

A Remark on Global Positioning from Local Distances

Amit Singer^{*†}

[†]Yale University, Department of Mathematics, New Haven, CT 06520

Submitted to Proceedings of the National Academy of Sciences of the United States of America

Finding the global positioning of points in Euclidean space from local or partial set of pairwise distances is a problem in geometry that emerges naturally in sensor networks and NMR spectroscopy of proteins. We observe that the eigenvectors of a certain sparse matrix exactly match the sought coordinates. This translates to a simple and efficient algorithm which is robust to noisy distance data.

distance geometry | sensor networks | multidimensional scaling | eigenvectors

Determining the configuration of N points r_i in \mathbb{R}^p (e.g., $p = 2, 3$) given their noisy distance matrix Δ_{ij} is a long standing problem. For example, the points r_i may represent coordinates of cities in the United States. Distances between nearby cities, such as New York and New Haven, are known, but no information is given for the distance between New York and Chicago. In general, for every city we are given distance measurements to its $k \ll N$ nearest neighbors or to neighbors which are at most Δ away. The distance measurements Δ_{ij} of $d_{ij} = \|r_i - r_j\|$ may also incorporate errors. The problem is to find the unknown coordinates $r_i = (x_i, y_i)$ of all cities from the noisy local distances.

Even in the absence of noise the problem is solvable only if there are enough distance constraints. By solvable we mean that there is a unique set of coordinates satisfying the given distance constraints up to rigid transformations (rotations, translations and reflection). Alternatively, we say that the problem is solvable only if the underlying graph consisting of the N cities as vertices and the distance constraints as edges is rigid in \mathbb{R}^p (see, e.g. [8]). Graph rigidity had drawn a lot of attention over the years, see [8, 9, 10] for some results in this area.

When all possible $\binom{N}{2}$ pairwise distances are known, the corresponding complete graph is rigid, and the coordinates can be computed using a classical method known as multidimensional scaling (MDS) [13, 14, 15]. The underlying principle of MDS is to use the law of cosines to convert distances into an inner product matrix, whose eigenvectors are the sought coordinates. The eigenvectors computed by MDS are also the solution to a specific minimization problem. However, when most of the distance matrix entries are missing, this minimization becomes significantly more challenging due to the low rank- p constraint which is not convex. Various solutions to this constrained optimization problem have been proposed, including relaxation of the rank constraint by semidefinite programming (SDP) [16], regularization [17, 18], smart initialization [19], expectation maximization [20] and other relaxation schemes [7]. Recently, the eigenvectors of the graph Laplacian [6] were used to approximate the large scale SDP problem by a much smaller one, noting that linear combinations of only a few Laplacian eigenvectors can well approximate the coordinates.

The problem of finding the global coordinates from noisy local distances arises naturally in localization of sensor net-

works [1, 2, 3] and protein structuring from NMR spectroscopy [4, 5], where noisy measurements of neighboring hydrogen atoms are inferred from the nuclear Overhauser effect (NOE). The theory and practice involves many different areas, such as machine learning [6], optimization techniques [7] and rigidity theory [8, 9, 10, 11, 2, 12]. We find it practically impossible to give a fair account of all the relevant literature, and recommend the interested reader to see the references within those listed.

In this paper we describe locally rigid embedding (LRE), a simple and efficient algorithm to find the global configuration from locally noisy distances, under the simplifying assumption of local rigidity. We assume that every city, together with its neighboring cities, form a rigid subgraph that can be embedded uniquely (up to a rigid transformation). That is, there are enough distance constraints among the neighboring cities that make the local structure rigid. The “pebble game” [27] is a fast algorithm to determine graph rigidity. Thus, given a data set of distances, the local rigidity assumption can be tested quickly by applying the pebble game algorithm N times, once for each city. In the summary section we discuss the restrictiveness of the local rigidity assumption (see [11, 2, 12] for conditions that ensure rigidity), as well as possible variants of LRE when the assumption does not hold.

The essence of LRE follows the observation that one can construct an $N \times N$ sparse weight matrix W which has the property that the coordinate vectors are an eigenspace. The matrix is constructed in linear time complexity and has only $O(N)$ non-zero elements, which enables efficient computation of that small eigenspace. The matrix W is formed by pre-processing the local distance information to repeatedly embed each city and its neighboring points (e.g., by using MDS or SDP), which is possible by the assumption that every local neighborhood is rigid. Once the local coordinates are obtained, we calculate weights much like the locally linear embedding (LLE) recipe [21] and its multiple weights (MLLE) modification [26]. Similar to recent dimensionality reduction methods [21, 22, 23, 24], the motif “think globally, fit locally” [25] is repeated here. This is just another example for the usefulness of eigenfunctions of operators on data sets, and their ability to integrate local information to a consistent global solution. In contrast to propagation algorithms that start with some local embedding and incrementally embed additional points while accumulating errors, the global eigenvector computation of LRE takes into account all local information at once, which makes it robust to noise.

Conflict of interest footnote placeholder

Insert 'This paper was submitted directly to the PNAS office.' when applicable.

* To whom correspondence should be addressed. E-mail: amit.singer@yale.edu

©2007 by The National Academy of Sciences of the USA

Locally Rigid Embedding

We start by embedding points locally. For every point r_i ($i = 1, \dots, N$) we examine its k_i neighbors $r_{i_1}, r_{i_2}, \dots, r_{i_{k_i}}$ for which we have distance information. We find a p -dimensional embedding of those $k_i + 1$ points. If the local $(k_i + 1) \times (k_i + 1)$ distance matrix is fully known then this embedding can be obtained using MDS. If some local distances are missing, then the embedding can be found using SDP or other optimization methods. The assumption of local rigidity ensures that in the noise-free case such an embedding is unique up to a rigid transformation. Note that the neighbors are not required to be physically close, but can rather be any collection of k_i points that the distance constraints among them make them rigid. This locally rigid embedding gives local coordinates for $r_i, r_{i_1}, \dots, r_{i_{k_i}}$ up to translation, rotation and perhaps reflection. We define k_i weights W_{i,i_j} that satisfy the following $p+1$ linear constraints

$$\sum_{j=1}^{k_i} W_{i,i_j} r_{i_j} = r_i \quad [1]$$

$$\sum_{j=1}^{k_i} W_{i,i_j} = 1 \quad [2]$$

The vector equation (1) means that the point r_i is the center of mass of its k_i neighbors, and the scalar equation (2) means that the weights sums up to one. Together, the weights are invariant to rigid transformations (translation, rotation and reflection) of the locally embedded points. It is possible to find such weights if $k_i \geq p + 1$. In fact, for $k_i > p + 1$ there is an infinite number of ways in which the weights can be chosen. In practice we choose the least square solution, i.e., that with $\min \sum_{j=1}^{k_i} W_{i,i_j}^2$. This prevents certain weights from becoming too large and keeps the weights balanced in magnitude. We rewrite equations (1)-(2) in matrix notation as

$$R_i w_i = b,$$

where $b = (0, \dots, 0, 1)^T \in \mathbb{R}^{p+1}$, $w_i = (W_{i,i_1}, W_{i,i_2}, \dots, W_{i,i_{k_i}})^T$ and R_i is a $(p+1) \times k_i$ matrix, whose first p rows are the relative coordinates and the last row is the all ones vector

$$R_i = \begin{pmatrix} x_{i_1} - x_i & x_{i_2} - x_i & \cdots & x_{i_{k_i}} - x_i \\ y_{i_1} - y_i & y_{i_2} - y_i & \cdots & y_{i_{k_i}} - y_i \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

It can be easily verified that the least squares solution is

$$w_i = R_i^T (R_i R_i^T)^{-1} b.$$

Note that some of the weights may be negative. In fact, negative weights are unavoidable for points on the boundary of the convex hull of the set.

The computed weights W_{i,i_j} ($j = 1, \dots, k_i$) are assigned to the i 'th row of W , while all other $N - k_i$ row entries, including the diagonal element $W_{i,i}$, are set to zero. This procedure of locally rigid embedding is repeated for every $i = 1, \dots, N$, each time filling another row of the weight matrix W . Therefore, it takes $O(N)$ operations to complete the construction of the weight matrix W , which ends up being a sparse matrix with kN non-zero elements, where $k = \frac{1}{N} \sum_{i=1}^N k_i$ is the average degree of the graph (i.e., the average number of neighbors). Both the least squares solution and the local embedding (MDS or SDP) are polynomial in k_i , so repeating them N times is also $O(N)$.

Next, we examine the spectrum of W . Observe that the all ones vector $\mathbf{1} = (1, 1, \dots, 1)^T$ satisfies

$$W\mathbf{1} = \mathbf{1} \quad [3]$$

due to (2). That is, $\mathbf{1}$ is an eigenvector of W with an eigenvalue $\lambda = 1$. We will refer to $\mathbf{1}$ as the trivial eigenvector. If the locally rigid embeddings were free of any error (up to rotations and reflections), then the p coordinate vectors $\mathbf{x} = (x_1, x_2, \dots, x_N)^T, \mathbf{y} = (y_1, y_2, \dots, y_N)^T, \dots$ are also eigenvectors of W with the same eigenvalue $\lambda = 1$:

$$W\mathbf{x} = \mathbf{x}, \quad W\mathbf{y} = \mathbf{y}, \quad \dots \quad [4]$$

This follows immediately from (1) by reading it one coordinate at a time,

$$\sum_{j=1}^N W_{ij} x_j = x_i, \quad \sum_{j=1}^N W_{ij} y_j = y_i, \quad \dots$$

Thus, the eigenvectors of W corresponding to $\lambda = 1$ give the sought coordinates, e.g. (x_i, y_i) for $p = 2$. It is remarkable that the eigenvectors give the global coordinates x and y despite the fact that different points have different local coordinate sets that are arbitrarily rotated, translated and perhaps reflected with respect to one another.

Note that W is not symmetric, so its spectrum may be complex. Although every row of W sums up to one, this does not prevent the possibility of eigenvalues $|\lambda| > 1$, because W is not stochastic, having negative entries. Therefore, we compute the eigenvectors with eigenvalues closest to one in the complex plane.

A symmetric weight matrix S with spectral properties similar to W can be constructed with the same effort (due to D. A. Spielman, private communication). For each point r_i , compute an orthonormal basis $\tilde{w}_{i,1}, \dots, \tilde{w}_{i,k_i-p} \in \mathbb{R}^{k_i+1}$ for the nullspace of the $(p+1) \times (k_i+1)$ matrix \tilde{R}_i

$$\tilde{R}_i = \begin{pmatrix} x_i & x_{i_1} & x_{i_2} & \cdots & x_{i_{k_i}} \\ y_i & y_{i_1} & y_{i_2} & \cdots & y_{i_{k_i}} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \quad [5]$$

In other words, we find $\tilde{w}_{i,j}$ that satisfy

$$\tilde{R}_i \tilde{w}_{i,j} = 0, \quad j = 1, \dots, k_i - p$$

and

$$\langle \tilde{w}_{i,j_1}, \tilde{w}_{i,j_2} \rangle = \begin{cases} 1 & j_1 = j_2 \\ 0 & j_1 \neq j_2 \end{cases}$$

The symmetric product $S_i = \sum_{j=1}^{k_i-p} \tilde{w}_{i,j} \tilde{w}_{i,j}^T$ of the nullspace with itself is a $(k_i+1) \times (k_i+1)$ positive semidefinite (PSD) matrix. We sum the N PSD matrices S_i ($i = 1, \dots, N$) to form an $N \times N$ PSD S by updating for each i only the corresponding $(k_i+1) \times (k_i+1)$ block of the large matrix

$$S = \sum_{i=1}^N \sum_{j=1}^{k_i-p} \hat{w}_{i,j} \hat{w}_{i,j}^T, \quad [6]$$

where $\hat{w}_{i,j} \in \mathbb{R}^N$ is obtained from $\tilde{w}_{i,j}$ by padding it with zeros. Similar to the non-symmetric case, it can be verified that the $p+1$ vectors $\mathbf{1}, \mathbf{x}, \mathbf{y}, \dots$ belong to the nullspace of S

$$S\mathbf{1} = 0, \quad S\mathbf{x} = 0, \quad S\mathbf{y} = 0, \quad \dots$$

In fact, since the global graph of N points is rigid, any vector in the nullspace of S is a linear combination of those $p+1$ vectors (the previous construction of W may not enjoy this nice property, i.e., it is possible to create examples for which the dimension of the eigenspace of W is greater than $p+1$). In rigidity theory we would call S a stress of the framework [8].

From the computational point of view, we remark that, if needed, the eigenvector computation can be done in parallel. In order to multiply a given vector by the weight matrix W (or S), the storage of W (S) is distributed between the N points, such that the i 'th point stores only k_i values of W (S) together with k_i values of the given vector that correspond to the neighbors.

Multiplicity and Noise

The eigenvalue $\lambda = 1$ of W (or 0 for S) is degenerate with multiplicity $p+1$. Hence, the corresponding $p+1$ computed eigenvectors $\phi^0, \phi^1, \dots, \phi^p$ may be any linear combination of $\mathbf{1}, \mathbf{x}, \mathbf{y}, \dots$. We may assume that $\phi^0 = \mathbf{1}$ and that $\langle \phi^j, \mathbf{1} \rangle = 0$ for $j = 1, \dots, p$. We now look for a $p \times p$ matrix A that represents the linear transformation from the original coordinate set $r_i = (x_i, y_i, \dots)$ to the eigenmap $\phi_i = (\phi_i^1, \phi_i^2, \dots)$

$$r_i = A\phi_i, \quad \text{for } i = 1, \dots, N.$$

The squared distance between r_i and r_j is

$$d_{ij}^2 = \|r_i - r_j\|^2 = (\phi_i - \phi_j)^T A^T A (\phi_i - \phi_j). \quad [7]$$

Every pair of points for which the distance d_{ij} is known gives a linear equation for the coefficients of the matrix $A^T A$. Solving that equation set results in $A^T A$, whose Cholesky decomposition yields A (up to an orthogonal transformation).

The locally rigid embedding incorporates errors when the distances are noisy. The noise breaks the degeneracy and splits the eigenvalues. If the noise level is small, then we still expect the first p non-trivial eigenvectors to match the original coordinates, because the small noise does not overcome the spectral gap

$$\min_{\lambda_i \neq 1} |1 - \lambda_i|,$$

where λ_i are the eigenvalues of the noiseless W . For higher levels of noise, the noise may be large enough so that some of the first p eigenvalues cross the spectral gap, causing the coordinate vectors to spill over the remaining eigenvectors.

To overcome this problem, we allow A to be $p \times m$ instead of $p \times p$, with $m > p$, but still $m \ll N$. In other words, every coordinate is approximated as a linear combination of the m non-trivial eigenvectors ϕ^1, \dots, ϕ^m , whose eigenvalues are the closest to one. We replace equation (7) by a minimization problem for the $m \times m$ PSD matrix $P = A^T A$

$$\min \sum_{i \sim j} \left[(\phi_i - \phi_j)^T P (\phi_i - \phi_j) - \Delta_{ij}^2 \right]^2, \quad [8]$$

where the sum is over all pairs $i \sim j$ for which there exists a distance measurement Δ_{ij} . The least squares solution for P is often not valid, when it ends up not being a PSD matrix or

when its rank is greater than p . It was shown in [6] that the optimization problem (8) with the PSD constraint $P \succ 0$ can be formulated as SDP using the Schur complement lemma. The matrix A is reconstructed from the singular vectors of P corresponding to the largest p singular values.

Numerical Results

We consider a data set of $N = 1097$ cities in the United States, see Figure 1 (top left). For each city we have distance measurements to its 18 nearest cities ($k_i = \bar{k} = 18$). For simplicity, we assume that the $\binom{18}{2}$ distances between the neighboring cities are also collected, so that MDS is used to locally embed the neighboring cities. We tested the locally rigid embedding algorithm for three different levels of multiplicative Gaussian noise: no noise, 1% noise, and 10% noise. That is, $\Delta_{ij} = d_{ij} [1 + \mathcal{N}(0, \delta^2)]$, with $\delta = 0, 0.01, 0.1$. We used SeDuMi [28] for solving the SDP (8).

The behavior of the numerical spectrum for the different noise levels is illustrated in Figure 2, where the magnitude of $1 - \lambda_i$ for the ten eigenvalues that are closest to one is plotted. It is apparent in the noiseless case that $\lambda = 1$ is degenerated with multiplicity 3 corresponding to $\mathbf{1}, \mathbf{x}, \mathbf{y}$. Note that the spectral gap is quite small. The LRE finds the exact configuration without introducing any errors as can be seen in Figure 1 (top right).

Adding 1% noise breaks the degeneracy, splits the eigenvalues, and due to the small spectral gap, the coordinates spill over the fourth eigenvector. Therefore, for that noise level we solved the minimization problem (8) with P being a 3×3 matrix ($m = 3$). The largest singular values of P were 101.7, 42.5, and 0.13 indicating a successful two dimensional embedding.

When the noise is increased more, the spectrum changes even further. We used $m = 7$ for the noise level of 10%. The largest singular values of the optimal P were 76.6, 8.3, 2.0, and 0.002, which implies that the added noise effectively increased the dimension of points from two to three. The LRE distorts boundary points, but does a good job in the interior.

Algorithm

In this section we summarize the various steps of LRE that should be taken in order to find the global coordinates from a data set of vertices and distances, using the symmetric matrix S .

1. *Input:* A weighted graph $G = (V, E, \Delta)$, Δ_{ij} is the measured distance between vertices i and j , and $|V| = N$ is the number of vertices. G is assumed to be locally rigid.
2. Allocate memory for a sparse $N \times N$ matrix S .
3. For $i = 1$ to N
 - (a) Set k_i to be the degree of vertex i .
 - (b) i_1, \dots, i_{k_i} are the neighbors of i : $(i, i_j) \in E$ for $j = 1, \dots, k_i$.
 - (c) Use SDP or MDS to find an embedding $r_i, r_{i_1}, \dots, r_{i_{k_i}}$ of i, i_1, \dots, i_{k_i} .
 - (d) Form the $(p+1) \times (k_i+1)$ matrix \tilde{R}_i following eq.(5).

- (e) Compute an orthogonal basis $\tilde{w}_{i,1}, \dots, \tilde{w}_{i,k_i-p}$ for the nullspace of \tilde{R}_i .
 - (f) Update $S \leftarrow S + \sum_{j=1}^{k_i-p} \tilde{w}_{i,j} \tilde{w}_{i,j}^T$.
4. Compute $m+1$ eigenvectors of S with eigenvalues closest to 0: $\phi_0, \phi_1, \dots, \phi_m$, where ϕ_0 is the all-ones vector.
 5. Use SDP as in eq.(8) to find coordinate vectors x_1, \dots, x_p as linear combinations of ϕ_1, \dots, ϕ_m .

Summary and Discussion

We presented an observation that leads to a simple algorithm for finding the global configuration of points from their local noisy distances under the assumption of local rigidity. The LRE algorithm that uses the matrix W is similar to the LLE algorithm [21] for dimensionality reduction. The input to LLE are high dimensional data points for which LLE constructs the weight matrix W by repeatedly solving overdetermined linear equation systems. In the global positioning problem the input is not high dimensional data points, but rather distance constraints that we preprocess (using MDS or SDP) in order to get a locally rigid low dimensional embedding. Therefore, the weights are obtained by solving an underdetermined system (1)-(2) instead of an overdetermined system. The LRE variant that uses the symmetric matrix S is similar to the MLE algorithm [26] in the sense that it uses more than a single weight vector, taking all basis vectors of the nullspace into account in the construction of S .

The assumption of local rigidity of the input graph is crucial for the algorithm to succeed. It is reasonable to ask whether or not this assumption holds in practice, and how the algorithm can be modified to prevail even if the assumption fails to hold. These are important research questions that we will attempt to address in a future publication. Early numerical experiments with globally rigid two-dimensional data

sets show that even if a few vertices have corresponding subgraphs that are not rigid (non-localizable), so their weights are not included in the construction of the stress matrix S , the matrix S still has a nullspace of dimension three. This can happen if the non-localizable vertices have enough localizable neighbors. Still, the algorithm fails when there are too many non-localizable vertices. For such non-localizable vertices, one possible approach is to consider larger subgraphs, such as the one containing all 2-hop neighbors (the neighbors of the neighbors), hoping that the larger subgraph would be rigid. A different approach may be to consider smaller subgraphs, hoping that the removal of some neighbors would make the subgraph rigid. This is obviously possible if the vertex is included in a triangle (or some other small simple rigid subgraph), but at risk that the small subgraph would be too small to be rigidly connected to the remaining graph. It is not clear which approach would work better and requires numerical experiments. However, this is out of scope of the current paper.

There are many other interesting questions that follow from both the theoretical and algorithmic aspects, such as proving error bounds, finding other suitable operators for curves and surfaces, and dealing with specific constraints that arise from real life data sets.

I would like to thank Ronald R. Coifman who inspired this work. The center of mass operator emerged from my joint work with Yoel Shkolnisky on CryoEM tomography. Dan Spielman is acknowledged for designing the symmetrized weight matrix. I would also like to thank Jia Fang, Yosi Keller, Yuval Kluger and Richard Yang for valuable discussions on sensor networks, NMR spectroscopy and rigidity. Finally, thanks are due to Rob Kirby for making valuable suggestions that improved the clarity of the manuscript. Research supported by NGA NURI 2006.

1. Biswas, P., Liang, T. C., Toh, K.C., Wang, T. C., and Ye, Y. (2006) *IEEE Transactions on Automation Science and Engineering*, **3**(4):360–371.
2. Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., Anderson, B. D. O., and Belhumeur, P. N. (2006) *IEEE Transactions on Mobile Computing*, **5** (12):1663–1678.
3. Ji, X. and Zha, H. (2004) In *INFOCOM 2004* **4**:2652–2661.
4. Crippen, G. M. and Havel, T. F. (1988) *Distance Geometry and Molecular Conformation*. John Wiley & Sons.
5. Berger, B., Kleinberg, J. and Leighton, F.T. (1999) *Journal of the ACM* **47** (2):212–235.
6. Weinberger, K. Q., Sha, F., Zhu, Q., and Saul, L. K. (2007) In Schölkopf, B., Platt, J., and Hofmann, T., editors, *Advances in Neural Information Processing Systems (NIPS) 19*. MIT Press, Cambridge, MA.
7. More, J. J. and Wu, Z. (1997) *SIAM Journal on Optimization*, **7**(3):814–836.
8. Roth, B. (1981) *The American Mathematical Monthly*, **88**(1):6–21.
9. Hendrickson, B. (1995) *SIAM Journal on Optimization*, **5** (4):835–857.
10. Hendrickson, B. (1992) *SIAM J. Comput.* **21** (1):65–84.
11. Anderson, B. D. O., Belhumeur, P. N., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., and Yang, Y. R. (2007) In *ACM Baltzer Wireless Networks (WINET)*, Springer.
12. Aspnes, J., Goldenberg, D. K., and Yang, Y. R. (2004) In *Proceedings of Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSORS)*, Lecture Notes in Computer Science 3121, Springer-Verlag, 32–44.
13. Schoenberg, I. J. (1935) *Ann. Math.* **36**:724–732.
14. Young, G. and Householder, A. S. (1938) *Psychometrika*, **3**:19–22.
15. Cox, T. F. and Cox, M. A. A. (2001) *Multidimensional Scaling*. Monographs on Statistics and Applied Probability 88. Chapman & Hall/CRC.
16. Vandenberghe, L., and Boyd, S. (1996) *SIAM Review*, **38**(1):49–95.
17. Wright S. J., Wahba, G., Lu, F., and Keles, S. (2005) *Proceedings of the National Academy of Sciences*, **102**(35):12332–12335.
18. Sun, J., Boyd, S., Xiao, L., and Diaconis, P. (2006) *SIAM Review*, **48**(4):681–699.
19. Gotsman, C. and Koren, Y. (2005) *Journal of Graph Algorithms and Applications*, **9**(3):327–346.
20. Srebro N. and Jaakkola T. (2003) In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 720–727.
21. Roweis, S. T. and Saul, L. K. (2000) *Science*, **290**(5500):2323–2326.
22. Donoho, D. L. and Grimes, C. (2003) *Proceedings of the National Academy of Sciences*, **100**(10):5591–5596.
23. Belkin, M. and Niyogi, P. (2003) *Neural Computation*, **15**:1373–1396.
24. Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S.W. (2005) *Proceedings of the National Academy of Sciences*, **102**(21):7426–7431.
25. Saul, L. K. and Roweis, S. T. (2003) *The Journal of Machine Learning Research*, **4**:119–155.
26. Zhang, Z. and Wang, J. (2007) In Schölkopf, B., Platt, J., and Hofmann, T., editors, *Advances in Neural Information Processing Systems (NIPS) 19*. MIT Press, Cambridge, MA, pp. 1593–1600.
27. Jacobs, D. J. and Hendrickson, B. (1997) *Journal of Computational Physics*, **137**:346–365.
28. Sturm, J. F. (1999) *Optimization Methods and Software*, **11** (1):625–653.

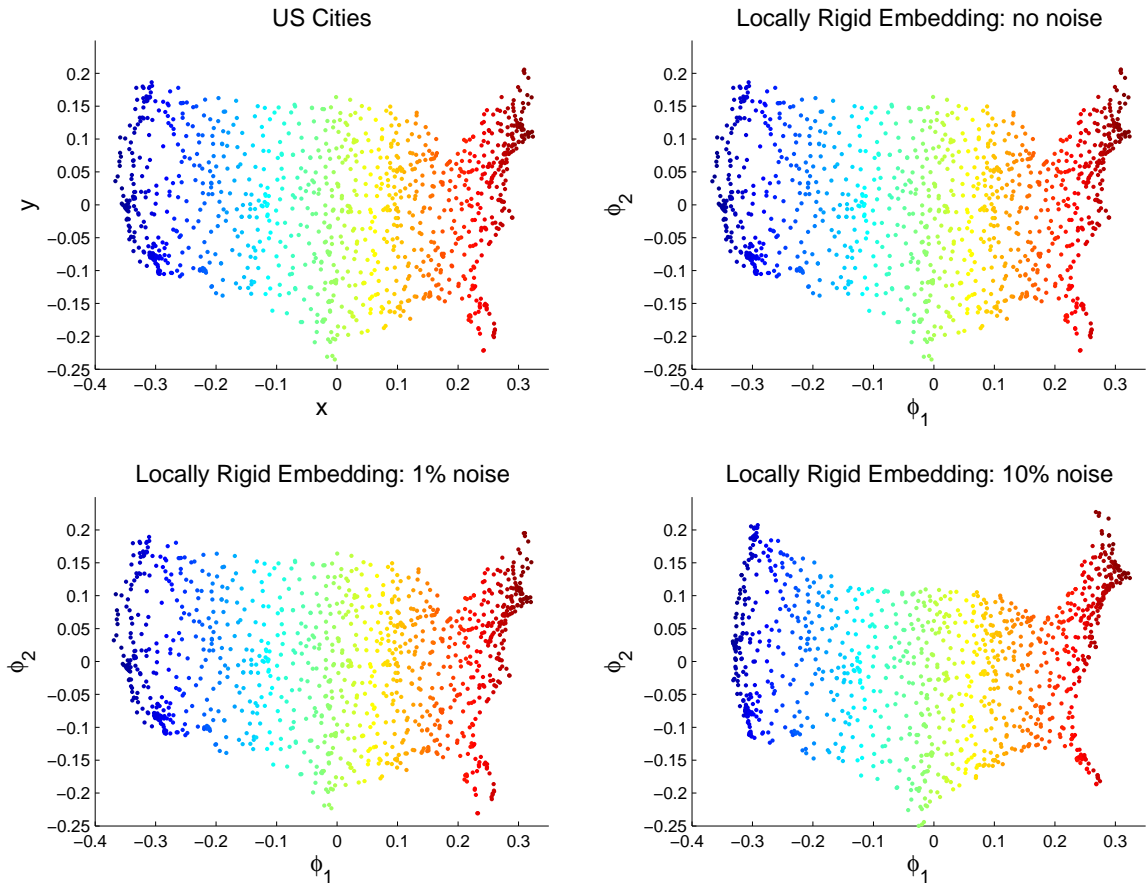


Fig. 1: Top left: map of 1097 cities in the United States. Top right: the locally rigid embedding obtained from clean local distances. Bottom left: LRE with 1% noise. Bottom right: LRE with 10% noise. Cities are colored by their x coordinate.

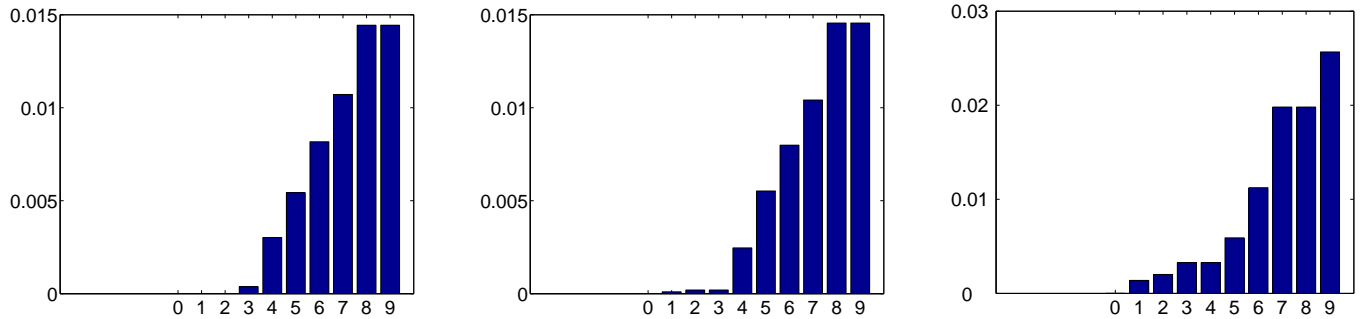


Fig. 2: Numerical spectrum of W for different levels of noise: clean distances (left), 1% noise (center), and 10% noise (right).