On the Selection of Tuning Parameters for Patch-Stitching Embedding Methods

Phong Alain Chau[1] and Ery Arias-Castro[1,2]

[1]Department of Mathematics, University of California, San Diego
[2]Halıcıoğlu Data Science Institute, University of California, San Diego

Abstract

While classical scaling, just like principal component analysis, is parameter-free, other methods for embedding multivariate data require the selection of one or several tuning parameters. This tuning can be difficult due to the unsupervised nature of the situation. We propose a simple, almost obvious, approach to supervise the choice of tuning parameter(s): minimize a notion of stress. We apply this approach to the selection of the patch size in a prototypical patch-stitching embedding method, both in the multidimensional scaling (aka network localization) setting and in the dimensionality reduction (aka manifold learning) setting. In our study, we uncover a new bias–variance tradeoff phenomenon.

1 Introduction

In the general problem known as *multidimensional scaling (MDS)*, the primary objective is to represent a set of items as points within a Euclidean space of a specified dimension. This representation should ideally preserve the given pairwise dissimilarities as accurately as possible, by ensuring that the Euclidean distances between these points mirror the original dissimilarities. MDS is a extensively researched problem found in diverse fields such as psychometrics [16], mathematics, and computer science [9, 14, 57], engineering (where it is also known as *network localization*) [61], as well as statistics [3, 71] and machine learning [43, Ch 14].

Dimensionality reduction (DR) aims at embedding data points in a Euclidean space into a lower-dimensional Euclidean space while preserving, as much as possible, the geometry of the point cloud [36, 59]. When the data points are assumed to be on or near a smooth submanifold, a variant of DR known as *manifold learning*, this typically means preserving the pairwise intrinsic distances to the greatest extent. As is well-known, the two problems, MDS and DR, are closely related.

While classical scaling, and its equivalent in DR, principal component analysis, do not require the choice of a tuning parameter (other than the embedding dimension, whose choice we only discuss in Section 4.1), other methods for embedding data necessitate the specification of one or several parameters, and being the situation unsupervised in that the items (in MDS) or the points (in DR) are not labeled, it is not obvious how to tune these parameters. In fact, we are not aware of any data-driven procedures for choosing such parameters that are currently in use. There is no equivalent to cross-validation — a widely used method for parameter tuning in the context of supervised learning such as regression or classification — that we know of. We propose a simple approach: to use a notion of stress — a measure of quality of fit — to supervise the choice of tuning parameters. We apply this approach to the selection of the patch size in a prototypical patch-stitching embedding method, both in MDS and in DR.

Although the proposed approach this is rather natural, bordering on the obvious, the specter of overfitting might have dissuaded its use. We argue in this paper that there is no danger of overfitting when choosing tuning parameters other than the embedding dimension.

In our investigation, we uncover a new form of bias–variance tradeoff. While such a tradeoff is well-known in regression [43, Ch 7] — where it is foundational in the textbook understanding of statistical complexity and the need to appropriately select method parameters — the discussion of such a phenomenon appears to be absent from the MDS literature in particular. For patch-stitching methods, the choice of patch size needs to balance out how much noise is present in the dissimilarities with how complex the configuration domain is: a high level of noise requires a larger patch size, while a domain that is far from convex requires a smaller patch size. Our numerical experiments show that choosing a patch size that minimizes a notion of stress helps strike a seemingly good compromise when these two aspects need to be balanced out.

The remainder of the paper is organized as follows. In Section 2, we consider patch-stitching methods in the context of MDS. This is where we discuss the bias–variance phenomenon mentioned above. In Section 3, we consider patch-stitching methods in the context of DR. The particular variant that we introduce can be seen as a local form of *Isomap*. We conclude the paper with a brief discussion in Section 4.

2 Multidimensional scaling

In this section we consider multidimensional scaling (MDS). We first formalize the setting in Section 2.1. We then discuss methods in Section 2.2. We use a popular class of methods known under the umbrella name of patch-stitching to showcase the data-driven choice of tuning parameter — here the patch size — by stress minimization, and also the bias–variance tradeoff at play. We present the result of some numerical experiments in Section 2.3 meant to illustrate the proposed approach.

2.1 Setting

In MDS, the data consist of a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, d)$, with node set $\mathcal{V} = [n] := \{1, \dots, n\}$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, together with non-negative weights on the edges. The weight on $(i, j) \in \mathcal{E}$ is referred to as the dissimilarity between i and j, and denoted d_{ij}. The matrix $D = (d_{ij})$, which is incomplete unless the graph is complete, gathers these dissimilarities. Given a dimension $p \geq 1$, we seek a configuration $y_1, \dots, y_n \in \mathbb{R}^p$ such that $\|y_i - y_j\| \approx d_{ij}$ for all or most $(i, j) \in \mathcal{E}$. The problem is further formalized by translating it into an optimization problem that consists in minimization a notion of what is traditionally called *stress* in Psychometrics, for example,

$$\sum_{(i,j)\in\mathcal{E}} \big(\|y_i - y_j\|^2 - d_{ij}^2\big)^2. \tag{2.1}$$

(This stress variant was proposed by Takane et al. [76] and is called the s-stress in the psychometrics literature.)

We will focus on the noisy realizable situation in which

$$d_{ij} = (1 + \eta_{ij})\|x_i - x_j\|, \quad (i, j) \in \mathcal{E}, \tag{2.2}$$

where $\{x_1, \dots, x_n\} \in \mathbb{R}^p$ will be referred to as the *latent configuration* — although it is only determined up to a rigid transformation, as no anchor is assumed available — and $\{\eta_{ij} : (i, j) \in \mathcal{E}\}$

stands for (multiplicative) measurement error. The latent positions will be assumed to be somewhat dense in a subset of $\mathbb{R}^p$ referred to as the latent domain.

Throughout, the dimension p will be assumed given (see Section 4.1) and $\|\cdot\|$ will denote the Euclidean norm in the appropriate space (which will be $\mathbb{R}^p$ in the entire section).

2.2 Methods

Many approaches have been suggested in the literature, including classical scaling [38, 79, 80] and other spectral methods [41]; first-order [54, 55], second-order [48], as well as other Newton and quasi-Newton approaches [37, 49]; augmentation and majorization [27, 44], including the popular *SMACOF* [28, 29, 60]; incremental approaches [18, 21, 85]; semidefinite programs (SDP) [1, 12, 13, 19, 32, 47, 75, 84]; dissimilarity matrix completion by graph distances, including the original *MDS-D* of Kruskal and Seery [56], and its multiple incarnations [65, 66, 73]; and sequential lateration [7, 8, 33, 35, 40, 49, 58].

2.2.1 Patch-stitching methods

We focus on *patch-stitching methods*. These are divide-and-conquer methods that consist in embedding appropriately selected subgraphs as 'patches' in the target Euclidean space and then 'stitching' these patches together by applying a form of Procrustes analysis to align patches that have a large enough intersection. This alignment (aka synchronization) can be done in a greedy manner, by sequentially aligning a new patch with a sufficient overlap with the current embedding; or by a more global approach that attempts to align all patches simultaneously based on all (multiway) intersections. A wide variety of patch-stitching approaches have been proposed [24, 25, 32, 45, 52, 53, 63, 72, 74, 81, 86, 87], motivated by two main reasons. The first reason is computational: the computation of patches can be done in parallel, and the overall procedure can have, depending on the variant, low run time. The second reason is that such methods can work well even when the underlying domain that the latent positions populate has a complex shape. By contrast, for example, methods that rely on shortest path distances like MDS-D, and also some SDP methods like semidefinite embedding of Weinberger et al. [84], can have a substantial bias when the latent domain is far from non-convex.

Shang and Ruml [72] cite both reasons as motivation for their patch-stitching method, *MDS-MAP(P)*, and present them as advantages, in particular as compared to a method they had previously suggested, MDS-MAP [73], which was in fact a rediscovery of MDS-D. MDS-MAP(P) stitches the patches in a greedy fashion: see Figure 2.1 for a visualization of the patches and their sequential stitching.

2.2.2 Tuning by stress minimization

MDS-MAP(P), like any other patch-stitching method, requires the choice of a parameter controlling the patch size. In this particular case, it's the number of hops. A detail the variant that we implemented in Algorithm 1, which is a somewhat simplified variant of the original, which, for example, weighs connections according to the number of hops and uses these weights in the refinement step — while we do not do that. Although we could have used almost any other patch-based method, we adopt MDS-MAP(P) simply to illustrate how to choose that tuning parameter in a data-driven manner.

The approach we propose for choosing one or even several tuning parameters, such as the number of hops in MDS-MAP(P), consists in minimizing a notion of stress. We adopt the variant (2.1) for no particular reason other than it is fairly popular. Although the embedding dimension

Figure 2.1: In the MDS-MAP(P) algorithm, each patch is embedded separately and then merged sequentially. The points in the seed patch are in dark blue, while points added with subsequent patches are more and more red as the stitching progresses. Plotted above are embeddings with number of hops $h \in \{1, 2, 5\}$.

Algorithm 1: A variant of MDS-MAP(P)

Data: weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, D)$, number of hops h, embedding dimension p
Result: configuration $Y = [y_1 \cdots y_n]^\top \in \mathbb{R}^{n \times p}$

for $v \in \mathcal{V}$ **do**
  $N_v \leftarrow h$-hop neighborhood of v;
  $D_v \leftarrow D[N_v, N_v] = (d_{ij} : i, j \in N_v; (i, j) \in \mathcal{E})$;
  $\Lambda_v \leftarrow$ MDS-D applied to D_v;
  $X_v \leftarrow$ classical scaling applied to Λ_v;
  $X_v \leftarrow$ SMACOF applied to X_v;
end
$v^* \leftarrow \arg\max_v \operatorname{card}(N_v)$;
$Y \leftarrow X_{v^*}$;
$N_Y \leftarrow N_{v^*}$;
while *there exist unmapped nodes* **do**
  $v^* \leftarrow \arg\max_v \operatorname{card}(N_v \cap N_Y)$;
  $X_* \leftarrow$ align X_{v^*} to Y by Procrustes;
  $Y \leftarrow Y \cup X_*$;
end
$Y \leftarrow$ SMACOF applied to Y;
return Y;

may be seen as a tuning parameter, minimizing the stress is not an appropriate way to select it, simply because it will always lead to choosing the largest possible embedding dimension, which is $p = n - 1$. See Section 4.1 for further discussion.

Although this approach would seem rather natural, we have not seen it suggested in the literature, where the choice of tuning parameter is often ad hoc or just left to the user. There might be some hesitation to use the stress, as it stands for what is called empirical risk in statistical learning, and minimizing the risk is known to lead to overfitting unless the model complexity is under control. This is particularly true in nonparametric regression. The situation in MDS is not immediately translatable to regression, which is by now well-understood, but we can reason in similar terms. On the one hand, the parameter that we need to estimate is very high dimensional: it is the latent configuration $\{x_1, \ldots, x_n\}$ in (2.2) modulo an arbitrary rigid transformation. Thus, even if the embedding dimension is small, say $p = 2$, the parameter is of $O(n)$ dimension. This needs to be contrasted with the number of observations, which is $|\mathcal{E}|$. It turns out that, as long as the latent configuration is in general position and the graph is connected enough that it is *generically globally*

rigid [23] or even a *lateration graph* [7], minimizing the stress recovers the latent configuration in the noiseless setting ($\eta_{ij} = 0$ for all $(i, j) \in \mathcal{E}$); and although the recovery is no longer exact in the presence of noise, it degrades graceful with the noise level as shown in [2] and [4] in the same situations, respectively. Therefore, minimizing the stress is a reasonable target, and this can be done by all means necessary, as long as the embedding dimension is fixed, since being generically globally rigid or a lateration graph depends in a crucial manner on the dimension p.

2.2.3 Bias–variance tradeoff

In the standard textbook exposition of statistical complexity such as [43, Ch 7], one is taught that, in the context of regression, as the model being fitted to the data increases in complexity, the bias decreases while the variance increases. Model complexity is often driven by one or several tuning parameters (e.g., the bandwidth in kernel regression), and a 'good' selection of these parameters is one that results in a 'good' compromise between (squared) bias and variance, often understood as being equivalent to minimizing the prediction error. In [43, Fig 7.1], we see that, as the model complexity increases, the empirical error (as measured on training data) decreases, while the prediction error (as measured on the test data) decreases at first but then increases — and a 'good' selection of model complexity would be so that the prediction error is at its minimum.

The discussion of such a bias–variance tradeoff, or the choice of model complexity, seems absent from the MDS literature, except for the choice of embedding dimension (see Section 4.1). But it is clearly observed in our experiments involving a non-convex domain, in the case of a hollow rectangle (Figures 2.4–2.5), a C-shaped domain (Figures 2.6–2.7); and an H-shaped domain or 'dumbbell' (Figures 2.8–2.9). Indeed, we can clearly see that, as the number of hops increases, the embedding error decreases and then increases. On the other hand, we observe that the stress does not function as the empirical error does in regression.

In the particular case of MDS-MAP(P), we may explain this as follows. When a domain is non-convex, using a large enough patch that covers the entire domain, MDS-MAP(P) coincides with MDS-D, and MDS-D is known to be biased unless the domain is convex. This is because the shortest path distances are consistent for the intrinsic distances [6, 11], and the intrinsic distances are not Euclidean unless the domain is convex. The choice of a smaller patch size enables MDS-MAP(P) to better avoid that bias, as it only relies on being able to cover the domain with approximately convex balls the size of the patches. Thus the number of hops can be understood as controlling model complexity here: the smaller the number of hops, the smaller the patch size, and the more flexible and therefore complex the domain shape model being implicitly fitted. However, in the presence of noise, one also has to contend with the variance, as the smaller a patch is, the more difficult it may be to accurately embed it.

2.3 Experiments

2.3.1 Synthetic data

We start with some synthetic datasets that exemplify the setting of Section 2.1. All our experiments are in dimension $p = 2$. We first describe how the datasets used in the experiments are constructed. Recall that the latent configuration is denoted by $x_1, \dots, x_n \in \mathbb{R}^2$. In all cases, these points are chosen dense in a domain with varying shape. This is done by considering a fine square grid of points inside the domain to which we add some jitter. The added jitter is small and plays two roles: it makes the configuration generic and it also prevents some possible systematic bias from arising when computing shortest path distances in the graph (which is a building block of MDS-MAP(P)). The jittered grid inside the domain gives the configuration. The graph structure is

given by connecting each point to its $k = 15$ nearest neighbors. The pairwise Euclidean distances between configuration points that are connected in the graph are then corrupted by multiplicative noise as in (2.2), where the η_{ij} are drawn iid from the uniform distribution on $[-\sigma, \sigma]$, where the noise amplitude σ varies from experiment to experiment.

We work with some emblematic shapes: a rectangle in Figures 2.2–2.3; a hollow rectangle in Figures 2.4–2.5; a C-shaped domain in Figures 2.6–2.7; and an H-shaped domain or 'dumbbell' in Figures 2.8–2.9 and also in Figure 2.10 for different noise levels. For each dataset, we apply the variant of MDS-MAP(P) described in Algorithm 1 with different choices for the number of hops and track the (average) stress and the (average) embedding error. We align the output configuration with the true configuration by (orthogonal) Procrustes. We work with rather sparse graphs to better showcase the result of applying MDS-MAP(P) with different choices for the number of hops.

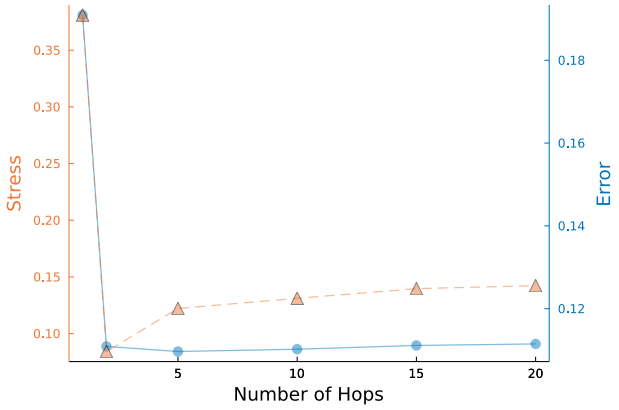

Figure 2.2: Experiment with $n = 4000$ points on a rectangle with $\sigma = 0.15$.

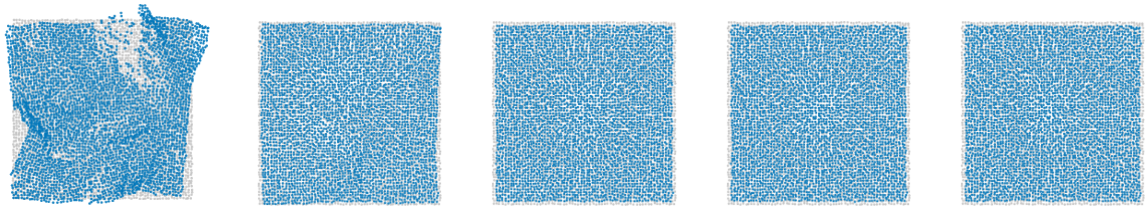

Figure 2.3: Same setting as Figure 2.2. Examples of embeddings with number of hops $h = 1, 2, 3, 5, 10$.

2.3.2 Real data: intercity distances

Besides synthetic datasets, we also examined the application of our approach to the problem of locating cities in a geographical region (California and Texas) using intercity distances. The latitude and longitude of each city are readily available online[1]. The haversine formula is used to construct the observed dissimilarity matrix of as-the-crow-flies distances from the geographical coordinates. That is, if $(\lambda_1, \varphi_1), (\lambda_2, \varphi_2)$ denote the latitude and longitude of a pair of cities, then their distance

[1] For example, at https://simplemaps.com/data/us-cities

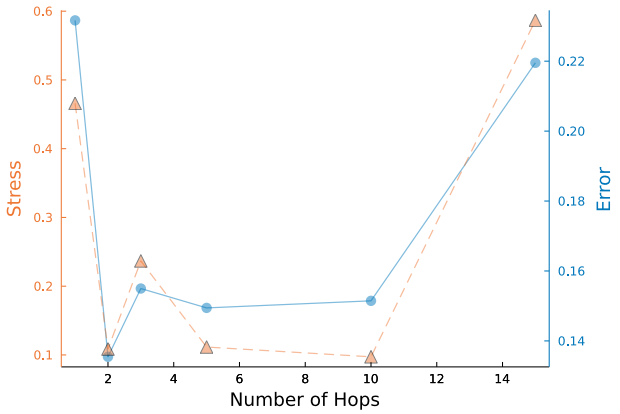

Figure 2.4: Experiment with $n = 4140$ points on a rectangle with a rectangular hole with $\sigma = 0.15$.

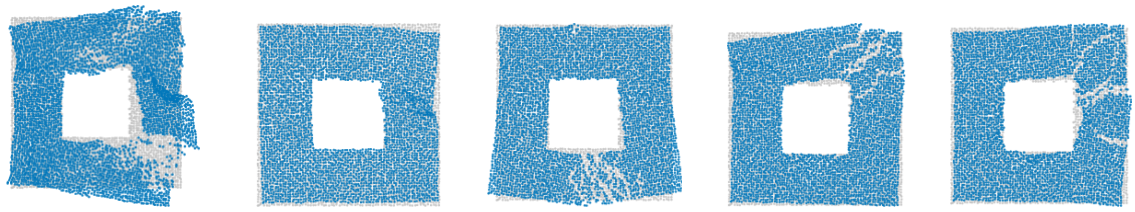

Figure 2.5: Same setting as Figure 2.4. Examples of embeddings with number of hops $h = 1, 2, 3, 5, 10$.

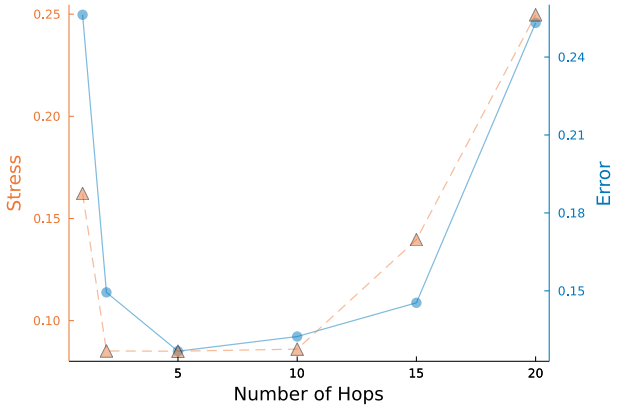

Figure 2.6: Experiment with $n = 4528$ points on a C-shaped domain with $\sigma = 0.15$.

is computed as follows

$$2 \arcsin \sqrt{\sin^2(\tfrac{1}{2}(\varphi_2 - \varphi_1)) + \cos(\varphi_1)\cos(\varphi_2)\sin^2(\tfrac{1}{2}(\lambda_2 - \lambda_1))} \ .$$

We work with the $k = 12$ nearest neighbor graph. Although no noise is added, we note that even without noise an exact realization in the plane is not possible since the points are effectively on a curved surface (the surface of the Earth). Figure 2.11 displays the result of applying MDS-MAP(P) to the intercity distances of California. To illustrate the size of patches for the different choices of number of hops, the patch originating in the capital Sacramento is highlighted. Figure 2.12 shows the result for Texas with the patch originating in the capital Austin being highlighted.

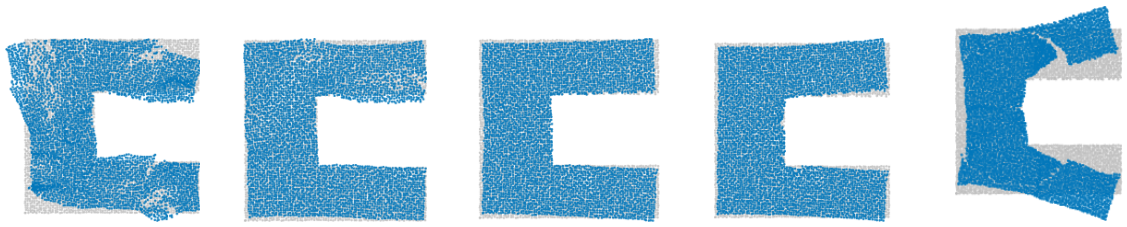

Figure 2.7: Same setting as Figure 2.6. Examples of embeddings with number of hops $h = 1, 2, 5, 10, 20$.

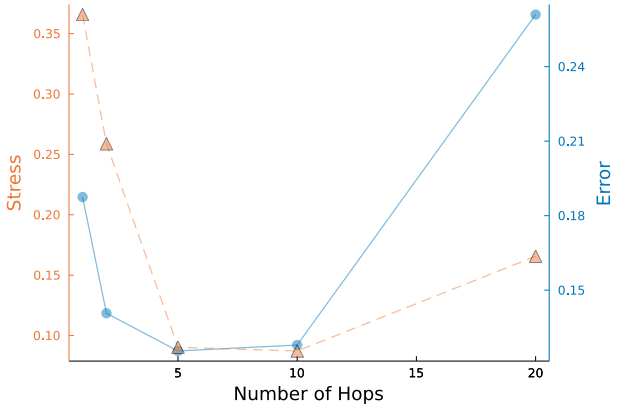

Figure 2.8: Experiment with $n = 4278$ points on an H-shaped domain or 'dumbbell' with $\sigma = 0.15$.

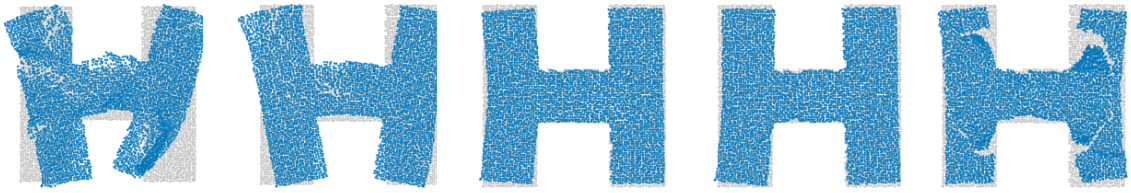

Figure 2.9: Same setting as Figure 2.8. Examples of embeddings with number of hops $h = 1, 2, 5, 10, 20$.

3 Dimensionality reduction

After discussing multidimensional scaling (MDS) in Section 2, we now turn to dimensionality reduction (DR). As is well-known to the expert, the two problems are intimately related. In fact, some of the most emblematic methods in DR can be recovered by applying methods for MDS to the pairwise Euclidean distances after discarding the larger distances. This is most famously true of PCA, whose embedding can be obtained by an application of classical scaling; but it is also true of Isomap [78], which can be obtained in this fashion from MDS-D [56]; of Laplacian eigenmaps [10], which corresponds to applying [41]; of maximum variance unfolding [83], which corresponds to semidefinite embedding [84]; and even recent approaches such as t-SNE [82] and UMAP [62] have been shown to be in correspondence with force-directed layouts popular in graph drawing in [15] and in [26], respectively. Because of this strong parallel, we are able to draw a parallel with the MDS setting discussed in the previous section. The structure of the section is very similar.

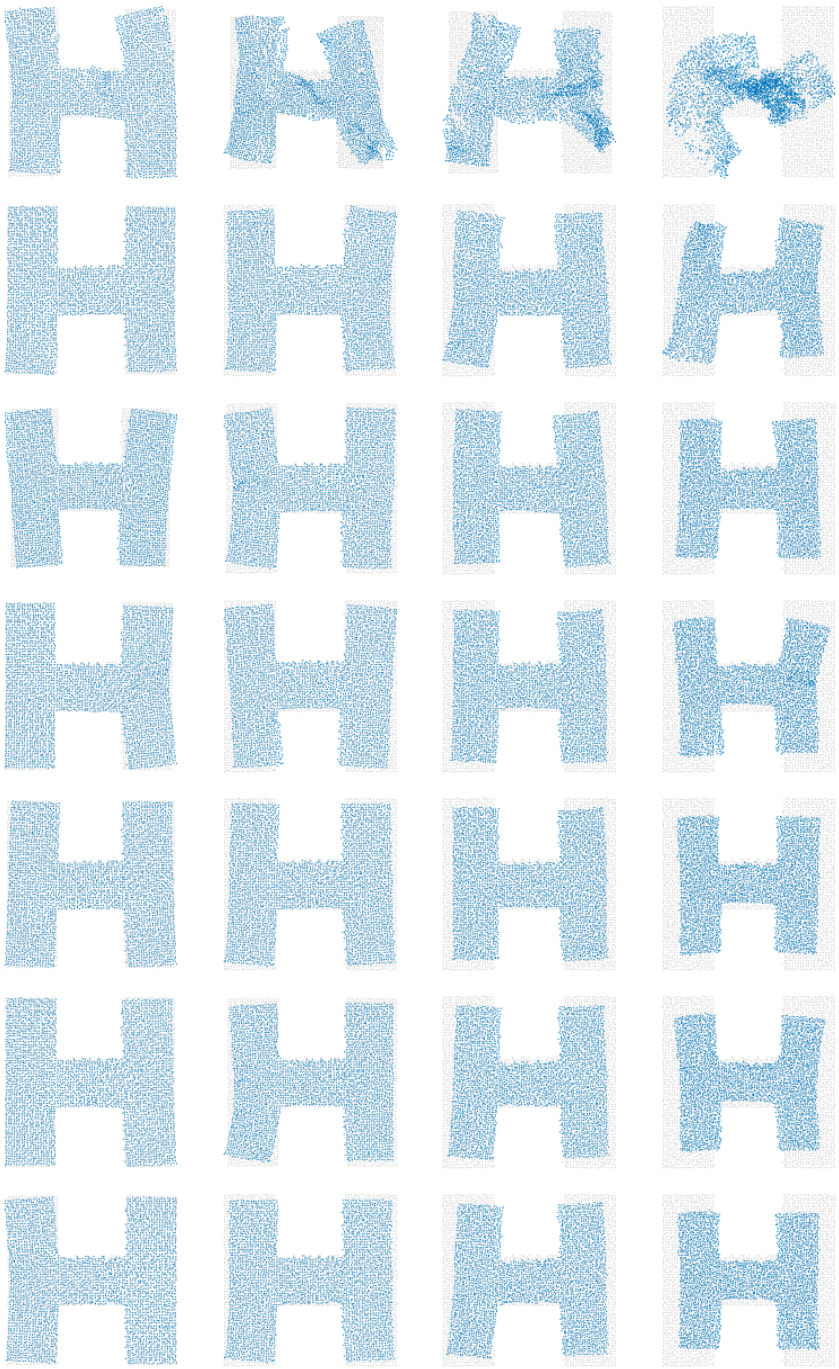

Figure 2.10: In this experiment, we look at the effect of noise on the optimal parameter choice. The dataset features $n = 2866$ points on an H-shaped domain or 'dumbbell'. Figures in the ith column have noise $\sigma = 0.1i$ and the jth row uses number of hops $= j$, for $1 \leq i \leq 4$ and $1 \leq j \leq 7$ integers. As the noise increases, the embeddings tend to shrink. The shrinkage appears to be caused by the under-estimation of some of the distances by graph distances when applying MDS-D.

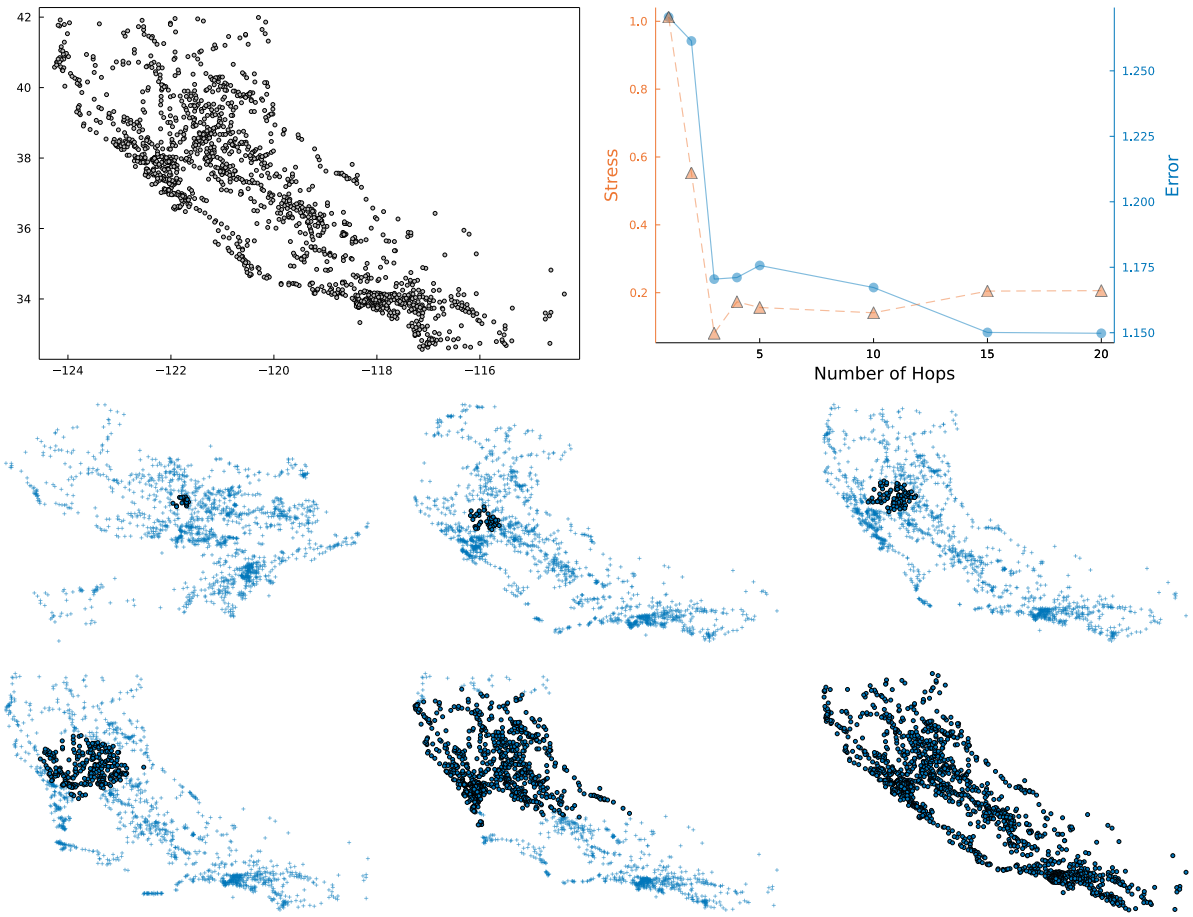

Figure 2.11: MDS-MAP(P) applied to the California intercity dataset. Top left: Plot of cities in California using ground truth latitude and longitude. Top right: Comparison between stress and embedding error. Bottom: In reading order, output of MDS-MAP(P) with number of hops $h = 1, 2, 3, 5, 10, 20$. In each case, the patch originating in Sacramento is highlighted.

3.1 Setting

In DR, the data consist of points $z_1, \ldots, z_n \in \mathbb{R}^{p_0}$, and given a dimension $p < p_0$, the goal is to embed these points into $\mathbb{R}^p$ as faithfully as possible. If by this we mean to preserve the pairwise distances as much as possible, then it can be done by principal component analysis (PCA), which is in fact optimal among linear projections for a particular way of quantifying the accuracy. We adopt the manifold learning setting in which the data points are assumed to be on or near a smooth submanifold of given dimension p and the goal is to preserve as much as possible the pairwise distances on the submanifold. In that case, PCA will not succeed unless the submanifold is affine or nearly so. Most of the DR methods suggested in recent times have been proposed for this setting and, as already noted, can be seen as *(i)* computing the Euclidean distances between the data points, i.e., $d_{ij} := \|z_i - z_j\|$ for all $i, j \in [n]$; *(ii)* only keeping the smallest distances, i.e., for the neighborhood graph with edge set $\mathcal{E} = \{(i, j) : d_{ij} \leq r\}$ where r is the connectivity radius and a tuning parameter; and then applying a method for MDS to the resulting weighted graph. The rationale for only keeping or trusting the smallest Euclidean distances is because, in the limit of an infinitesimally small neighborhood around a point on a smooth submanifold, the Euclidean

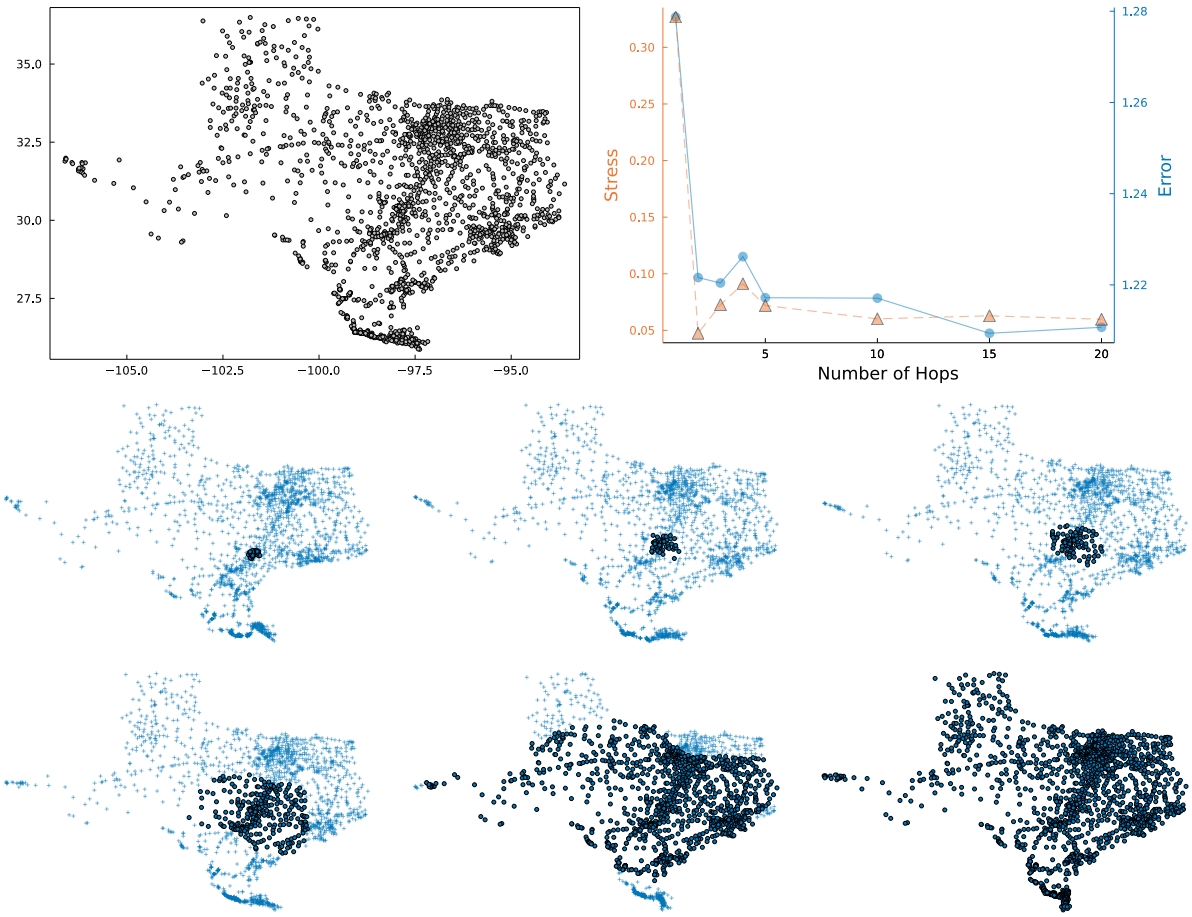

Figure 2.12: MDS-MAP(P) applied to the Texas intercity dataset. Top left: Plot of cities in Texas using ground truth latitude and longitude. Top right: Comparison between stress and embedding error. Bottom: In reading order, output of MDS-MAP(P) with number of hops $h = 1, 2, 3, 5, 10, 20$. In each case, the patch originating in Austin is highlighted.

distances are close to the distances on the submanifold.

3.2 Methods

Manifold learning has a substantial literature. We already mentioned that Isomap [30, 77, 78] is in correspondence with graph distance methods in MDS such as MDS-D [56] and MDS-MAP [73]; Laplacian eigenmaps [10], and the closely related diffusion maps [22], are in correspondence with spectral methods in MDS [41]; maximum variance unfolding [83] is an SDP method that was in fact simultaneously proposed for DR and MDS by the authors [84]; and some of the latest methods, such as t-SNE [82] and UMAP [62], are in correspondence with force-directed approaches in graph drawing [50, Sec 5.7] as argued in [15] and [26]. Not all methods proposed for DR can be derived from a method originally proposed for MDS. For example, self-organizing maps [51], principal surfaces [42], and Kernel PCA [69] approximate the data with a surface of given dimension directly in the ambient space.

Algorithm 2: Local Isomap via MDS-MAP(P)

Data: Data points $z_1, \ldots, z_n \in \mathbb{R}^{p_0}$, connectivity radius r, embedding dimension p
Result: Configuration $y_1, \ldots, y_n \in \mathbb{R}^p$
1 Form the neighborhood graph on $x_1, \ldots, x_n$ with connectivity radius r;
2 Apply MDS-MAP(P) to the resulting weighted graph to obtain an embedding $y_1, \ldots, y_n$;

3.2.1 Patch-stiching methods

Here too, we work with a class of methods that could also be referred to as patch-stitching methods, and work very much in the same way. To quote Brand [17], the protypical steps are "to decompose the sample data into locally linear low-dimensional patches, [and] merge these patches into a single low-dimensional coordinate system". Local linear embedding [67, 68] and manifold charting [17] are clearly of that type, but even more geometrical methods such as Hessian eigenmaps [31] and local tangent space alignment [88] operate in a similar fashion. This parallel has been known for quite some time, at least by some, including Chen and Buja [20], who draw inspiration from the extensive literature on graph drawing, and in particular, force-directed methods, to suggest their local MDS algorithm.

To illustrate the choice of tuning parameter in the context of DR, we simply leverage our variant of MDS-MAP(P) (Algorithm 1) into a method for manifold learning obtaining a local variant of isomap (Algorithm 2). In light of this close connection between MDS and DR, this is a natural idea, which has already been proposed, including in [70].

3.2.2 Tuning by stress minimization

Assuming, as we have done, that the embedding dimension p is given (see Section 4.1 for a discussion), local isomap (Algorithm 2) relies on two tuning parameters: the connectivity radius r in Step 1 and the number of hops h required by MDS-MAP(P) in Step 2.

While we continue to advocate that the number of hops be chosen by minimization of a notion of stress such as (2.1), the choice of connectivity radius r cannot be chosen in the same way for the simple reason that the connectivity radius defines the graph. In our experiments, we follow standard practice and choose r as a small multiple of what is needed for the resulting graph to be connected. We discuss the choice of connectivity radius further in Section 4.2.

3.2.3 Bias–variance tradeoff

A similar manifestation of bias–variance tradeoff as discussed in Section 2.2.3 in the MDS setting is at play in the DR setting when tuning the patch size parameter, and so for similar reasons.

3.3 Experiments

Although one could anticipate that local isomap behaves in the context of DR in a way that is parallel to how MDS-MAP(P) behaves in the context of MDS, we perform some simple numerical experiments to confirm this.

To simulate the data in the manifold learning setting, we start with data points in $\mathbb{R}^2$ as in Section 2.3, and then embed these into $\mathbb{R}^3$. We chose to work with the hollow rectangle of Figure 2.5. Note that here, unlike in the MDS setting, the graph structure is not given but needs to be chosen. This is done in Step 1 of local isomap. We used two different embeddings that seem popular in

the literature: a cylindrical surface based on an S curve and a cylindrical surface based on a spiral, often referred to as a Swiss roll. The 'S' surface is obtained via the following embedding of $[0,1]^2$:

$$\varphi_{S,\alpha}(u,v) = \big(\alpha^{-1}\sin(\alpha v),\ u,\ \alpha^{-1}(\cos(\alpha v)-1)\big); \tag{3.1}$$

the Swiss roll is obtained via the following embedding of $[0,1]^2$:

$$\varphi_{\mathrm{Swiss},\alpha}(u,v) = \big(s(v)\cos(\alpha s(v)),\ u,\ s(v)\sin(\alpha s(v))\big), \tag{3.2}$$

where $s(v)$ is the solution to $\int_0^s \sqrt{1+(\alpha t)^2}\,\mathrm{d}t = v$. The parameter α allows us to increase the curvature of the resulting surface. In our experiments, $\alpha = 10$ for the 'S' surface and $\alpha = 50$ for the Swiss roll. Although the estimation of the local intrinsic distances by the ambient Euclidean distances implies a bias which already plays the role of noise, we add a small amount of Gaussian noise to obtain the data points: see Figure 3.1 for the 'S' surface and Figure 3.2 for the Swiss roll. The result of applying local isomap for various choice of the number of hops is displayed in Figures 3.3–3.4 for the 'S' surface and Figures 3.5–3.6 for the Swiss roll.

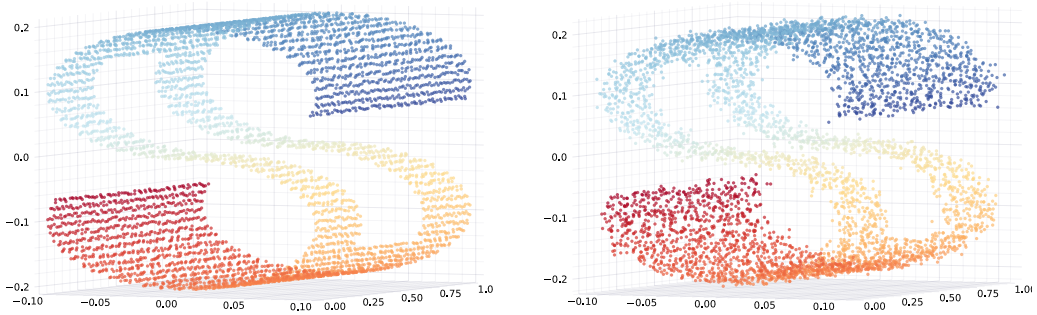

Figure 3.1: Data points (about $n \approx 4000$ of them) generated based on embedding the hollow rectangle of Figure 2.5 as an 'S' surface using (3.1), without (left) and with (right) added noise.

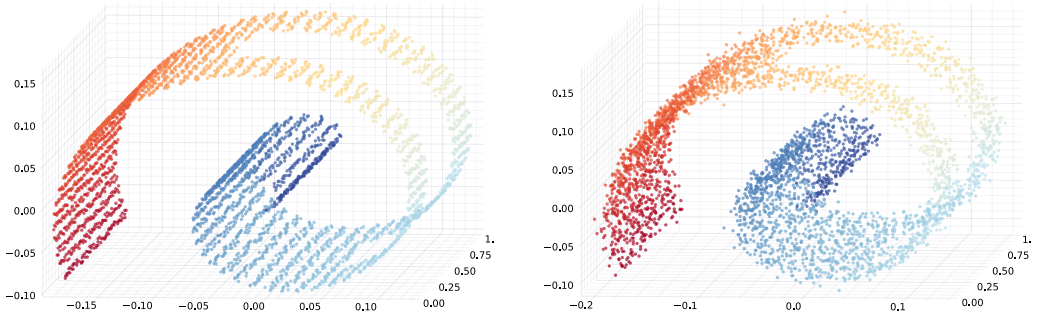

Figure 3.2: Data points (about $n \approx 4000$ of them) generated based on embedding the hollow rectangle of Figure 2.5 as a Swiss roll using (3.2), without (left) and with (right) added noise.

4 Discussion

4.1 Choice of embedding dimension

Some applications, as in sensor network localization where the items are known to be in a 2D physical space, the embedding dimension comes with the problem itself. In other situations, it

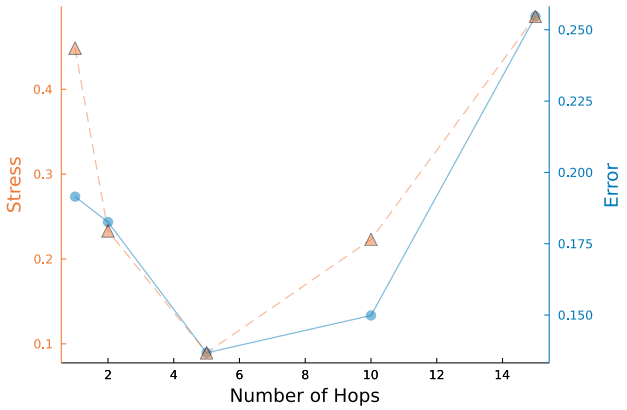

Figure 3.3: Experiment with $n = 5008$ points near an 'S' surface.

Figure 3.4: Same setting as Figure 3.3. Examples of embeddings with number of hops $h = 1, 2, 5, 10, 15$.

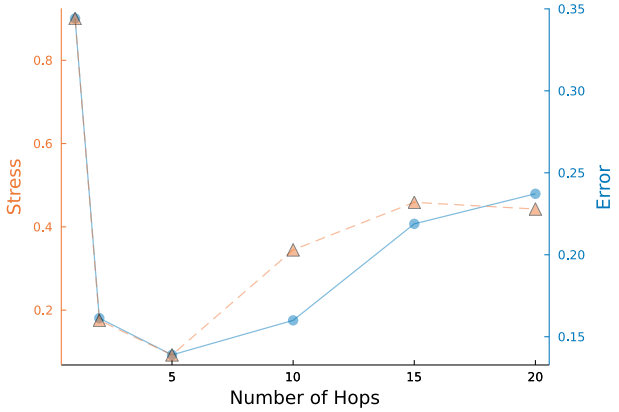

Figure 3.5: Experiment with $n = 4137$ points near an Swiss roll surface.

needs to be chosen by the analyst. In the context of MDS, this is discussed in [16, Sec 3.5], where the suggested approach consists, for a particular method under consideration, in plotting the stress for the output embedding as a function of the embedding dimension, and look for an 'elbow' in the resulting plot indicating that the gains in stress from increasing the dimension have started to dampen. Similar strategies have been suggested in DR, for example, [39], although a number of competing methods have also been proposed — [64, Sec 2.1] provides a partial review.

While this ad hoc approach can be formalized for particular methods (e.g., the one proposed in [39] is shown to be consistent in [5]), we can already see that the situation is very different as

Figure 3.6: Same setting as Figure 3.5. Examples of embeddings with number of hops $h = 2, 5, 10, 15, 20$.

compared to choosing a tuning parameter such as the number of hops for MDS-MAP(P) because the stress is decreasing as a function of the embedding dimension. Consequently, using the stress to choose the embedding dimension is useless as it would result in choosing the largest possible dimension, meaning $p = n - 1$. (n points, even in an infinite-dimensional linear space, are always contained in the affine subspace that they span, which is of dimension at most $n - 1$.) Thus, stress minimization is not a good strategy for choosing the embedding dimension.

4.2 Choice of connectivity radius

Most modern methods in manifold learning rely on a construction of a neighborhood graph, and this necessitates the choice of a connectivity radius.[2] In the literature, the choice of connectivity radius appears to be ad hoc. One of them is a small multiple of what is needed for the resulting graph to be connected, which was our choice in our numerical experiments.

It turns out that the connection with MDS can inform that choice in a more principled manner. Indeed, considerations of rigidity — in that we want the result to be well-defined up to a rigid transformation — would prompt us to choose the connectivity radius a little larger than what is needed for the resulting graph to be generically globally rigid. Actionable, sufficient conditions for that to be true exist. For example, in the important case of $p = 2$, it is enough that the graph be 6-connected [46, Th 7.2], and this can be checked using a variety of algorithms [34].

Acknowledgements

This work was partially supported by the US National Science Foundation (DMS 1916071).

References

[1] Alfakih, A. Y., A. Khandani, and H. Wolkowicz (1999). Solving Euclidean distance matrix completion problems via semidefinite programming. *Computational Optimization and Applications 12*(1), 13–30.

[2] Anderson, B. D., I. Shames, G. Mao, and B. Fidan (2010). Formal theory of noisy sensor network localization. *SIAM Journal on Discrete Mathematics 24*(2), 684–698.

[3] Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis* (3rd ed.). Hoboken: John Wiley and Sons.

[4] Arias-Castro, E. and P. A. Chau (2023). Stability of sequential lateration and of stress minimization in the presence of noise. arXiv preprint arXiv:2310.10900.

[2] While we speak of ball graphs, nearest neighbor graphs are sometimes preferred in practice. But the core issue remains, and simply becomes the choice of the number of nearest neighbors.

[5] Arias-Castro, E., G. Chen, and G. Lerman (2011). Spectral clustering based on local linear approximations. *Electronic Journal of Statistics 5*, 1537–1587.

[6] Arias-Castro, E. and T. Le Gouic (2019). Unconstrained and curvature-constrained shortest-path distances and their approximation. *Discrete & Computational Geometry 62*(1), 1–28.

[7] Aspnes, J., T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur (2006). A theory of network localization. *IEEE Transactions on Mobile Computing 5*(12), 1663–1678.

[8] Bakonyi, M. and C. R. Johnson (1995). The Euclidian distance matrix completion problem. *SIAM Journal on Matrix Analysis and Applications 16*(2), 646–654.

[9] Battista, G. D., P. Eades, R. Tamassia, and I. G. Tollis (1998). *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR.

[10] Belkin, M. and P. Niyogi (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*(16), 1373–1396.

[11] Bernstein, M., V. de Silva, J. Langford, and J. Tenenbaum (2000). Graph approximations to geodesics on embedded manifolds. Technical report, Department of Psychology, Stanford University.

[12] Biswas, P., T.-C. Lian, T.-C. Wang, and Y. Ye (2006). Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN) 2*(2), 188–220.

[13] Biswas, P., T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang (2006). Semidefinite programming approaches for sensor network localization with noisy distance measurements. *Automation Science and Engineering, IEEE Transactions on 3*(4), 360–371.

[14] Blumenthal, L. M. (1953). *Theory and applications of distance geometry*. Oxford University Press.

[15] Böhm, J. N., P. Berens, and D. Kobak (2022). Attraction-repulsion spectrum in neighbor embeddings. *Journal of Machine Learning Research 23*(95), 1–32.

[16] Borg, I. and P. J. Groenen (2005). *Modern multidimensional scaling: Theory and applications*. Springer.

[17] Brand, M. (2003). Charting a manifold. *Advances In Neural Information Processing Systems*, 985–992.

[18] Bronstein, M. M., A. M. Bronstein, R. Kimmel, and I. Yavneh (2006). Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications 13*(2-3), 149–171.

[19] Cayton, L. and S. Dasgupta (2006). Robust euclidean embedding. In *International Conference on Machine Learning*, pp. 169–176.

[20] Chen, L. and A. Buja (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association 104*(485), 209–219.

[21] Cohen, J. D. (1997). Drawing graphs to convey proximity: An incremental arrangement method. *ACM Transactions on Computer-Human Interaction (TOCHI) 4*(3), 197–229.

[22] Coifman, R. and S. Lafon (2006). Diffusion maps. *Applied And Computational Harmonic Analysis 21*(1), 5–30.

[23] Connelly, R. (2005). Generic global rigidity. *Discrete & Computational Geometry 33*(4), 549–563.

[24] Costa, J. A., N. Patwari, and A. O. Hero III (2006). Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks (TOSN) 2*(1), 39–64.

[25] Cucuringu, M., Y. Lipman, and A. Singer (2012). Sensor network localization by eigenvector synchronization over the euclidean group. *ACM Transactions on Sensor Networks (TOSN) 8*(3), 19.

[26] Damrich, S. and F. A. Hamprecht (2021). On umap's true loss function. *Advances in Neural Information Processing Systems 34*, 5798–5809.

[27] de Leeuw, J. (1975). An alternating least squares approach to squared distance scaling. Technical report, Department of Data Theory FSW/RUL.

[28] De Leeuw, J. (1977). Applications of convex analysis to multidimensional scaling. In J. Barra, F. Brodeau, G. Romier, and B. van Cutsem (Eds.), *Recent Developments in Statistics*. North-Holland Publishing Company.

[29] De Leeuw, J. and P. Mair (2009). Multidimensional scaling using majorization: Smacof in r. *Journal of Statistical Software 31*(i03).

[30] de Silva, V. and J. Tenenbaum (2002). Global versus local methods in nonlinear dimensionality reduction. *Advances In Neural Information Processing Systems 15*, 705–712.

[31] Donoho, D. and C. Grimes (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences 100*(10), 5591–5596.

[32] Drusvyatskiy, D., N. Krislock, Y.-L. Voronin, and H. Wolkowicz (2017). Noisy Euclidean distance

realization: Robust facial reduction and the Pareto frontier. *SIAM Journal on Optimization 27*(4), 2301–2331.

[33] Eren, T., O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur (2004). Rigidity, computation, and randomization in network localization. In *Joint Conference of the IEEE Computer and Communications Societies*, Volume 4, pp. 2673–2684.

[34] Esfahanian, A. H. and S. Louis Hakimi (1984). On computing the connectivities of graphs and digraphs. *Networks 14*(2), 355–366.

[35] Fang, J., M. Cao, A. S. Morse, and B. D. Anderson (2009). Sequential localization of sensor networks. *SIAM Journal on Control and Optimization 48*(1), 321–350.

[36] Ghojogh, B., M. Crowley, F. Karray, and A. Ghodsi (2023). *Elements of dimensionality reduction and manifold learning*. Springer Nature.

[37] Glunt, W., T. L. Hayden, and M. Raydan (1993). Molecular conformations from distance matrices. *Journal of Computational Chemistry 14*(1), 114–120.

[38] Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika 53*(3-4), 325–338.

[39] Grassberger, P. and I. Procaccia (1983). Measuring the strangeness of strange attractors. *Physica D 9*, 189–208.

[40] Grone, R., C. R. Johnson, E. M. Sá, and H. Wolkowicz (1984). Positive definite completions of partial Hermitian matrices. *Linear Algebra and its Applications 58*, 109–124.

[41] Hall, K. M. (1970). An r-dimensional quadratic placement algorithm. *Management Science 17*(3), 219–229.

[42] Hastie, T. and W. Stuetzle (1989). Principal curves. *Journal of the American Statistical Association 84*(406), 502–516.

[43] Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009). *The elements of statistical learning: Data mining, inference, and prediction*, Volume 2. Springer.

[44] Heiser, W. J. (1988). Multidimensional scaling with least absolute residuals. *Classification and related methods of data analysis*, 455–462.

[45] Hendrickson, B. (1995). The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization 5*(4), 835–857.

[46] Jackson, B. and T. Jordán (2005). Connected rigidity matroids and unique realizations of graphs. *Journal of Combinatorial Theory, Series B 94*(1), 1–29.

[47] Javanmard, A. and A. Montanari (2013). Localization from incomplete noisy distance measurements. *Foundations of Computational Mathematics 13*(3), 297–345.

[48] Kamada, T. and S. Kawai (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters 31*(1), 7–15.

[49] Kearsley, A. J., R. A. Tapia, and M. W. Trosset (1998). The solution of the metric STRESS and SSTRESS problems in multidimensional scaling using Newton's method. *Computational Statistics 13*(3), 369–396.

[50] Klimenta, M. (2012). *Extending the usability of multidimensional scaling for graph drawing*. Ph. D. thesis, Universität Konstanz.

[51] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics 43*(1), 59–69.

[52] Koren, Y., C. Gotsman, and M. Ben-Chen (2005). PATCHWORK: Efficient localization for sensor networks by distributed global optimization.

[53] Krislock, N. and H. Wolkowicz (2010). Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization 20*(5), 2679–2708.

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

[55] Kruskal, J. B. (1964b). Nonmetric multidimensional scaling: a numerical method. *Psychometrika 29*(2), 115–129.

[56] Kruskal, J. B. and J. B. Seery (1980). Designing network diagrams. In *Conference on Social Graphics*, pp. 22–50.

[57] Laurent, M. (2001a). Matrix completion problems. In *Encyclopedia of Optimization*, pp. 221–229. Springer.

[58] Laurent, M. (2001b). Polynomial instances of the positive semidefinite and Euclidean distance matrix completion problems. *SIAM Journal on Matrix Analysis and Applications 22*(3), 874–894.

[59] Lee, J. and M. Verleysen (2007). *Nonlinear dimensionality reduction*. Information Science and Statistics. Springer New York.

[60] Mair, P., P. J. Groenen, and J. de Leeuw (2022). More on multidimensional scaling and unfolding in R: smacof version 2. *Journal of Statistical Software 102*, 1–47.

[61] Mao, G., B. Fidan, and B. D. Anderson (2007). Wireless sensor network localization techniques. *Computer networks 51*(10), 2529–2553.

[62] McInnes, L., J. Healy, and J. Melville (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

[63] Moore, D., J. Leonard, D. Rus, and S. Teller (2004). Robust distributed network localization with noisy range measurements. In *ACM Conference on Embedded Networked Sensor Systems*, pp. 50–61.

[64] Mordohai, P. and G. Medioni (2010). Dimensionality estimation, manifold learning and function approximation using tensor voting. *Journal of Machine Learning Research 11*(1).

[65] Niculescu, D. and B. Nath (2003). DV based positioning in ad hoc networks. *Telecommunication Systems 22*(1-4), 267–280.

[66] Priyantha, N. B., H. Balakrishnan, E. Demaine, and S. Teller (2003). Anchor-free distributed localization in sensor networks. In *Conference on Embedded Networked Sensor Systems*, pp. 340–341. AMC.

[67] Roweis, S. and L. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science 290*(5500), 2323–2326.

[68] Saul, L. and S. Roweis (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research 4*, 119–155.

[69] Schölkopf, B., A. Smola, and K.-R. Müller (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10*(5), 1299–1319.

[70] Schwartz, A. and R. Talmon (2019). Intrinsic isometric manifold learning with application to localization. *SIAM Journal on Imaging Sciences 12*(3), 1347–1391.

[71] Seber, G. A. (2004). *Multivariate Observations*. John Wiley & Sons.

[72] Shang, Y. and W. Ruml (2004). Improved mds-based localization. In *Conference of the IEEE Computer and Communications Societies*, Volume 4, pp. 2640–2651. IEEE.

[73] Shang, Y., W. Ruml, Y. Zhang, and M. P. Fromherz (2003). Localization from mere connectivity. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 201–212.

[74] Singer, A. (2008). A remark on global positioning from local distances. *Proceedings of the National Academy of Sciences 105*(28), 9507–9511.

[75] So, A. M.-C. and Y. Ye (2007). Theory of semidefinite programming for sensor network localization. *Mathematical Programming 109*(2-3), 367–384.

[76] Takane, Y., F. W. Young, and J. De Leeuw (1977). Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika 42*(1), 7–67.

[77] Tenenbaum, J. (1997). Mapping a manifold of perceptual observations. *Advances in Neural Information Processing Systems 10*.

[78] Tenenbaum, J. B., V. de Silva, and J. C. Langford (2000). A global geometric framework for nonlinear dimensionality reduction. *Science 290*(5500), 2319–2323.

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

[80] Torgerson, W. S. (1958). *Theory and Methods of Scaling*. Wiley.

[81] Tzeng, J., H. H.-S. Lu, and W.-H. Li (2008). Multidimensional scaling for large genomic data sets. *BMC Bioinformatics 9*(1), 1–17.

[82] Van der Maaten, L. and G. Hinton (2008). Visualizing data using t-sne. *Journal of Machine Learning Research 9*(11).

[83] Weinberger, K. Q. and L. K. Saul (2006). An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *National Conference on Artificial Intelligence (AAAI)*, Volume 2, pp. 1683–1686.

[84] Weinberger, K. Q., F. Sha, Q. Zhu, and L. K. Saul (2006). Graph laplacian regularization for large-scale semidefinite programming. In *Advances in Neural Information Processing Systems*, pp. 1489–1496.

[85] Williams, M. and T. Munzner (2004). Steerable, progressive multidimensional scaling. In *Symposium*

on Information Visualization, pp. 57–64. IEEE.

[86] Yang, T., J. Liu, L. McMillan, and W. Wang (2006). A fast approximation to multidimensional scaling. In *Workshop on Computation Intensive Methods for Computer Vision*. ECCV.

[87] Zhang, L., L. Liu, C. Gotsman, and S. J. Gortler (2010). An as-rigid-as-possible approach to sensor network localization. *ACM Transactions on Sensor Networks (TOSN) 6*(4), 35.

[88] Zhang, Z. and H. Zha (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing 26*(1), 313–338.