed two nonconvex approaches, resulting in two methods SSMDS and FSMDS. We establish the convergence of the two methods and compare them with several state-of-the art methods. And they work very well for the synthetic and the real data problems.

There are a few topics that are definitely worth further investigation. We discuss them below.

(i) (exact penalty) The penalty method used in both SSMDS and FSMDS is not an exact penalty. Recall for a constrained optimization below

$$\min \ f(\mathbf{x}) \quad \text{s.t.} \ \ h(\mathbf{x}) = 0, \tag{5.1}$$

where $f : \Re^n \mapsto \Re$ and $h : \Re^n \mapsto \Re^m$ are two given functions, we consider its penalty problem:

$$\min \ f(\mathbf{x}) + \rho h(\mathbf{x}) \tag{5.2}$$

where $\rho > 0$ is a penalty parameter. Through penalty, the constrained problem (5.1) becomes an unconstrained one (5.2).

If the optimal solution of (5.2) is also an optimal solution of (5.1), we say the penalty is exact. In our research (Chapter 4), the penalty is not exact. It certainly is an interesting problem that if we can design an exact penalty for both SSMDS and FSMDS.

(ii) (a two-stage approach) We have recognized the tremendous difficulty in outlier removal. Our approach so far aims to accomplish this task by solving one optimization problem. We wonder if it is more reasonable to another step that makes use of the solution provided here and continue to the stage of fine removal. We regard it as a two-stage approach. The second step would be more on heuristic (e.g., random removal). But it remains unclear how we shall proceed.

(iii) (extension) Our study is mainly on the `cMDS`. In literature, there are other types of MDS. For example, the stress criterion [12] is another popular choice. We wonder whether the analysis here can be extended to the stress criterion. It would be interesting to see how it works when coming to testing those data that we tested in this thesis.

# References

[1] A.Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comput. Optim. Appl., 12 (1999), pp. 13–30.

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

[3] S. Bai and H.-D. Qi, *Tackling the flip ambiguity in wireless sensor network localization and beyond*, Digital Signal Process., 55 (2016), pp. 85-97.

[4] S. Bai, H.-D. Qi and N. Xiu, *Constrained best Euclidean distance embedding on a sphere: a matrix optimization approach*, SIAM J. Optim., 25 (2015), pp. 439–467.

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

[6] H.M. Berman, J. Westbrook, Z. Feng, G. Gillilan, T.N. Bhat, H. Weissig, I.N. Shindyalov and P.E. Bourne, The protein data bank, *Nucleic Acids Res.* 28, pp. 235–242, 2000.

[7] P. Biswas and Y. Ye, Semidefinite programming for ad hoc wireless sensor network localization, Information Processing in Sensor Networks, 2004, pp. 46-54.

[8] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang and Y. Ye, *Semidefinite programming approaches for sensor network localization with noisy distance measurements*, IEEE Trans. Auto. Sci. Eng., 3, pp. 360-371, 2006.

[9] L. Blouvshtein and D. Cohen-Or, *Outlier detection for robust multi-dimensional scaling*, IEEE Trans. Pattern Recog. Machine Intell. to appear.

[10] C. Boesch, Molecular aspects of magnetic resonance imaging and spectroscopy, *Mol. Aspects Med.* 20 (1999), pp. 185–318.

[11] J.F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems.* Springer Series in Operations Research, Springer, 2000.

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

[13] R. Borsdorf and N.J. Higham, *A preconditioned Newton algorithm for the nearest correlation matrix*, IMA Journal of Numerical Analysis 94 (2010), pp. 94-107.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Foundations and Trends in Machine Learning 3(1),(2011), 1–122.

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

[16] S. Cacciatore, C. Luchinat and L. Tenori, *Knowledge discovery by accuracy maximization*, Proc. Natl. Acad. Sci. 111 (2014), pp. 5117–5122.

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

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

[19] X. Chen, H.-D. Qi, and P. Tseng, *Analysis of nonsmooth symmetric matrix valued functions with applications to semidefinite complementarity problems*, SIAM J. Optim. 13 (2003), pp. 960–985.

[20] Z.X. Chan and D.F. Sun, *Constraint nondegeneracy, strong regularity and nonsingularity in semidefinite programming*, SIAM J. Optim. 19, 370–396 (2008).

[21] 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 (2012), pp. 108-116.

[22] H. Choi and S. Choi, *Robust kernel Isomap*, Pattern Recognition 40 (2007), pp. 853–862.

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

[24] F.H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983.

[25] L.G. Cooper, *A new solution to the additive constant problem in metric multidimensional scaling*, Psychometrika 37 (1972), pp. 311–322.

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

[27] G. Crippen and T. Havel, *Distance Geometry and Molecular Conformation.* New York: Wiley, 1988

[28] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry.* Meboo Publishing USA,2005.

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

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

[31] V. de Silva and J.B. Tenenbaum, *Sparse multidimensional scaling using landmark points*, Technical report, Stanford University, 2004.

[32] I. Dokmanic, R. Parhizkar, J. Renieri and M. Vetterli, *Euclidean distance matrices: Essential theory, algorithms and applications*, IEEE Signal Processing Magazine, November (2015), pp. 12–30.

[33] J. Fan and J. Lv, *Nonconcave penalized likelihood with NP-dimensionality.* IEEE Trans. Inform. Theory 57 (2011), pp. 5467–5484.

[34] M. Fazel, T.K. Pong, D.F. Sun and P. Tseng, *Hankel matrix rank minimization with applications to system identification and realization*, SIAM J. Matrix Analysis and Applications, 34 (2013), pp. 946–977.

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

[36] L.C. Freeman, *Graphic techniques for exploring social networks data*, Models and Methods in Social Network Analysis, 2005, pp. 248–269.

[37] N. Gaffke and R. Mathar, *A cyclic projection algorithm via duality*, Metrika, 36 (1989), pp. 29–54.

[38] Y. Gao, *Structured Low Rank Matrix Optimization Problems: a Penalty Approach*, PhD Thesis (2010), National University of Singapore.

[39] Y. Gao and D.F. Sun, *A majorized penalty approach for calibrating rank constrained correlation matrix problems.* Technical Report, Department of Mathematics, National University of Singapore, March 2010.

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

[41] W. Glunt, T.L. Hayden, and W.-M. Liu, *The embedding problem for predistance matrices*, Bulletin of Mathematical Biology, 53 (1991), pp. 769–796.

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

[43] W. Glunt, T.L. Hayden, and R. Raydan, *Preconditioners for distance matrix algorithms*, J. Computational Chemistry, 15 (1994), pp. 227–232.

[44] 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.

[45] J.C. GOWER, *Some distance properties of latent root and vector methods used in multivariate analysis*, Biometrika (1966), 53, 3 and 4, pp. 325–338.

[46] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms I*, Springer-Verlag, Berlin 1993.

[47] M.R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards 49 (1952), pp. 409–436.

[48] R. HORAU, *A Short Tutorial on Graph Laplacians, Laplacian Embedding, and Spectral Clustering.* Radu Horaud INRIA Grenoble Rhone-Alpes, France. Available from http://perception.inrialpes.fr/

[49] K.F. JIANG, D.F. SUN, AND K.-C. TOH, *Solving nuclear norm regularized and semidefinite matrix least squares problems with linear equality constraints*, Fields Institute Communications Series on Discrete Geometry and Optimization, K. Bezdek, Y. Ye, and A. Deza eds., 2013.

[50] C.R. JOHNSON AND P. TARAZAGA, *Connections between the real positive semidefinite and distance matrix completion problems*, Linear Algebra Appl. 223/224 (1995), pp. 375–391.

[51] I.T. JOLLIFFE, *Principal Component Analysis*, 2nd Eds, Springer, 2002.

[52] *S. Kim, M. Kojima, H. Waki, M. Yamashita*, *Algorithm 920: SFSDP: A sparse version of full semide*

*nite programming relaxation for sensor network localization problems*, ACM Trans. Math. Softw. 38 (4) (2012), pp. 1-27.

[53] N. KRISLOCK AND H. WOLKOWICZ, *Euclidean distance matrices and applications*, In *Handbook of Semidefinite, Cone and Polynomial Optimization*, M. Anjos and J. Lasserre (Editors), 2010.

[54] J.B. KRUSKAL, *Nonmetric multidimensional scaling: a numerical method*, *Psychometrika*, 29 (1964), pp. 115-129.

[55] Y. LeCun, C. Cortes and C.J.C. Burges, *Mnist*, 1998. URL: `http://yann.lecun.com/exdb/mnist.`

[56] P. Legendre and M.J. Anderson, *Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments*, Ecological Monographs, 69 (1999), pp. 1–24.

[57] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, *Euclidean distance geometry and applications*, SIAM Rev. 56 (2014), pp. 3–69

[58] J.C. Lingoes, *Some boundary conditions for a monotone analysis of symmetric matrices*, Psychometrika 36 (1971), pp. 195–203.

[59] 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.

[60] F.D. Mandanas and C.L. Kotropoulos, *M-estimators for robust multidimensional scaling employing $\ell_{21}$-norm regularization*, Pattern Recognition, 73 (2018), 235-246.

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

[62] K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis*, Academic Press, London, 1979.

[63] R. Mathar, *The best Euclidean fit to a given distance matrix in prescribed dimensions*, Linear Algebra and Its Applications 67 (1985), pp. 1–6.

[64] C.A. Micchelli, *Interpolation of scattered data: distance matrices and conditional positive definite functions*, Constr. Approx. 2 (1986), pp. 11–22.

[65] C.A. Micchelli and F.I.Utreras, *Smoothing and interpolation in a convex subset of a Hilbert space,* SIAM J. Sci. Statist. Comput. 9 (1988), pp. 728–747.

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

[67] E. Pękalska, P. Paclík, and P.W. Duin, *A generalized kernel approach to dissimilarity-based classification*, J. Machine Learn. Res., 2 (2002), pp. 175–211.

[68] 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 (2003), 2137–2148.

[69] T.K. Pong, *Edge-based semidefinite programming relaxation of sensor network localization with lower bound constraints*, Comput. Optim. Appl., 53 (2012), pp. 23–44.

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

[71] H.-D. Qi, *Conditional quadratic semidefinite programming: examples and methods*, J. Oper. Res. Soc. China 2 (2014), pp. 143–170.

[72] H.-D. Qi, *A convex matrix optimization for the additive constant problem in multidimensional scaling with application to locally linear embedding*, SIAM J. Optimization, 26-4 (2016), pp. 2564–2590.

[73] H.-D. Qi and D.F. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl. 28 (2006), pp. 360–385.

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

[75] H.-D. Qi and X.M. Yuan, *Computing the nearest Euclidean distance matrix with low embedding dimensions*, Math. Program. 147 (2014), pp. 351–389.

[76] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Prog. 58 (1993), 353–367.

[77] S.T. Roweis and L.K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.

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

[79] 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.

[80] L.K. Saul and S.T. Roweis, *Think globally, fit locally: unsupervised learning of low dimensional manifolds*, J. Machine Learning Res., 4 (2003), pp. 119–155.

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

[82] I.J. Schoenberg, *Metric spaces and positive definite functions*, Trans. Amer. Math. Soc., 44 (1938), pp. 522–536.

[83] F. Shang, L.C. Jiao, J. Shi and J. Chai, *Robust positive semdefinite L-Isomap ensemble*, Pattern Recognition Letters 32 (2011), pp. 640–649.

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

[85] H. Siu, L. Jin and M. Xiong, *Manifold learning for human population structure studies*, PLoS ONE 7 (2012), pp. 1–18.

[86] D.F. Sun, *The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications*, Math. Oper. Res. 31 (2006), 761–776.

[87] D.F. Sun and J. Sun, *Semismooth matrix valued functions*, Math. Oper. Res. 27 (2002), pp. 150–169.

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

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

[90] K.C. Toh, *An inexact path-following algorithm for convex quadratic SDP*, Mathematical Programming, 112 (2008), pp. 221–254.

[91] W.S. Torgerson, *Multidimensional scaling: I. Theory and method*, Psychometrika, 17 (1952), pp. 401–419.

[92] M.W. Trosset, *Distance matrix completion by numerical optimization*, Computational Optimization and Applications, 17 (2000), 11–22.

[93] 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.

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

[95] J. Wang, *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*, Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg 2012.

[96] 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(2016), pp. 3281-3296.

[97] J.F. Yang and Y. Zhang *Alternating direction algorithms for $\ell_1$-problems in compressive sensing*, SIAM journal on scientific computing 33-1 (2011), pp. 250-278.

[98] P. Yin, Y. Lou, Q. He, and J. Xin, *Minimization of $\ell_{1-2}$ for compressed sensing*, SIAM J. Sci. Comput., 37 (2015), A536-A563.

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

[100] L. Zhang, G. Wahaba and M. Yuan, *Distance shrinkage and Euclidean embedding via regularized kernel estimation*, J. R. Statist. Soc. B (2016).

[101] X. Zhao, D. Sun and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim. 20 (2010), pp. 1737–1765.

[102] 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 (2018), pp. 4331-4346.