

Fast Euclidean Embedding of Ordinal Nearest Neighbor Graphs

Brent Castle*

Michael W. Trosset†

Abstract

A number of interesting data sets provide information about the nearest neighbors of a set of objects. For example, each book sold by `Amazon.com` is accompanied by a list of its most similar neighbors, measured by the proportion of viewers who ultimately buy the other book. These data can be conceived as a directed graph in which vertices represent objects and edges identify nearest neighbor structure. For `Amazon.com`, the edges are weighted by a quantitative measure of proximity; in other applications, the edge weights may be ordinal, i.e., the ranks of the nearest neighbors may be provided without a corresponding quantitative measure of proximity. We develop methods for mapping such data sets, i.e., for embedding ordinal nearest neighbor graphs in low-dimensional Euclidean spaces. We borrow ideas from nonmetric multidimensional scaling (MDS); however, the graphs in question are typically too large and too sparse for traditional nonmetric MDS to succeed.

Key Words: graph layout; graph embedding; nonmetric multidimensional scaling; raw stress criterion; partial ordering; isotonic regression

1. Introduction

A number of interesting data sets provide information about the nearest neighbors of a set of objects. For example, each book sold by `Amazon.com` is accompanied by a list of its most similar neighbors, measured by the proportion of buyers who ultimately also buy the other book. Likewise, entities such as Netflix and internet radio site `Last.fm` make recommendations based on nearest neighbors, determined in various ways. In many such applications, information is available (or reliable) for only a small set of nearest neighbors; furthermore, information about proximity may be ordinal, or measured on a scale that necessitates an unspecified monotonic transformation.

The data sets of interest can be conceived as directed graphs in which vertices represent objects and edges identify nearest neighbor structure. In order to visualize such data, we consider the problem of how to represent a sparse directed graph with ordinal edge weights in a low-dimensional Euclidean space. Such problems are the concern of graph layout and/or graph embedding, distinguished by the latter's greater concern that Euclidean distances between points in the representation should approximate pairwise dissimilarities of objects.

The problem of embedding graphs that describe local structure arises elsewhere. In computational chemistry, one may attempt to determine a molecule's 3-dimensional structure from information about its atomic bond lengths and angles. In manifold learning, techniques such as ISOMAP [10] and Laplacian eigenmaps [1] construct and embed graphs that describe the local Euclidean structure of curved surfaces. The edges of these (undirected) graphs may or may not be weighted, but the problem of embedding graphs with ordinal edge weights has not been considered.

*School of Informatics & Computing and Department of Statistics, Indiana University, 309 N Park Ave, Bloomington, IN 47408. Email: bscastle@cs.indiana.edu

†Department of Statistics and School of Informatics & Computing, Indiana University, 309 N Park Ave, Bloomington, IN 47408. Email: mtrosset@indiana.edu

In statistics and psychometrics, techniques for embedding pairwise proximity data are called multidimensional scaling (MDS) and techniques that use only ordinal information about the proximities are called nonmetric MDS. There is a vast literature on nonmetric MDS, but nonmetric MDS usually embeds a complete set of pairwise proximity ranks. In contrast, scant attention has been paid to problems in which only local proximity information is available. A recent exception is the Local MDS method of Chen and Buja [3], inspired in part by algorithms for graph layout. Although Local MDS assumes the existence of metric proximities, it most nearly embodies the ideas described herein.

2. Embedding

Consider a directed graph $G = (V, E)$ with vertices $V = \{1, \dots, n\}$. For each vertex i , assume that the edges $(i, j) \in E$ can be ranked from smallest to largest and let r_{ij} denote the rank of edge (i, j) . If $r_{ij_1} = 1$, then we say that vertex j_1 is the nearest neighbor of vertex i ; if $r_{ij_2} = 2$, then we say that vertex j_2 is the second nearest neighbor of vertex i ; etc. For simplicity, we assume that all ties have been broken. For convenience, we then set $r_{ij} = \infty$ if $(i, j) \notin E$.

Let d denote the dimension of the Euclidean space in which we seek to embed G . Ideally, we would like to find a set of points $x_1, \dots, x_n \in \mathfrak{R}^d$ with the following property:

$$(I) \text{ If } r_{ij} < r_{ik}, \text{ then } d_{ij} = \|x_i - x_j\| \leq \|x_i - x_k\| = d_{ik}.$$

Henceforth, we denote a possible configuration of n points in \mathfrak{R}^d by X and the interpoint distances for X by $d_{ij}(X)$. We would like to find an X whose interpoint distances satisfy a prescribed partial ordering.

Several considerations affect our formulation of the above embedding problem.

1. Existence In practice, it is often the case that no X has interpoint distances for which the desired partial ordering is fully satisfied. This consideration obliges us to formulate an error criterion that quantifies the suitability of each X , then search for X that minimize the error criterion.

2. Asymmetric Information Euclidean distances are necessarily symmetric: $d_{ij}(X) = d_{ji}(X)$. In contrast, nearest neighbor graphs are typically asymmetric, i.e., one usually finds that $r_{ij} \neq r_{ji}$. A reasonable error criterion should accommodate this asymmetry.

3. Local Structure The graphs that we are attempting to embed are sparsely connected. In particular, we only have information about a small number of nearest neighbors for each object. Graef and Spence [6] demonstrated that the pattern of sparsity plays a crucial role in determining the quality of an embedding. In fact, their simulations indicate that nearest neighbor proximities are less important than non-neighbor proximities for determining global structure. (This revelation is hardly surprising: consider, for example, that the chemical structure of a molecule may be consistent with a number of different 3-dimensional structures.) In consequence, it is important to use what little global information can be inferred, viz., that non-neighbors of vertex i are less like i than are neighbors of i . Nevertheless, we should not expect too much from any embedding method. As with graph layout, a variety of configurations may be deemed of comparable quality.

4. Ordinal Constraints Traditional formulations of nonmetric MDS, including Kruskal’s [8] seminal contribution, incorporate a complete ordering of $m = n(n - 1)/2$ pairwise proximities into a scale-invariant error criterion. The cost of isotonic regression is $O(m) = O(n^2)$. Scale invariance precludes degenerate solutions. In contrast, Trosset [12, 11] demonstrated that the objectives of nonmetric MDS can be achieved by enforcing explicit order and nondegeneracy constraints. For general partial orderings, the cost of Burdakov’s [2] algorithm for isotonic regression is $O(m^2) = O(n^4)$.

5. Scalable Algorithms As with Amazon.com, nearest neighbor graphs are especially popular when n is quite large. Nonmetric MDS was developed for situations in which n is fairly small, no more than several hundred. To embed much larger graphs with ordinal weights, we combine several ideas. We construct inexpensive initial configurations. We adopt an error criterion that can be decreased by an inexpensive fixed point method, thereby avoiding the usual costs of numerical optimization. We use specific partial orderings that allow us to use fast algorithms for isotonic regression. We stop after a fixed number of iterations, without monitoring convergence to a minimizer of our error criterion.

3. Optimization

We seek a configuration that, in some sense, captures the local structure of a nearest neighbor graph. Often, the only configurations that satisfy property (I) are degenerate in the sense that they place each point at the same location, so that each $d_{ij}(X) = 0$. We might search for a nondegenerate X that violates property (I) as rarely as possible, but the resulting optimization problem seems intractable. Instead of counting how often the prescribed partial ordering is violated, we measure the magnitude of the violations.

Given $x_1, \dots, x_n \in \mathfrak{R}^d$ and $j \neq i$, let $r_{ij}(X)$ denote the rank of $d_{ij}(X)$ in $\{d_{ik}(X) : k \neq i\}$. For example, if x_j is x_i ’s second nearest neighbor, then $r_{ij}(X) = 2$. It is natural to seek small values of the following error criterion:

$$M_1(X) = \frac{1}{|E|} \sum_{(i,j) \in E} |r_{ij}(X) - r_{ij}|.$$

Alternatively, Chen and Buja [3] counted how many nearest neighbor pairs in G are still nearest neighbor pairs in X . Let $E_k(X)$ denote the edges of the nearest neighbor graph constructed from X . Then, after scaling by the number of edges in G , the Chen-Buja criterion is

$$M_2(X) = \frac{|E_k(X) \cap E|}{|E|}.$$

This criterion seems most appropriate when G is unweighted.

Both M_1 and M_2 are intuitively appealing error criteria, but neither appears to be amenable to direct optimization. Although we will monitor these quantities, our searches will be guided by solving a more tractable optimization problem that was inspired by certain formulations of nonmetric MDS.

Recall that a matrix, $\Delta = [\delta_{ij}]$, is nonnegative iff each entry $\delta_{ij} \geq 0$ and hollow iff each diagonal entry $\delta_{ii} = 0$. Let $\mathcal{M}(G)$ denote the polyhedral cone of hollow nonnegative matrices that satisfy the partial ordering induced by G , i.e., if $r_{ij} < r_{ik}$, then $\delta_{ij} \leq \delta_{ik}$.

Given $x_1, \dots, x_n \in \mathfrak{R}^d$, let $D(X) = [d_{ij}(X)]$. We seek an X for which $D(X)$ is near $\mathcal{M}(G)$; however, because $[0] \in \mathcal{M}(G)$, we must be careful to avoid degenerate solutions. One possible antidote is as follows. For some $c > 0$, let

$$\mathcal{N} = \left\{ \Delta : \sum_i \sum_j \delta_{ij}^2 \geq c \right\}$$

and seek an X for which $D(X)$ is near $\mathcal{M}(G) \cap \mathcal{N}$.

We measure the discrepancy between $D(X)$ and $\Delta \in \mathcal{M}(G) \cap \mathcal{N}$ by an asymmetrically weighted version of Kruskal's [8] raw stress criterion:

$$\sigma_a(\Delta, X) = \frac{1}{2} \sum_i \sum_j w_{ij} [d_{ij}(X) - \delta_{ij}]^2.$$

We emphasize that we require neither $\delta_{ij} = \delta_{ji}$ nor $w_{ij} = w_{ji}$.

The choice of weights is crucial. We restrict attention to $w_{ij} \in \{0, 1\}$ and consider three possible weighting schemes.

1. $w_{ij} = 1$ iff $(i, j) \in E$

This choice corresponds to using exactly the information contained in the nearest neighbor graph. Although it is conceptually attractive, the information is so limited that one rarely obtains a satisfying configuration. In numerical experiments with metric MDS, Graef and Spence [6] found that such schemes often lead to undesirable configurations. In our experience, the problem is just as acute with ordinal information. The difficulty is that local information is rarely sufficient to infer a plausible global structure.

2. $w_{ij} = 1$ for all i, j

This choice is the usual raw stress criterion. In the present context, it corresponds to using all of the information contained in the nearest neighbor graph plus all of the additional information that can be inferred from it, i.e.,

$$\text{if } (i, j) \notin E \text{ and } (i, k) \in E, \text{ then } \delta_{ij} \geq \delta_{ik}.$$

The problem with this choice is that of too much of a good thing. Including so many non-edge pairs causes σ_a to undervalue the edge; furthermore, for large n , the expense of embedding $n(n-1)/2$ pairs may be prohibitive.

3. $w_{ij} = 1$ for all $(i, j) \in E$ and some additional $(i, j) \notin E$

This choice is a compromise between the first two. We randomly select a small set of non-edge pairs and include them in σ_a . The additional information tends to produce more plausible configurations than are typically obtained from just the edge pairs. We denote the augmented set of pairs by E^+ .

Combining the above, we obtain the following optimization problem in the decision variables (Δ, X) :

$$\begin{aligned} & \text{minimize} && \sigma_a(\Delta, X) && (1) \\ & \text{subject to} && \Delta \in \mathcal{M}(G) \cap \mathcal{N} \end{aligned}$$

Although (1) is a tractable optimization problem and algorithms for solving it tend to produce plausible embeddings, we emphasize that configurations with

vastly different local structure may have comparable values of σ_a . Consider Figure 1, which caricatures the problem of inferring molecular conformation from chemical structure. Within the subsets defined by shape/color, the two configurations have identical local structure. The global structure is evidently quite different, as is the local structure between the subsets. If $w_{ij} = 1$ for pairs within the same subset and $w_{ij} = 0$ for pairs between subsets, then σ_a cannot distinguish these configurations. Notice, however, that these configurations do have different values of M_1 and M_2 . We will further explore the relationship between these error criteria in Section 5.2.

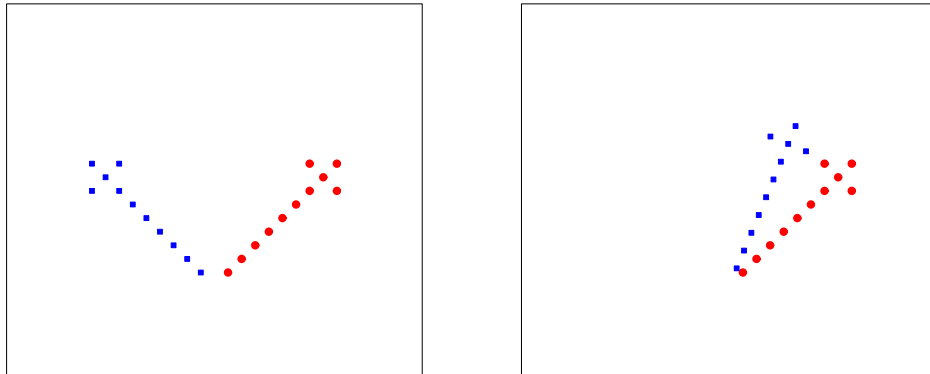


Figure 1: Identical σ_a , different M_1 and M_2 .

4. The NNScal Algorithm

Given (Δ_c, X_c) , we can find a better (Δ_+, X_+) by first fixing Δ_c and finding a better X_+ , then fixing X_+ and finding a better Δ_+ . Doing so is an example of *variable alternation*. To adopt this strategy, it is necessary to construct an initial (Δ, X) . We construct an initial $\Delta = [\delta_{ij}]$ by fixing $\hat{r} > \max(\{r_{ij} : r_{ij} < \infty\})$, then setting $\delta_{ij} = \min(r_{ij}, \hat{r})$. To construct an initial X , we first symmetrize Δ , setting $\delta_{ij} = \delta_{ji} = \min(\delta_{ij}, \delta_{ji})$, then embed Δ_{sym} by metric MDS to obtain an initial X . Here is the complete NNScal (Nearest Neighbor SCALing) algorithm:

1. Specify \hat{r} , e.g., $\hat{r} = 2 \max(\{r_{ij} : r_{ij} < \infty\})$.
2. Set $\Delta = [\min(r_{ij}, \hat{r})]$ and $\Delta_{\text{sym}} = [\min(r_{ij}, r_{ji}, \hat{r})]$.
3. Construct an initial configuration X by embedding Δ_{sym} .
4. Repeat until convergence:
 - (a) Fix Δ and minimize σ_a with respect to X by asymmetric stress Majorization.
 - (b) Fix X and minimize σ_a with respect to $\Delta \in \mathcal{M}(G)$ by isotonic regression.
 - (c) Scale Δ to lie in $\mathcal{M}(G) \cap \mathcal{N}$.

As stated, the NNScal algorithm converges to stationary points of (1). In practice, we only perform several iterations of asymmetric stress majorization, obtaining an improved but suboptimal X . Furthermore, we do not repeat the variable alternation cycle until convergence. Several iterations usually suffice to obtain a reasonable embedding, and that may be all that we can afford if G is large.

The remainder of this section explicates various details of NNScal.

4.1 Initial Configuration

The objective function σ_a is not convex with respect to (Δ, X) . It may have many nonglobal minimizers; hence, the quality of the embedding returned by NNScal is sensitive to the choice of initial (Δ, X) . Given Δ_{sym} , one might construct X by classical MDS. This is a common way to construct an initial configuration from which to begin minimizing the raw stress criterion, but it requires computing the spectral decomposition of an $n \times n$ matrix. This computation is prohibitively expensive when n is large. One can dramatically decrease the computational expense of the initial embedding through techniques such as Landmark MDS [4], FastMap [5], and Metric Map [14], each of which can be viewed as a crude Nyström approximation of the eigenvalues and eigenvectors used in classical MDS [9]. However, these techniques may produce configurations of low quality and they do not exploit the special structure of nearest neighbor graphs. We leave the problem of how to do so for future research. In Section 5, we start NNScal with initial configurations obtained by classical MDS with imputed \hat{r} (IC1) and generated at random (IC2).

4.2 Asymmetric Stress Majorization

Once a configuration has been obtained, Step 4a improves upon it by fixing Δ and minimizing σ_a with respect to X .

For $i < j$, define square matrices A_{ij} by setting the ii and jj entries equal to 1, the ij and ji entries equal to -1 , and all other entries equal to 0. Kruskal's raw stress criterion,

$$\begin{aligned}\sigma(\Delta, X) &= \sum_{i < j} w_{ij} (\delta_{ij} - d_{ij}(X))^2 \\ &= \eta_\delta^2 + \text{tr}(X^T V X) - 2\text{tr}(X^T B(X) X),\end{aligned}$$

where $V = \sum_{i < j} w_{ij} A_{ij}$ and $B(X) = \sum_{i < j} \frac{w_{ij} \delta_{ij}}{d_{ij}} A_{ij}$, was intended for use with symmetric Δ . We seek to minimize the asymmetric raw stress criterion:

$$\begin{aligned}\sigma_a(\Delta, X) &= \frac{1}{2} \sum_i \sum_j w_{ij} (\delta_{ij} - d_{ij}(X))^2 \\ &= \frac{1}{2} [\eta_\delta^2 + \text{tr}(X^T V_a X) - 2\text{tr}(X^T B_a(X) X)],\end{aligned}$$

where $V_a = \sum_{i < j} (w_{ij} + w_{ji}) A_{ij}$ and $B_a(X) = \sum_{i < j} \frac{w_{ij} \delta_{ij} + w_{ji} \delta_{ji}}{d_{ij}} A_{ij}$.

For fixed Δ , both criteria can be majorized by functions of the same form. For σ_a , the Diagonal Majorization Algorithm (DMA) of Trosset and Groenen [13] updates the configuration as follows:

$$X \leftarrow X + \frac{1}{2} \text{diag}(V_a)^{-1} [B_a(X) - V_a] X.$$

For nearest neighbor graphs, the update can also be computed as

$$x_i \leftarrow x_i + \frac{1}{2\gamma(i)} \left[\sum_{(i,j) \in E^+} \left(\left(\frac{\delta_{ij}}{d_{ij}} - 1 \right) (x_i - x_j) \right) + \sum_{(j,i) \in E^+} \left(\left(\frac{\delta_{ij}}{d_{ij}} - 1 \right) (x_j - x_i) \right) \right],$$

where $\gamma(i)$ is the in-degree plus the out-degree of vertex v_i . The second update formula is evaluated by looping through the edges in E^+ and updating x_i for each (i, j) and (j, i) pair. The first update formula requires $O(dn^2)$ operations, whereas the second update formula requires only $O(nkd)$ operations.

For fixed Δ , repeated iterations of DMA will converge to a stationary configuration of σ_a . In practice, we do not monitor a convergence criterion in Step 4a, but simply perform a fixed number of iterations (typically 4 or 5) before proceeding to Step 4b.

4.3 Isotonic Regression

Step 4b of NNScal fixes X and updates Δ subject to $\Delta \in \mathcal{M}(G)$. The unique minimizer of this constrained least squares problem can be computed efficiently. The partial ordering induced by the constraints is a set of n distinct, complete orderings. Each of the n subproblems has a predefined rank ordering for the nearest neighbors and no predefined ranking for the additional pairs in $E^+ - E$. Without loss of generality, consider the case where $i = n$ and v_i 's k nearest neighbors, in order, are v_1, v_2, \dots, v_k . There is also an additional set of m edges in E^+ that are directed edges from v_i to $v_{k+1}, v_{k+2}, \dots, v_{k+m}$. Then, for a fixed X where $D(X) = [d_{ij}]$, we seek solutions of:

$$\begin{aligned} \min_{\delta_{i1}, \dots, \delta_{i, k+m}} \quad & \frac{1}{2} \sum_{j=1}^{k+m} w_{ij} (\delta_{ij} - d_{ij})^2 \\ \text{s.t.} \quad & \delta_{i1} \leq \delta_{i2} \leq \dots \leq \delta_{ik} \preceq \{\delta_{n, k+1}, \delta_{n, k+2}, \dots, \delta_{n, k+m}\}. \end{aligned}$$

Here, the \preceq means that δ_{ik} is entry-wise less than each element of the succeeding set.

To perform this optimization problem we first sort the non-neighbor distances from the configuration X with non-zeros weights. Again, without loss of generality let us assume that the distances are $d_{i, k+1}, \dots, d_{i, k+m}$ and satisfy $d_{i, k+1} \leq d_{i, k+2} \leq \dots \leq d_{i, k+m}$. Then we can solve the completely ordered isotonic regression problem:

$$\begin{aligned} \min_{\delta_{i1}, \dots, \delta_{i, k+m}} \quad & \frac{1}{2} \sum_{j=1}^{k+m} w_{ij} (\delta_{ij} - d_{ij})^2 \\ \text{s.t.} \quad & \delta_{i1} \leq \delta_{i2} \leq \dots \leq \delta_{i, k+m}. \end{aligned}$$

We perform n completely ordered isotonic regressions, one for each i , using the algorithm in [7]. Since each isotonic regression requires a sort to determine the

ordinal constraints with an average case computation time $O(m \log m)$, an isotonic regression that requires $O(k + m)$ time, and it is assumed that $m \in \Omega(k)$ then the resulting average case computation time is $O(nm \log m)$.

4.4 Precluding Degenerate Solutions

Without the final constraint in (1) an optimal solution would be $\Delta = 0$ and $X = 0$. Clearly, this solution is undesirable. In [12], it was proposed that one might preclude such degenerate solutions by bounding the sum of the squared dissimilarities from below. We take the set of updated dissimilarities from the isotonic regression step (NNScal 4b) and denote it $\overline{\Delta} = [\overline{\delta_{ij}}]$. Next, we compute the sum $\overline{c} = \sum_i \sum_j w_{ij} \overline{\delta_{ij}}$. By lemma 2 in [12], we satisfy the nondegeneracy constraint if we update the dissimilarities as follows: iff $\overline{c} < c$, we rescale the dissimilarities by $\delta_{ij} \leftarrow c \overline{\delta_{ij}} / \overline{c}$.

5. Numerical Experiments

In this section, we perform numerical experiments to demonstrate NNScal as a method for embedding ordinal nearest neighbor graphs. In 5.1, we exhibit the effect of the sparsity of the nearest neighbor graph. In 5.2, we demonstrate the relationship between σ_a , M_1 , and M_2 . In 5.3, we demonstrate its performance on a large, sparse nearest neighbor graph ($n = 50,000$).

5.1 The effect of sparsity

Inspired by the experimentation in [6], we constructed 10 unique configurations of 200 points randomly drawn from the unit circle in \mathbb{R}^2 . The proximity graphs were then the k -nearest neighbors according to Euclidean distance in \mathbb{R}^2 . Configurations were used only if their 4-nearest neighbor graph was connected. Graef and Spence [6] were interested in studying the effect of missing proximities on metric MDS given perturbed interpoint distances. The loss of information relevant to our formulation is that no metric information is maintained, however, we know an accurate ordering of neighboring proximities. We use M_1 and M_2 (see Section 3) to assess the quality of the embedding.

Two initial configurations were generated, one by Classical MDS with $\hat{r} = 2r_{\max}$ (IC1) and one at random (IC2). In practice, i.e. with large graphs, generating an initial configuration with Classical MDS is not feasible due to computational expense. This example demonstrates the importance of the initial configuration.

We performed two separate experiments. First, for each initial configuration and dataset we construct a k -nearest neighbor graph and add $m = 1$ non-neighbor edge. Second, we construct a k nearest neighbor graph and add $m = 5k$ non-neighbor edges. In each of these trials $k \in \{10, 13, 16, 19, 22, 25, 28\}$. The two criteria over the 10 datasets ± 1 standard deviation are plotted in Figure 2.

The importance of the initial configuration is evident in all four plots. It is also worth noting that the performance of the embedding with more proximities included in the stress calculation (Experiment 2) did not necessarily outperform the embedding with less proximities included (Experiment 1). For example, consider how Experiment 1 outperformed Experiment 2 for both configurations with respect to M_1 when $k = 13, 16, 19, 22, 25$, and 28.

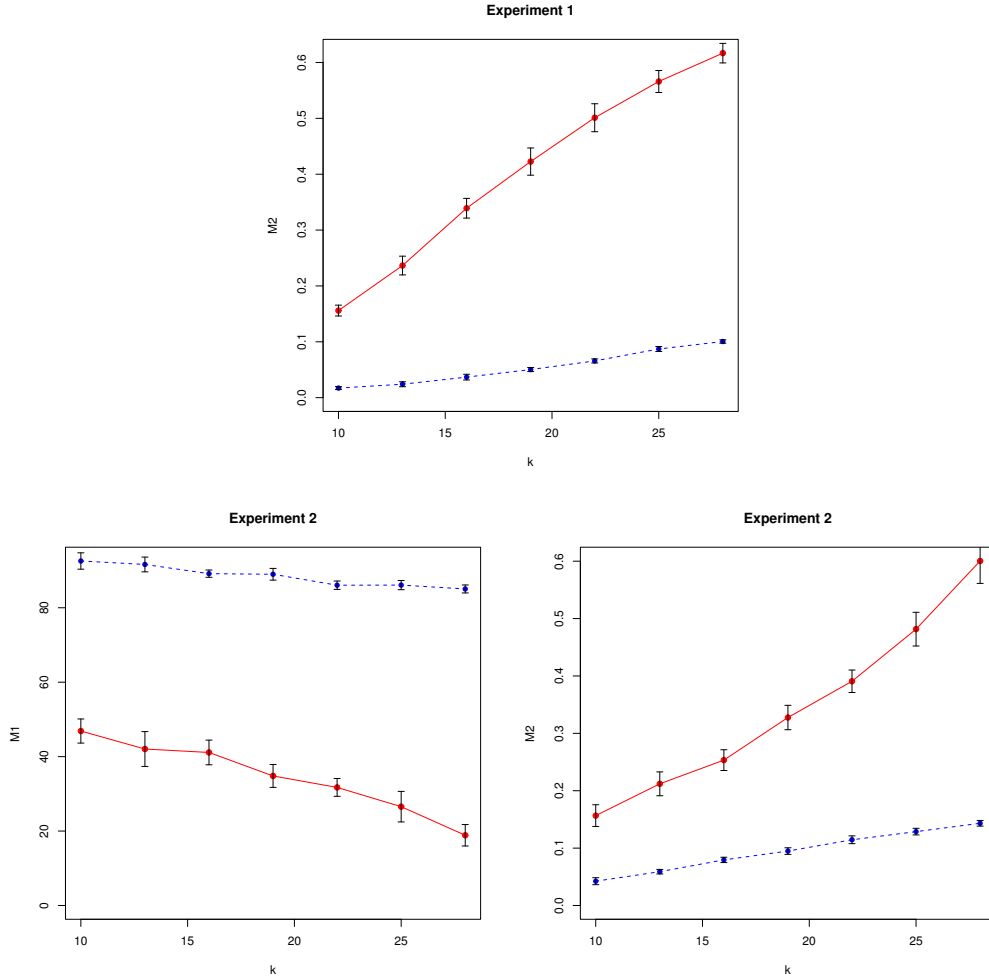


Figure 2: Solid line = IC1, Dashed line = IC2

5.2 Stress vs. M_1 and M_2

One might wonder how effective stress is for improving M_1 and M_2 . In this experiment, we use one dataset from above and construct its $k = 20$ -nearest neighbor graph with $5k$ non-neighbor edges included. For this example, we start with a randomly generated configuration, repeated step 4 of NNScal 50 times, and for each step 4a we iterated DMA 4 times. Below, we have plotted $\log(\sigma_a)$, M_1 , and M_2 (Figure 3) during the optimization. Each criterion was computed after each update of either X or Δ . It is important to note, however, that M_1 and M_2 are constant for fixed X .

It is clear that, at least in this experiment, the relationship between asymmetric stress, M_1 , and M_2 is essentially monotonic. However, the plots indicate that while stress improves significantly at the beginning of the algorithm and slows at the end, the improvement in M_1 or M_2 does not “kick in” until later. We believe that this indicates that M_1 and M_2 are only improved by minimizing stress once a “reasonable” configuration is obtained. For the sake of comparison we have repeated the above experiment, but instead of using a randomly generated configuration (IC2) we start with a more principled configuration (IC1) (see Figure 4). This experiment

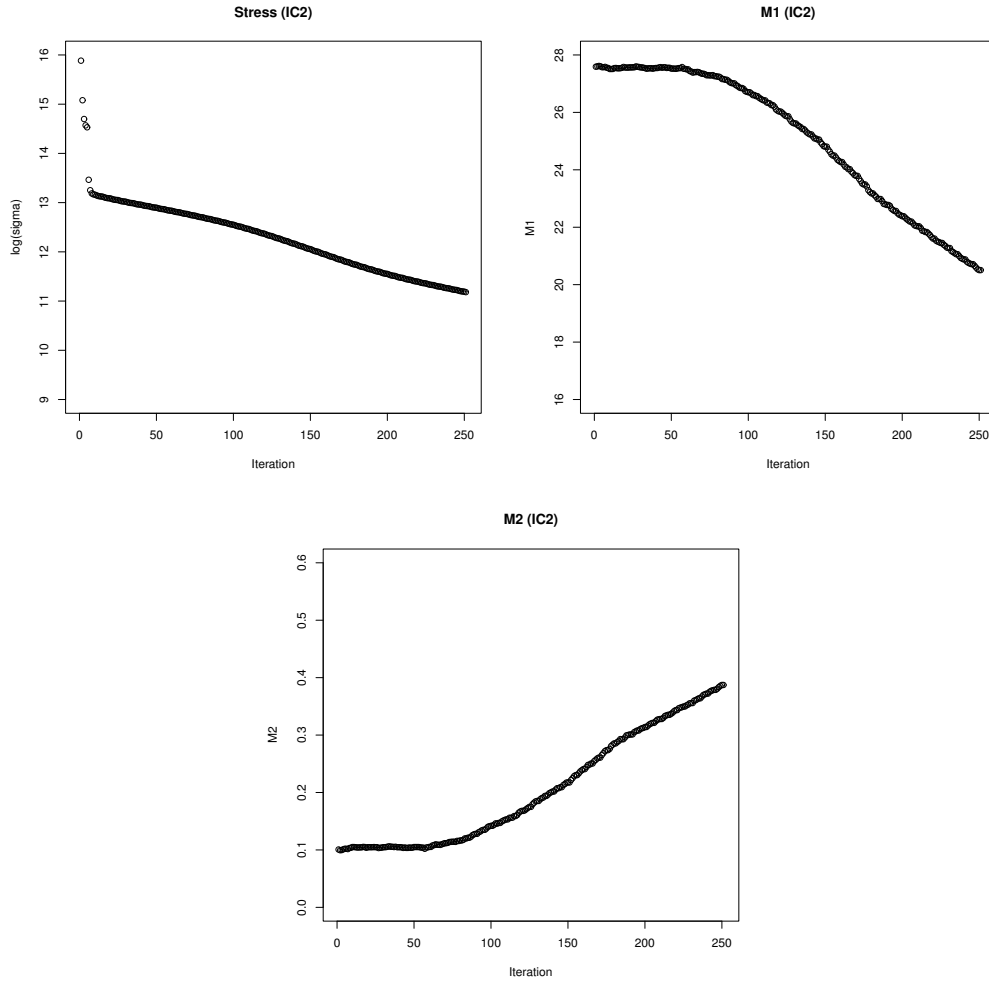


Figure 3: Asymmetric stress (left), M_1 (center), M_2 (right) – Randomly generated configuration (IC2)

demonstrates the importance of the initial configuration on the ability to improve M_1 and M_2 .

A representation of the embedding starting with a random configuration (IC2) is in the top left plot of Figure 5. To demonstrate that local structure was maintained we have added lines representing the original edges in the nearest neighbor graph (top right plot in Figure 5). To remove clutter for the purpose of visualizing the local structure we picked two sets of 5 points at random and drew the edges to their respective nearest neighbors (bottom left and right plots in Figure 5).

5.3 “Large Graph”

A random, connected 9-nearest neighbor graph containing $|V| = 50,000$ nodes was constructed. For each node, $m = 5k = 45$ non-neighbors were selected at random to form E^+ . Step 4 of NNScal was repeated a total of four times and for each iteration of step 4, step 4a was repeated three times. The purpose of this experiment is to demonstrate that we can minimize asymmetric stress on large proximity graphs. The computation was performed on a single core of a 2.4 GHz Intel Core 2 Duo. In

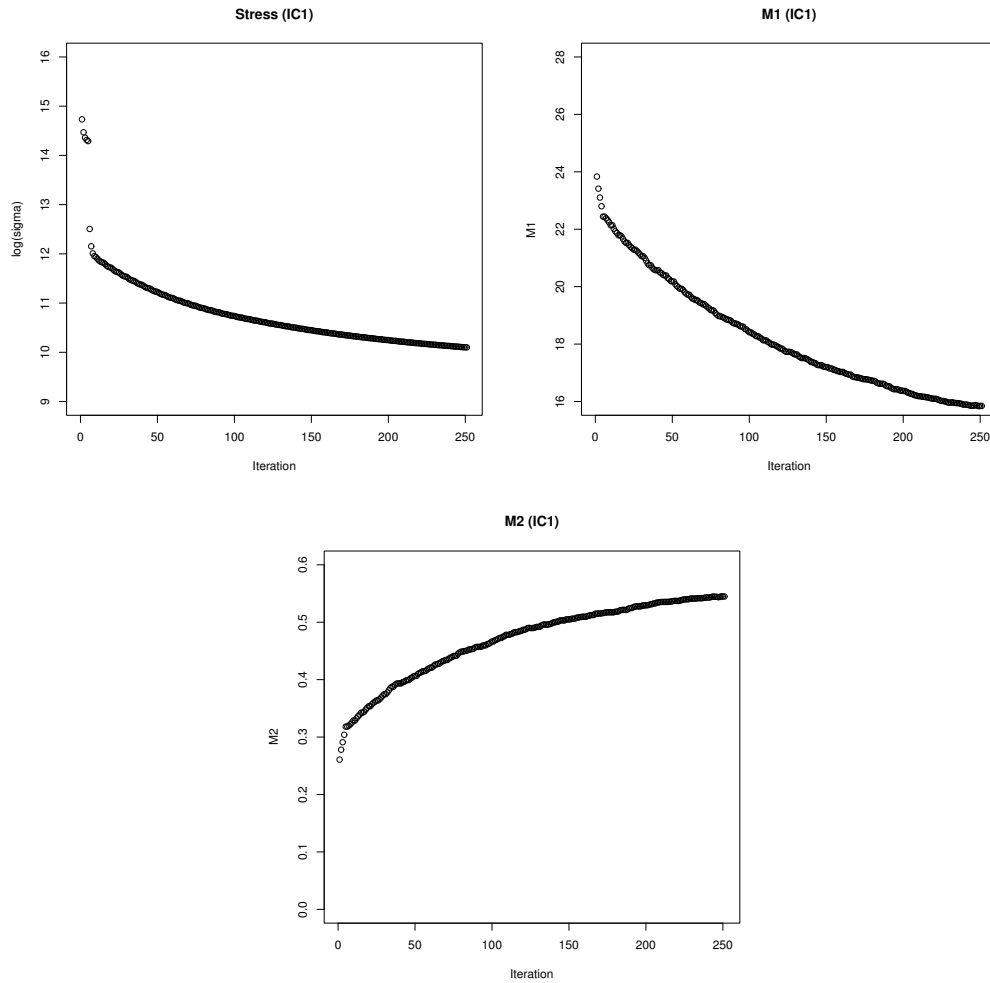


Figure 4: Asymmetric stress (left), M_1 (center), M_2 (right) – Classical MDS initial configuration (IC1)

Figure 6 we plot the log of the asymmetric stress with respect to wall clock time.

6. Acknowledgements

The research described herein was supported by a grant from the Office of Naval Research and presented in a contributed poster session at the Joint Statistical Meetings in Vancouver, Canada, July 31 to August 5, 2010. This paper will also appear as an Indiana University technical report. The authors thank Carey E. Priebe and Minh Tang for helpful discussions and comments.

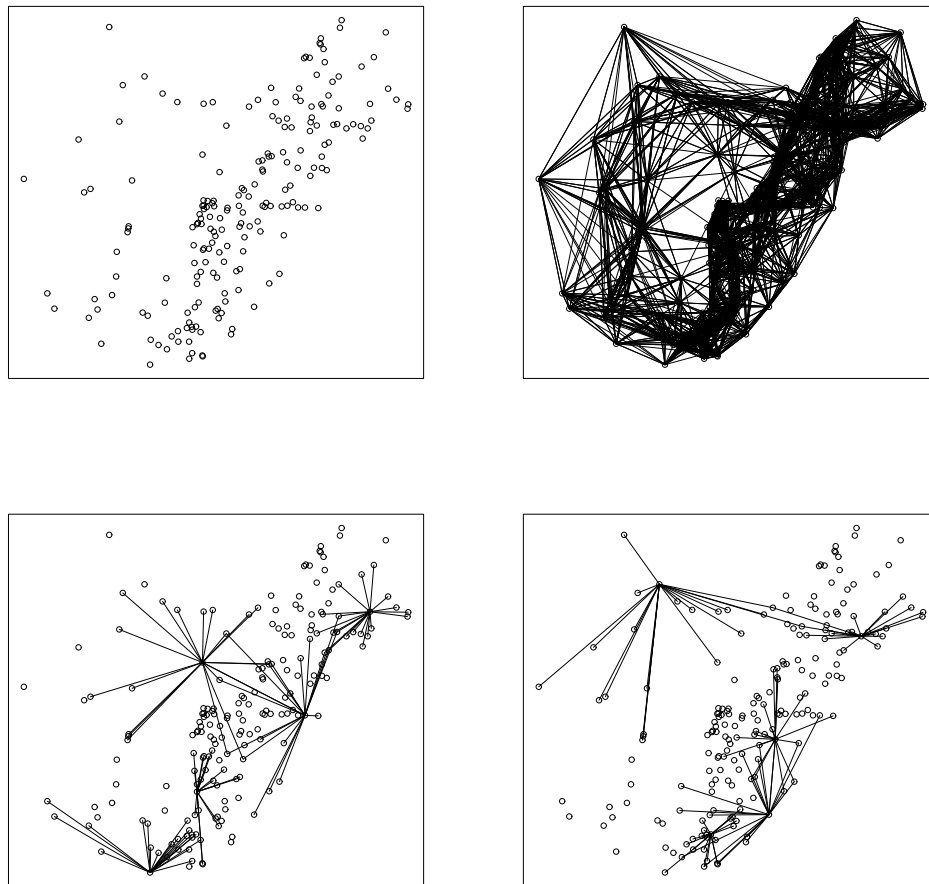


Figure 5: Embedding (top left) with edges drawn to connect nearest neighbors (other three plots).

References

- [1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002.
- [2] O. Burdakov, O. Sysoev, A. Grimvall, and M. Hussian. An $O(n^2)$ algorithm for isotonic regression. 2006.
- [3] L. Chen and A. Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing and proximity analysis. *Journal of American Statistical Association*, 104:209–219, 2009.
- [4] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *S. Becker, S. Thrun, and K. Obermayer, editors, Proc. NIPS*, 15:721–728, 2003.
- [5] C. Faloutsos and K. Lin. FastMap: a fast algorithm for indexing, data-mining, and visualization. In *Proc. ACM SIGMOD*, pages 163–174, 1995.

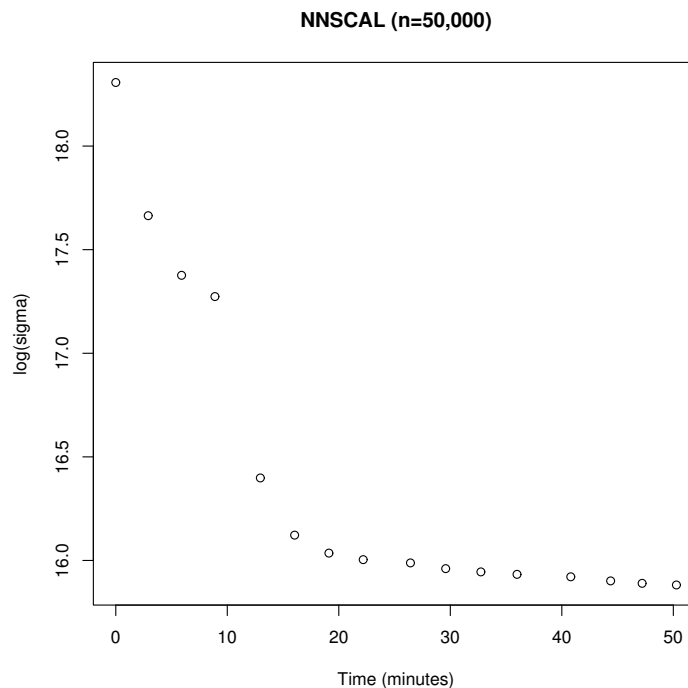


Figure 6: $\log(\sigma_a)$ on a large graph ($n=50,000$)

- [6] J. Graef and I. Spence. Using distance information in the design of large multidimensional scaling experiments. *Psychological Bulletin*, 86:60–66, 1979.
- [7] S. Grotzinger and C. Witzgall. Projections onto order simplexes. *Applied Mathematics and Optimization*, 12:247–270, 1984.
- [8] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.
- [9] J. C. Platt. FastMap, MetricMap, and Landmark MDS are all Nyström Algorithms. *Technical Report, Microsoft Corporation*, 2004.
- [10] J. B. Tenenbaum, V. Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.
- [11] M. W. Trosset. Formulations of multidimensional scaling for cluster analysis and classification. 1997.
- [12] M. W. Trosset. A new formulation of the nonmetric strain problem in multidimensional scaling. *Journal of Classification*, 15:15–35, 1998.
- [13] M. W. Trosset and P. J. F. Groenen. Multidimensional scaling algorithms for large data sets. *Computing Science and Statistics CD-ROM*, 2005.
- [14] J. T.-L. Wang, X. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang. Evaluating a class of distance-mapping algorithms for data mining and clustering. In *Proc. ACM KDD*, pages 307–311, 1999.