How can classical multidimensional scaling go wrong?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Given a matrix D describing the pairwise dissimilarities of a data set, a common task is to embed the data points into Euclidean space. The classical multidimensional scaling (cMDS) algorithm is a widespread method to do this. However, theoretical analysis of the robustness of the algorithm and an in-depth analysis of its performance on non-Euclidean metrics is lacking.

In this paper, we derive a formula, based on the eigenvalues of a matrix obtained from D, for the Frobenius norm of the difference between D and the metric D_{cmds} returned by cMDS. This error analysis leads us to the conclusion that when the derived matrix has a significant number of negative eigenvalues, then $\|D - D_{\text{cmds}}\|_F$, after initially decreasing, will eventually increase as we increase the dimension. Hence, counterintuitively, the quality of the embedding degrades as we increase the dimension. We empirically verify that the Frobenius norm increases as we increase the dimension for a variety of non-Euclidean metrics. We also show on several benchmark datasets that this degradation in the embedding results in the classification accuracy of both simple (e.g., 1-nearest neighbor) and complex (e.g., multi-layer neural nets) classifiers decreasing as we increase the embedding dimension.

Finally, our analysis leads us to a new efficiently computable algorithm that returns a matrix D_l that is at least as close to the original distances as D_t (the Euclidean metric closest in ℓ_2 distance). While D_l is not metric, when given as input to cMDS instead of D, it empirically results in solutions whose distance to D does not increase when we increase the dimension and the classification accuracy degrades less than the cMDS solution.

1 Introduction

Multidimensional scaling (MDS) refers to a class of techniques for embedding data into Euclidean space given pairwise dissimilarities [Carroll and Arabie, 1998, Borg and Groenen, 2005]. Apart from the general usefulness of dimensionality reduction, MDS has been used in a wide variety of applications including data visualization, data preprocessing, network analysis, bioinformatics, and data exploration. Due to its long history and being well studied, MDS has many variations such as non-metric MDS [Shepard, 1962a,b], multi-way MDS [Kroonenberg, 2008], multi-view MDS [Bai et al., 2017], confirmatory or constrained MDS [Heiser and Meulman, 1983], etc. (See France and Carroll [2010], Cox and Cox [2008] for surveys).

The basic MDS formulation involves minimizing an objective function over a space of embeddings. There are two main objective functions associated with MDS: STRESS and STRAIN.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

The STRAIN objective (Equation 1 below) was introduced by Torgerson [1952], whose algorithm to solve for this objective is now commonly referred to as the classical MDS algorithm (cMDS).

$$X_{\text{cmds}} := \underset{X \in \mathbb{R}^{r \times n}}{\arg\min} \left\| X^T X - \left(\frac{-VDV}{2} \right) \right\|. \tag{1}$$

Here V is the centering matrix given by

$$V := I - \frac{1}{n} J, \tag{2}$$

and I is the identity matrix and J is the matrix of all ones. cMDS first centers the squares of the given distance matrix and then uses its spectral decomposition to extract the low dimensional embedding. cMDS is one of the oldest and most popular methods for MDS, and its popularity is in part due to the fact that this decomposition is fast and can scale to large matrices. The point set produced by cMDS, however, is not necessarily the point set whose Euclidean distance matrix is closest under say Frobenius norm to the input dissimilarity matrix. This type of objective is instead captured by STRESS, which comes in a variety of related forms. In particular, in this paper we consider the SSTRESS objective (see Equation 3).

Specifically, given an embedding X, let $EDM(X)$ be the corresponding Euclidean distance matrix, that is $\text{EDM}(X)_{ij} = \|X_i - X_j\|_F^2$, where X_i, X_j are the ith and jth columns of X. If D is a dissimilarity matrix whose entries are squared, then we are interested in the matrix,

$$D_t := \underset{D' = EDM(X), X \in \mathbb{R}^{r \times n}}{\arg\min} \|D' - D\|_F^2. \tag{3}$$

Note that the reason we assume our dissimilarity matrix has squared entries is because the standard EDM characterizations uses squared entries (see further discussion of EDMs below). Equation 3 is a well studied objective [Takane et al., 1977, Hayden and Wells, 1988, Qi and Yuan, 2014].

There are a number of similarly defined objectives. If one considers this objective when the matrix entries are not squared (i.e. $\sqrt{D'_{ij}} - \sqrt{D_{ij}}$), then it is referred to as STRESS. If one further normalizes each entry of the matrix difference by the input distance value (i.e. $(\sqrt{D'_{ij}} - \sqrt{D_{ij}})/\sqrt{D_{ij}}$) then it is called Sammon Stress. In this paper, we are less concerned with the differences between different types of Stress, and instead focus on how the cMDS solution behaves generally under a Stress type objective. Thus for simplicity we focus on SSTRESS. It is important to note that there are algorithms to solve the SSTRESS objective, but the main drawback is that they are slow in comparison to cMDS [Takane et al., 1977, Hayden and Wells, 1988, Qi and Yuan, 2014]. Thus, many practitioners default to using cMDS and do not optimize for SSTRESS.

In this paper, we shed light on the theoretical and practical differences between optimizing for these two objectives. Let $D_{\text{cmds}} := \text{EDM}(X_{\text{cmds}})$, where X_{cmds} is the solution to Equation 1, and let D_t be the solution to Equation 3. We are interested in understanding the quantity

$$\text{err} := \|D - D_{\text{cmds}}\|_F^2. \tag{4}$$

Doing so will provide practitioners with multiple advantages and will guide the development of better algorithms. In particular,

1. Understanding err is the first step in rigorously quantifying the robustness of the cMDS algorithm.
2. If err is guaranteed to be small, then we can use the cMDS algorithm without having to worry about loss in quality of the solution.
3. If err is big, we can make an informed decision about the benefits of the speed of the cMDS algorithm versus the quality of the solution.
4. Understanding when err is big helps us design algorithms to approximate D_t that perform better when cMDS fails.

Contributions. Our main theorem, Theorem 1, decomposes err into three components. This decomposition gives insight into when and why cMDS can fail with respect to the SSTRESS objective. In particular, for Euclidean inputs, err naturally decreases as the embedding dimension increases. For non-Euclidean inputs, however, our decomposition shows that after an initial decrease,

counterintuitively err can actually increase as the embedding dimension increases. In practice one may not know a priori what dimension to embed into, though one might assume it suffices to embed into some sufficiently large dimension. Importantly, these results demonstrate that when using cMDS to embed, choosing a dimension too large can actually increase error.

This degradation of the cMDS solution is of particular concern in relation to the robustness in the presence of noisy or missing data, as may often be the case for real world data. Several authors [Cayton and Dasgupta, 2006, Mandanas and Kotropoulos, 2016, Forero and Giannakis, 2012] have proposed variations to specifically address robustness with cMDS. However, our decomposition of err, suggests a novel approach. Specifically, by attempting to directly correct for the problematic term in our decomposition (which resulted in err increasing with dimension) we produce a new lower bound solution. We show empirically that this lower bound corrects for err increasing, both by itself and when used as a seed for cMDS. Crucially the running time of our new approach is comparable to cMDS, rather than the prohibitively expensive optimal SSTRES solution. Finally, and perhaps more importantly, we show that if we add noise or missing entries to real world data sets, then our new solution outperforms cMDS in terms of the downstream task of classification accuracy, under various classifiers. Moreover, our decomposition can be used to quickly predict the dimension where the increase in err might occur.

The main contributions of our paper are as follows.

1. We decompose the error in Equation 4 into three terms that depend on the eigenvalues of a matrix obtained from D. Using this analysis, we show that there is a term that tells us that as we increase the dimension that we embed into, eventually, the error starts increasing.
2. We verify, using classification as a downstream task, that this increase in the error for cMDS results in the degradation of the quality of the embedding, as demonstrated by the classification accuracy decreasing.
3. Using this analysis, we provide an efficiently computable algorithm that returns a matrix D_l such that if D_t is the solution to Equation 3, then $\|D_l - D\|_F \leq \|D_t - D\|_F$, and empirically we see that $\|D_l - D_t\|_F \leq \|D_{\text{cmds}} - D_t\|_F$.
4. While D_l is not metric, when given as input to cMDS instead of D, it results in solutions that are empirically better than the cMDS solution. In particular, this modified procedure results in a more natural decreasing of the error as the dimension increases and has better classification accuracy.

2 Preliminaries and Background

In this section, we lay out the preliminary definitions and necessary key structural characterizations of Euclidean distance matrices.

cMDS algorithm

For completeness, we include the classical multidimensional scaling algorithm in Algorithm 1.

Algorithm 1 Classical Multidimensional Scaling.

```
function cMDS(D, r)
    X = −V ∗ D ∗ V/2
    Compute μ_1 ≥ ... ≥ μ_r > 0, U as the eigenvalues and eigenvectors of X
    return U ∗ diag(√μ_1, ..., √μ_r, 0, ..., 0)
```

EDM Matrices

Definition 1. *$D \in \mathbb{R}^{n\times n}$ is a Euclidean Distance Matrix (EDM) if and only if there exists a $d \in \mathbb{N}$ such that there are points $x_1, \ldots, x_n \in \mathbb{R}^d$ with*

$$D_{ij} = \|x_i - x_j\|_2^2.$$

Note that unlike other distance matrices, an EDM consists of the squares of the (Euclidean) distances between the points.

This form permits several important structural characterizations of the cone of EDM matrices.

1. Gower [1985], Schoenberg [1935] show that a symmetric matrix D is an EDM if only if

$$F := -(I - \mathbf{1}s^T)D(I - s\mathbf{1}^T)$$

is a positive semi-definite matrix for all s such that $\mathbf{1}^T s = 1$ and $Ds \neq 0$.

2. Schoenberg [1938] showed that D is an EDM if and only if $\exp(-\lambda \mathrm{D})$ is a PSD matrix with 1s along the diagonal for all $\lambda > 0$. Note here exp is element wise exponentiation of the matrix.

3. Another characterization is given by Hayden and Wells [1988] in which D is an EDM if and only if D is symmetric, has 0s on the diagonal (i.e., the matrix is hollow), and $\hat{D}$ is negative semi-definite, where $\hat{D}$ is defined as follows

$$QDQ = \begin{bmatrix} \hat{D} & f \\ f^T & \xi \end{bmatrix}. \tag{5}$$

Here f is a vector and

$$Q = I - \frac{2}{v^T v} vv^T \text{ for } v = [1, \ldots, 1, 1 + \sqrt{n}]^T. \tag{6}$$

Note Q is Householder reflector matrix (in particular, it's unitary) and it reflects a vector about the hyperplane span$(v)^{\perp}$.

The characterization from Hayden and Wells [1988] is the main characterization that we shall use. Hence we establish some important notation.

Definition 2. *Given any symmetric matrix $A \in \mathbb{R}^{n \times n}$, let us define $\hat{A} \in R^{n-1 \times n-1}$, $f(A) \in \mathbb{R}^{n-1}$, and $\xi(A) \in \mathbb{R}$ as follows*

$$QAQ = \begin{bmatrix} \hat{A} & f(A) \\ f(A)^T & \xi(A) \end{bmatrix}.$$

In addition to characterizations of the EDM cone, we are also interested in the dimension of the EDM.

Definition 3. *Given an EDM D, the dimensionality of D is the smallest dimension d, such that there exist points $x_1, \ldots, x_n \in \mathbb{R}^d$ with $D_{ij} = \|x_i - x_j\|_2^2$.*

Let $\mathcal{E}(r)$ be the set of EDM matrices whose dimensionality is at most r.

Using Hayden and Wells [1988]'s characterization of EDMs, Qi and Yuan [2014] show $D \in \mathcal{E}(r)$ if and only if D is symmetric, hollow (i.e., 0s along the main diagonal), and $\hat{D}$ in Equation 5 is negative semi-definite with rank at most r.

Conjugation matrices: Q and V. Conjugating distance matrices either by Q (in Equation 6) or by the centering matrix V (in Equation 2) is an important component of understanding both EDMs and the cMDS algorithm. We observe that V is also (essentially) a Householder matrix like Q. Let $w = [1, \ldots, 1]^T$ and observe that $J = ww^T$ so that $V = I - \frac{1}{w^T w} ww^T$.

Qi and Yuan [2014] establishes an important connection between Q and V that we make use of in our analysis in Section 3. Specifically, for any symmetric matrix A, we have that

$$VAV = Q \begin{bmatrix} \hat{A} & 0 \\ 0 & 0 \end{bmatrix} Q. \tag{7}$$

Here $\hat{A}$ is the matrix given by Definition 2. This connection gives a new framework in which we can understand the cMDS algorithm. We know that given D, cMDS first computes $-VDV/2$. Using Equation 7, this is equal to $-Q \begin{bmatrix} \hat{D} & 0 \\ 0 & 0 \end{bmatrix} Q/2$. Thus, when cMDS computes the spectral decomposition of $-VDV/2$, this is equivalent of computing the spectral decomposition of $\hat{D}$. Then using the characterization of $\mathcal{E}(r)$ from Qi and Yuan [2014], setting all but the largest r eigenvalues to 0 might seem like the optimal solution. However, this procedure ignores condition that the matrix must be hollow. As we shall show, this results in sub-optimal solutions.

3 Theoretical Results

Throughout this section we fix the following notation. Let D be a distance matrix with squared entries. Let r be the dimension into which we are embedding. Let $\lambda_1 \leq \ldots \leq \lambda_{n-1}$ be the eigenvalues of $\hat{D}$ and U be the eigenvectors. Let $\boldsymbol{\lambda}$ be an n-dimensional vector where $\boldsymbol{\lambda}_i = \lambda_i \mathbb{1}_{\lambda_i > 0 \text{ or } i > r}$ for $i = 1, \ldots, n-1$ and $\boldsymbol{\lambda}_n = 0$. Let $S = Q * \begin{bmatrix} U & 0 \\ 0 & 1 \end{bmatrix}$. Let D_t be the solution to Problem 3 and D_{cmds} the resulting EDM from the solution to Problem 1. Let

$$C_1 = \sum_{i=1}^{n-1} \boldsymbol{\lambda}_i^2, \quad C_2 = -\sum_{i=1}^{n-1} \boldsymbol{\lambda}_i, \quad C_3 = \frac{n\|(S \circ S)\boldsymbol{\lambda}\|_F^2 - C_2^2}{2}.$$

Then the main result of the paper is the following spectral decomposition of the SSTRESS error.

Theorem 1. *If D is a symmetric, hollow matrix then, $\|D_{\text{cmds}} - D\|_F^2 = C_1 + C_2^2 + C_3$.*

The idea behind the proof of Theorem 1 is to decompose

$$\begin{aligned} \|D_{\text{cmds}} - D\|_F^2 &= \|QD_{\text{cmds}}Q - QDQ\|_F^2 \\ &= 4\|\hat{D}/2 - \hat{D}_{\text{cmds}}/2\|_F^2 + (\xi(D) - \xi(D_{\text{cmds}}))^2 + 2\|f(D) - f(D_{\text{cmds}})\|_F^2, \end{aligned}$$

which follows from Definition 2. We relate each of these three terms to C_1, C_2, and C_3. The following lemmas will work towards expressing each of these terms separately. In the following discussion, let $Y_r := {X_{\text{cmds}}}^T X_{\text{cmds}}$ and recall that X_{cmds} is the solution to the classical MDS problem given in Equation 1. The proofs for the following lemmas are in the appendix.

Lemma 1. *If G is a positive semi-definite Gram matrix, then $-\frac{1}{2}V\,EDM(G)\,V = Q \begin{bmatrix} \hat{G} & 0 \\ 0 & 0 \end{bmatrix} Q$*

Lemma 2. *The value of the objective function obtained by X_{cmds} in Equation 1 is $\frac{C_1}{4}$. Specifically, we have that*

$$4\|Y_r - (-VDV)/2\|_F^2 = \sum_{i=1}^{n-1} \boldsymbol{\lambda}_i^2 =: C_1.$$

Lemma 3. $-\frac{1}{2}\hat{D}_{\text{cmds}} = \hat{Y}_r$.

Lemma 4. *If $Tr(D) = 0$, then $(\xi(D) - \xi(D_{\text{cmds}}))^2 = \left(\sum_{i=1}^{n-1} \boldsymbol{\lambda_i}\right)^2 =: C_2^2$.*

Lemma 5. *If D is hollow, then $2\|f(D) - f(D_{\text{cmds}})\|_F^2 = \dfrac{n\|(S \circ S)\boldsymbol{\lambda}\|_F^2 - C_2^2}{2} =: C_3$.*

The first term in the error is C_1. From the definition, we can see C_1 is the sum of the squares of the eigenvalues of $\hat{D}$ corresponding to eigenvectors that are not used (or are discarded) in the cMDS embedding. As we increase the embedding dimension, we use more eigenvectors. Hence as the embedding dimension increases, we see that C_1 monotonically decreases. In the case when D is an EDM, we can use all of the eigenvectors so this term will go to zero. On the other hand, if D is not an EDM and $\hat{D}$ has positive eigenvalues (i.e. negative eigenvalues of $-VDV$), then these are eigenvalues that correspond to eigenvectors that cannot be used. Thus, cMDS will always exhibit this phenomenon for C_1 regardless of the input D (for both STRAIN and SSTRESS).

The second term is C_2^2. This term looks similar to C_1, but instead of summing the eigenvalues squared, we first sum the eigenvalues and then take the square. This subtle difference has a big impact. First, we note that as r increases, C_2 becomes more negative. Suppose that D is not an EDM, (i.e., $\hat{D}$ has positive eigenvalues) and let K be the number of negative eigenvalues of $\hat{D}$. Then, since D has trace 0, when $r = K$, C_2 is negative. Hence, C_2 decreases and is eventually negative as r increases which implies that as r increases, there is a a certain value of r after which C_2^2 **increases. This results in the quality of the embedding decreasing. As we will see, this term will be the dominant term in the total error.**

While C_1 and C_2 are simple to understand, C_3 is more obtuse. To simplify it, we consider the following. If δ is an entry of a random n by n unitary matrix, then as n goes to infinity the distribution

for $\delta\sqrt{n}$ converges to $\mathcal{N}(0,1)$ and the total variation between the two distributions is bounded above by $8/(n-2)$ [Easton, 1989, Diaconis and Freedman, 1987]. Therefore, we can assume that the variance of an entry of a random n by n orthogonal matrix is about $1/n$. So, heuristically, $S \circ S \approx \frac{1}{n}11^T$ and

$$n\|(S \circ S)\vec{\lambda}\|_F^2 \approx \frac{n}{n^2}\|11^T\boldsymbol{\lambda}\|_F^2 = \frac{n}{n^2}\|1C_2\|_F^2 = C_2^2.$$

Thus, C_3 is roughly constant.

Algorithm 2 Lower Bound Algorithm.

function LOWER(D, r)
  Compute $\hat{D}$, $f(D)$ and $\xi(D)$.
  Compute $\lambda_1 \leq \ldots \leq \lambda_{n-1}$, U as the eigenvalues and eigenvectors of $\hat{D}$
  Initialize $c_i = \lambda_i$ and $c_n = \xi(D)$
  Let negative_C$_2$ = 0.
  for $i = 1 \ldots n-1$ **do**
    if $\lambda_i > 0$ or $i > r$ **then**
      negative_C$_2$ += λ_i
      Set c_i to 0
  E := number of c's not equal to 0
  sub := negative_C$_2$/E
  for $i = 1 \ldots n-1$ **do**
    if $c_{n-i}! = 0$ **then**
      if $c_{n-i} + \mathrm{sub} \leq 0$ **then**
        c_{n-i} += sub, E −= 1, negative_C$_2$ −= sub
      else
        E −= 1, negative_C$_2$ += $\mathrm{c_{n-1}}$, sub = negative_C$_2$/E
        $c_{n-1} = 0$
  c_n += sub, negative_C$_2$ −= sub
  Let $\hat{D} = U * \mathrm{diag}(c_1, \ldots, c_{n-1}) * U^T$
  return $Q * \begin{bmatrix} \hat{D} & f(D) \\ f(D)^T & c_n \end{bmatrix} * Q$

From the previous discussion, we see that the C_2^2 term in the decomposition is the most vexing due to the term $(\xi(D) - \xi(D_{\mathrm{cmds}}))^2$. This term is from the excess in the trace; that is, the result of discarding the eigenvalues changes the value of the trace from 0 to non-zero. From Lemma 3, we see that cMDS projects $-\hat{D}/2$ onto the cone of PSD matrices (which do not necessarily have trace 0). To retain the trace 0 condition, we perform the following adaptation. Let $\kappa(r)$ be the space of symmetric, trace 0 matrices A, such that $\hat{A}$ is negative semi-definite of rank at most r and project D onto $\kappa(r)$ instead. That is, we seek a solution the following problem.

$$D_l := \underset{A \in \kappa(r)}{\arg\min} \|A - D\|_F^2. \tag{8}$$

Clearly this problem is a strict relaxation of the SSTRESS MDS problem and hence we get that

$$\|D - D_t\|_F \geq \|D_l - D\|_F$$

Because we no longer require the solution to be hollow, conjugating by S permits us to rewrite Problem 8 as:

$$\boldsymbol{c} := \underset{\substack{\boldsymbol{c} \in \mathbb{R}^n, \boldsymbol{c}^T 1 = 0, \\ \forall\, n > i,\ c_i \leq 0, \\ \forall\, n > j > r,\ c_j = 0}}{\arg\min} \sum_{i=1}^{n-1} (c_i - \lambda_i)^2 + (c_n - \xi(D))^2 \tag{9}$$

Theorem 2. *If $\boldsymbol{c}$ is the solution to Problem 9 and D_l is the solution to problem 8 and if we let M be the $n-1$ by $n-1$ diagonal matrix with the first $n-1$ terms of $\boldsymbol{c}$ on the diagonal, then* $S^T D_l S = \begin{bmatrix} M & f(D) \\ f(D)^T & c_n \end{bmatrix}$.

Unlike Problem 3, Problem 9 can be solved directly. We do so via Algorithm 2. We see that in the first loop (lines 6-9), we set c_i to be 0 if $\lambda_i > 0$ or $i > r$ to ensure the proper constraints on c_is (i.e., the eigenvalues). Doing so, we incur a cost of C_1. That is, we sum the squares of those eigenvalues. Now the solution after line 9 does not necessarily satisfy $\boldsymbol{c}^T 1 = 0$. In fact, at this stage of the algorithm, we have that $\boldsymbol{c}^T 1 = C_2$, and we have E many c's that are non zero that we can adjust. That is, we modify these entries on average by C_2/E. Because we use a Frobenius norm to measure error, this incurs a cost of $C_2^2 E/E^2 = C_2^2/E$. Finally, we note that the major computational step of the algorithm is the spectral decomposition. Hence it has a comparable run time to cMDS.

Theorem 3. *If D is a symmetric, hollow matrix, then Algorithm 2 computes D_l the solution to 8.*

Proposition 1. *If D is a symmetric, hollow matrix, then*

$$\|D_l - D\|_F^2 \geq C_1 + \frac{C_2^2}{r+1}$$

It is important to note that D_l is not a metric. However, if we want an EDM and hence an embedding, we can use D_l as the input to cMDS algorithm. Noting that $\hat{D}_l$ is negative semi-definite of rank at most r, we see that if D_{lcmds} is the metric returned by cMDS on input D_l, then we have that $\|D_l - D_{\text{lcmds}}\|_F^2 = 2\|f(D_l) - f(D_{\text{lcmds}})\|_F^2 = 2\|f(D) - f(D_{\text{lcmds}})\|_F^2$. Similar to Lemma 5, we get that

Lemma 6.

$$2\|f(D) - f(D_{\text{lcmds}})\|_F^2 = \frac{n\|(S \circ S)\boldsymbol{c}\|_F^2 - \frac{C_2^2}{r+1}}{2} =: C_4.$$

Then we see that

$$\begin{aligned}\|D_{\text{lcmds}} - D\|_F^2 &\leq \|D_{\text{lcmds}} - D_l\|_F^2 + \|D_l - D\|_F^2 \\ &\leq 2\|f(D_{\text{lcmds}}) - f(D_l)\|_F^2 + \|D_t - D\|_F^2. \\ \Rightarrow \|D_{\text{lcmds}} - D\|_F^2 - \|D_t - D\|_F^2 &\leq \frac{n\|(S \circ S)\boldsymbol{c}\|_F^2 - \frac{C_2^2}{r+1}}{2} \approx constant.\end{aligned}$$

On the other hand, if we compare the cMDS error against the lower bound error from proposition 1. Taking the difference, we get

$$\|D_{\text{cmds}} - D\|_F^2 - \|D_t - D\|_F^2 \leq C_3 + \frac{r}{r+1}C_2^2 \approx \frac{r}{r+1}C_2^2 + constant$$

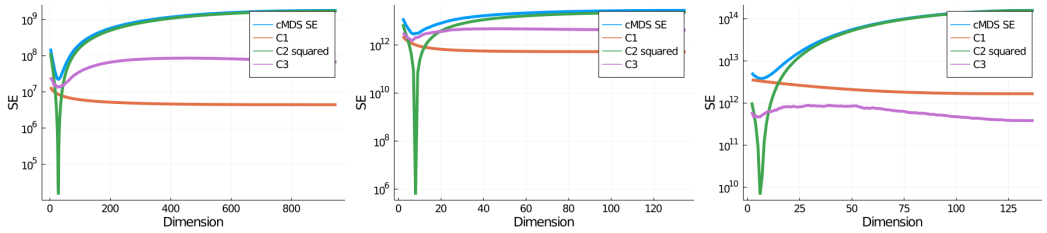
Using our heuristic for C_3, we expect that it is a constant. Similarly, C_4 is a approximately a constant. Thus, comparing the two, we expect D_{cmds} to be be much further from D_t, than D_{lcmds}. These claims are experimentally verified in the next section.

Solving Equation 3. Another approach would be to solve Equation 3 directly. However, solving this problem is much more computationally expensive and works such as Qi and Yuan [2014] present new algorithms to do so. Algorithm 2 can be used to compute the solution as well. Here we Dykstra's method of alternating projections as Hayden and Wells [1988] do, but instead of only projecting $\hat{D}$ on the cone of negative semi definite matrices, without any rank constraint, we project onto $\kappa(r)$ using Algorithm 2. The second, projection is then onto the the space of hollow matrices. Alternating these projections gives us an iterative method to compute D_t.

4 Experiments

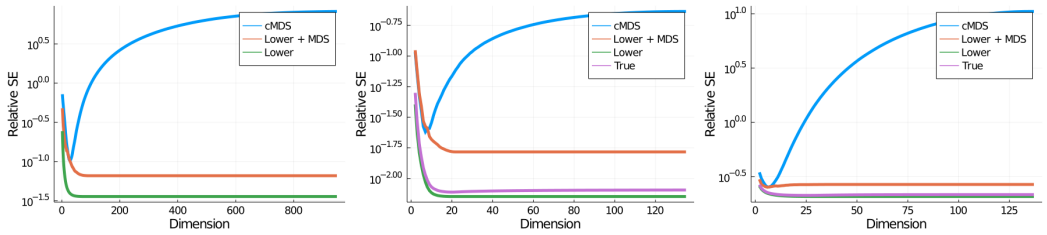
In this section, we do two things. First, we empirically verify all of the theoretical claims. Second, we show that on the downstream task of classification, if we use cMDS to embed the data, then as the embedding dimension increases, the classification accuracy gets worse. Thus, suggesting that the embedding quality of cMDs degrades.

Verifying theoretical claims. To do this, we look at three different types of metrics. First, are metrics that come from graphs and for these we use Celegans Rossi and Ahmed [2015] and Portugal Rozemberczki et al. [2019] datasets. Second, are metrics obtained in an intermediate step of Isomap.

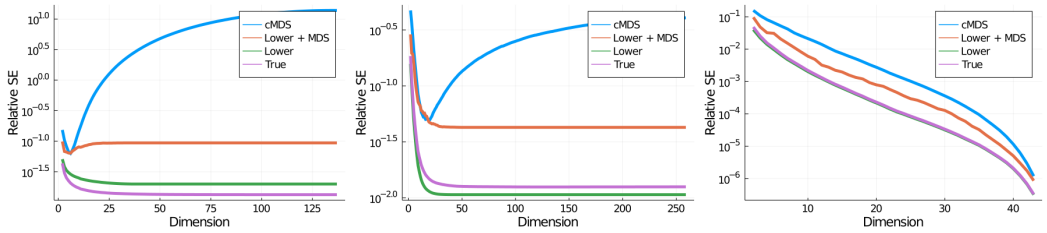


(a) Portugal (b) Isomap with heart (c) Perturbed with SNR 3.24

Figure 1: Plots showing the cMDS error as well as the three terms that we decompose the error into. For the perturbed EDM input, this is error with respect to the perturbed EDM.



(a) Portugal (b) Isomap with Heart (c) Perturbed - Input



(d) Perturbed - Original (e) Celegans (f) Heart

Figure 2: Plots showing the relative squared error of the solutions with respect to the input matrix. For the perturbed EDM input, we show the relative squared error with respect to the original EDM (figure (d)) and the perturbed EDM (figure (c)). For the Portugal data set, we couldn't compute the true solution due to computational restraints.

Third, are perturbations of Euclidean metrics. For both of these metrics we use the heart dataset Detrano et al. [1989]. In all cases, we show that the classical MDS algorithm does not perform as previously expected; instead, it matches our new error analysis. In each case, we demonstrate that our new algorithm Lower + cMDS outperforms cMDS in relation to SSTRESS.

Results. First, let us see what happens when we perturb an EDM. Here we consider two different measures. Let D be the original input, D_p be the perturbed, and D_{cmds} be the cMDS solution. Then we measure

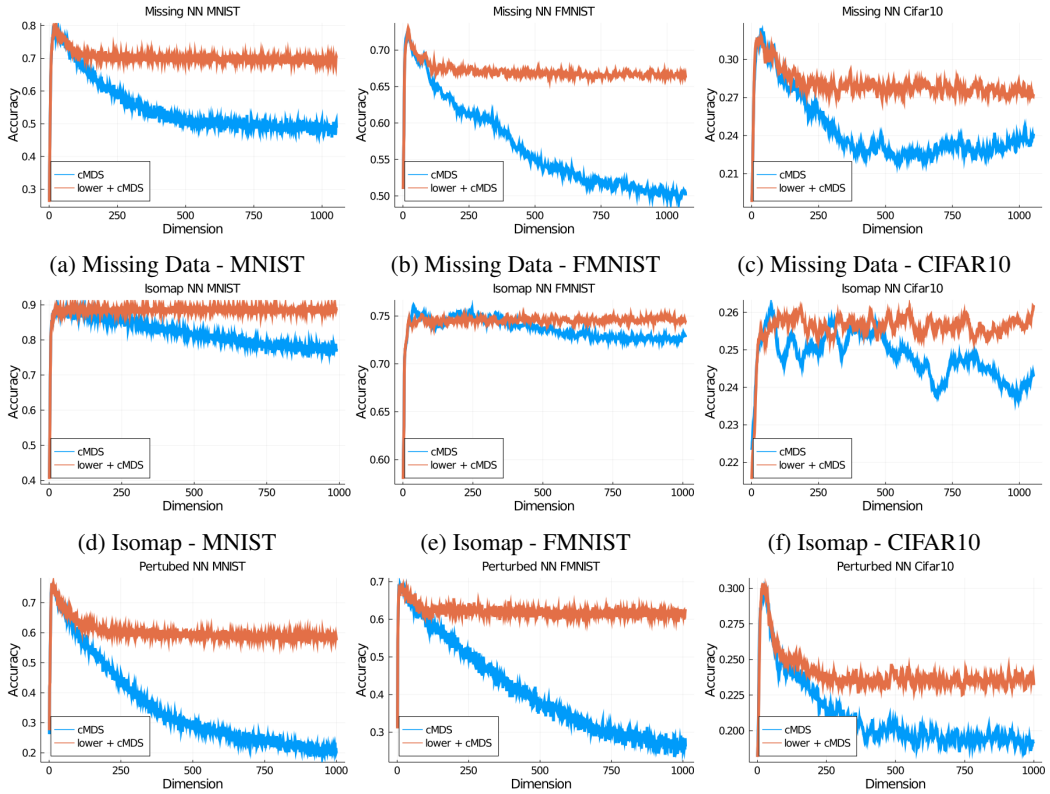
$$\frac{\|D_p - D_{\text{cmds}}\|_F^2}{\|D_p\|_F^2} \quad \text{and} \quad \frac{\|D - D_{\text{cmds}}\|_F^2}{\|D\|_F^2}.$$

As we can see from Figure 2c and 2d, as the embedding dimension increases both quantities eventually increase. The increase in the relative SSTRESS is as we predicted theoretically. The increase in the second quantity suggests that cMDS does not do a good job of denoising either. This suggests that the cMDS objective is not a good one.

Next, we see how cMDS does on graph datasets and with Isomap on Euclidean data. Here we plot the relative squared error ($\|D - D_{\text{cmds}}\|_F^2 / \|D\|_F^2$). Figure 2 shows that, as predicted, as the embedding dimension increases, the cMDS error eventually increases as well. For both the Portugal dataset alone and with Isomap on the heart dataset, this error eventually becomes worse than the error when embedding into two dimensions! As we can see from Figure 1, this increase is **exactly** due to the C_2^2 term. Also, as heuristically predicted, the C_3 term is roughly constant.

Let us see how the true SSTRESS solution and new approximation algorithm perform. We look at the relative squared error again. First, Figure 2 shows us that our lower bound tracks the true SSTRESS solution extremely closely. We can also see from Figure 2d that the true SSTRESS solution and the Lower + cMDS solution are closer to the original EDM compared to the perturbed matrix. Thus, they are better at denoising than cMDS. Finally, we see that our new algorithm Lower + cMDS performs better than just cMDS and, in most cases, fixes the issue of the SSTRESS error increasing with dimension. We also see that $\|D_{\text{lcmds}} - D\|_F^2 - \|D_t - D\|_F^2$ is also roughly constant.

Classification. For the classification tasks, we switch to more standard benchmark datatsets: MNIST, Fashion MNIST, and CIFAR10. If we treat each pixel as a coordinate then these datasets are Euclidean and we cannot demonstrate the issue with cMDS. We obtain non-Euclidean metrics in three ways. First, we compute the Euclidean metric and then perturb it with a symmetric, hollow matrix whose off diagonal entries are drawn from a Gaussian. Second, we construct a k nearest neighbor graph and then compute the all pair shortest path metric on this graph. This is the metric that Isomap tries to embed using cMDS. Third, we imagine each image has missing pixels and then compute the distance between any pair of images using the pixels that they have in common (as done in Balzano et al. [2010], Gilbert and Sonthalia [2018]). Thus, we have 9 different metrics to embed. For each dataset, we constructed all 3 metrics for the first 2000 images. We embedded each of the nine metrics into Euclidean space of dimensions from 1 to 1000 using cMDS and our Lower + cMDS algorithm. We do not embed by solving Equation 3 as this is computationally expensive. For each embedding, we then took the first 1000 images are training points and trained a feed-forward 3 layer neural network to do classification. Results with a nearest neighbor classifier are in the appendix. We then tested the network on the remaining 1000 points. We can see from Figure 3, that as the embedding dimension increases, the classification accuracy drops significantly when trained on the cMDS embeddings. On the other hand, when trained on the Lower + cMDS embeddings, the classification accuracy does not drop, or if it degrades, it degrades significantly less. Thus, showing that the increase in SSTRESS for cMDS on non Euclidean metrics, does result in a degradation of the quality of the embedding and that our new method fixes this issues to a large extent.



(a) Missing Data - MNIST (b) Missing Data - FMNIST (c) Missing Data - CIFAR10

(d) Isomap - MNIST (e) Isomap - FMNIST (f) Isomap - CIFAR10

(g) Perturbed - MNIST (h) Perturbed - FMNIST (i) Perturbed - CIFAR10

Figure 3: Plots showing the classification accuracy for a 3 layer neural network.

References

Phipps Arabie, Mark S Aldenderfer, Douglas Carroll, and Wayne S DeSarbo. *Three Way Scaling: A Guide to Multidimensional Scaling and Clustering*, volume 65. Sage, 1987.

Song Bai, Xiang Bai, Longin Jan Latecki, and Qi Tian. Multidimensional scaling on multiple input distance matrices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Laura Balzano, Robert Nowak, and Waheed Bajwa. Column subset selection with missing data. In *NIPS workshop on low-rank methods for large-scale machine learning*, volume 1. Citeseer, 2010.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

J Douglas Carroll and Phipps Arabie. Multidimensional scaling. *Measurement, judgment and decision making*, pages 179–250, 1998.

J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

Lawrence Cayton and Sanjoy Dasgupta. Robust euclidean embedding. In *Proceedings of the 23rd international conference on machine learning*, pages 169–176, 2006.

Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. doi: 10.1137/0707001. URL `https://doi.org/10.1137/0707001`.

R. Detrano, A. Jánosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64 5:304–10, 1989.

Persi Diaconis and D. Freedman. A dozen de finetti-style results in search of a theory. *Annales De L Institut Henri Poincare-probabilites Et Statistiques*, 23:397–423, 1987.

Daniel B. Dias, R. B. Madeo, Thiago Rocha, H. H. Bíscaro, and S. M. Peres. Hand movement recognition for brazilian sign language: A study using distance-based neural networks. *2009 International Joint Conference on Neural Networks*, pages 697–704, 2009.

Thomas G. Dietterich, A. Jain, R. Lathrop, and Tomas Lozano-Perez. A comparison of dynamic reposing and tangent distance for drug activity prediction. In *NIPS*, 1993.

I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Morris L. Easton. *Chapter 7: Random orthogonal matrices*, volume Volume 1 of *Regional Conference Series in Probability and Statistics*, pages 100–107. Institute of Mathematical Statistics and American Statistical Association, Haywood CA and Alexandria VA, 1989. doi: 10.1214/cbms/1462061037. URL `https://doi.org/10.1214/cbms/1462061037`.

I. Evett and E. Spiehler. Rule induction in forensic science. *Knowledge Based Systems*, pages 152–160, 1989.

Pedro A Forero and Georgios B Giannakis. Sparsity-exploiting robust multidimensional scaling. *IEEE Transactions on Signal Processing*, 60(8):4118–4134, 2012.

Stephen L France and J Douglas Carroll. Two-way multidimensional scaling: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(5):644–661, 2010.

Anna C. Gilbert and Rishi Sonthalia. Unsupervised metric learning in presence of missing data. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 313–321, 2018. doi: 10.1109/ALLERTON.2018.8635955.

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(4):589–600, 1990. ISSN 0895-4798. doi: 10.1137/0611042. URL `https://doi.org/10.1137/0611042`.

R. P. Gorman and T. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.

J. C. Gower. Euclidean distance geometry. *Math. Sci.*, 7(1):1–14, 1982. ISSN 0312-3685.

J.C. Gower. Properties of euclidean and non-euclidean distance matrices. *Linear Algebra and its Applications*, 67:81–97, jun 1985. doi: 10.1016/0024-3795(85)90187-9.

H. A. Guvenir, B. Açar, G. Demiroz, and A. Çekin. A supervised machine learning algorithm for arrhythmia analysis. *Computers in Cardiology 1997*, pages 433–436, 1997.

T.L. Hayden and Jim Wells. Approximation by matrices positive semidefinite on a subspace. *Linear Algebra and its Applications*, 109:115 – 130, 1988. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(88)90202-9. URL `http://www.sciencedirect.com/science/article/pii/0024379588902029`.

Willem J Heiser and Jacqueline Meulman. Constrained multidimensional scaling, including confirmation. *Applied Psychological Measurement*, 7(4):381–404, 1983.

D. Kleinman and M. Athans. The design of suboptimal linear time-varying systems. *IEEE Transactions on Automatic Control*, 13(2):150–159, 1968.

Pieter M Kroonenberg. *Applied multiway data analysis*, volume 702. John Wiley & Sons, 2008.

Monique Laurent. A connection between positive semidefinite and euclidean distance matrix completion problems. *Linear Algebra and its Applications*, 273(1):9 – 22, 1998. ISSN 0024-3795.

Fotios D Mandanas and Constantine L Kotropoulos. Robust multidimensional scaling using a maximum correntropy criterion. *IEEE Transactions on Signal Processing*, 65(4):919–932, 2016.

E. Million. The hadamard product elizabeth million april 12 , 2007 1 introduction and basic results. 2007.

Hou-Duo Qi and Xiaoming Yuan. Computing the nearest euclidean distance matrix with low embedding dimensions. *Mathematical Programming*, 147(1):351–389, 2014.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `http://networkrepository.com`.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding, 2019.

I. J. Schoenberg. Remarks to maurice fréchet's article "sur la définition axiomatique d'une classe d'espaces distanciés vectoriellement applicable sur l'espace de hilbert". annals of mathematics 36(3, 1935.

I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938. ISSN 00029947. URL `http://www.jstor.org/stable/1989894`.

Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962a.

Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27(3):219–246, 1962b.

V. Sigillito, S. Wing, L. Hutton, and K. Baker. Classification of radar returns from the ionosphere using neural networks. 1989.

Y-H Taguchi and Yoshitsugu Oono. Relational patterns of gene expression via non-metric multidimensional scaling analysis. *Bioinformatics*, 21(6):730–740, 2005.

Yoshio Takane, Forrest, W. Young, and Jan De Leeuw. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, pages 7–67, 1977.

J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290 5500:2319–23, 2000.

W. S. Torgerson. Multidimensional scaling i: Theory and method. *Psychometrika*, 17(4):401–419, 1952.

Warren S Torgerson. Theory and methods of scaling. 1958.

Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 04 2014. ISSN 0006-3444. doi: 10.1093/biomet/asv008. URL `https://doi.org/10.1093/biomet/asv008`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Theoretical work, not applicable.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Sections 2 has background. The beginning of Section 3 has the rest of the notation.
   (b) Did you include complete proofs of all theoretical results? [Yes] It is in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] As supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The full set of details are in the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]