Laplacian Eigenmap-based Algorithm for Distance Geometry Problem

Zhenli Sheng (盛镇醴)

szl@lsec.cc.ac.cn

Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and Systems Science,
Chinese Academy of Sciences

March 4, 2014
seminar talk

Outline

Outline

the Problem

Distance Geometry Problem (DGP)

For a graph $G = (V, E)$, given a distance matrix D (L and U, respectively) on E and an integer d, find $x_1, x_2, \dots, x_n \in \mathbb{R}^d$, such that

$$\|x_i - x_j\| = d_{ij}, \quad (i, j) \in E.$$

or

$$l_{ij} \leq \|x_i - x_j\| \leq u_{ij}, \quad (i, j) \in E.$$

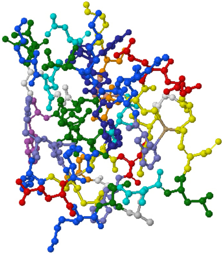
Related Applications/Problems

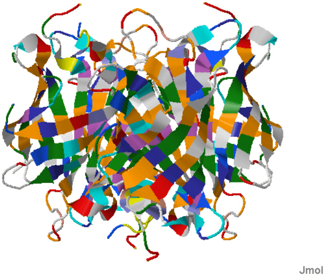
Some closely related applications/problems:

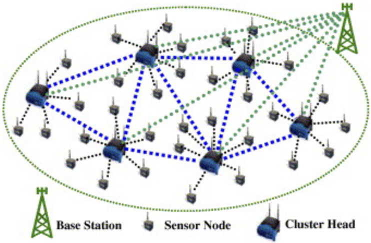
- Sensor network localization
  - $d = 2$, anchors
- Protein structure determination (molecular conformation)
  - $d = 3$, no anchors
- Graph realization
- Dimensionality reduction
  - d unknown, "distances" need be constructed
- Euclidean matrix completion

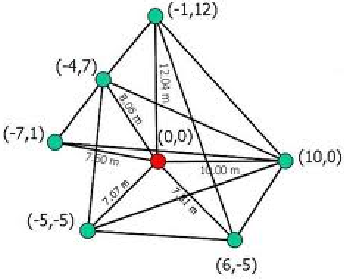
Note that

- In most cases, the distances are local and sparse.
- Usually, the distances are noisy.




Jmol


Base Station
Sensor Node
Cluster Head


(-1,12)
(-4,7)
(-7,1)
(0,0)
(10,0)
(-5,-5)
(6,-5)
7.50 m
10.00 m
7.07 m

Solution Methods

- Matrix Decomposition Method (Blumenthal 1953, Torgerson,1958)
- The Embedding Algorithm (Crippen-Havel, 1988)
- Global Smoothing Algorithm (Moré-Wu, 1997)
- Geometric Buildup Method (Dong-Wu, 2002, Sit-Wu-Yuan, 2009)
- SDP Relaxation Method (Biswas-Toh-Ye, 2007)
- Nearest Euclidean Distance Matrix (Qi-Yuan, 2013)
- The Branch-and-Prune Algorithm (Liberti-Lavor-Maculan-Mucherino, 2014)

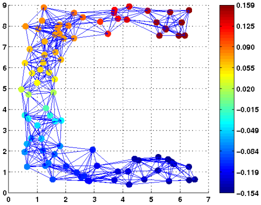
Remark that

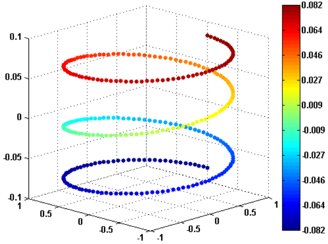
- Those are just some representative papers.
- Rigidity is also a very important problem.

Outline

2. Laplacian Eigenmap-based Algorithm
   - Laplacian Eigenmap
   - Algorithm Framework
   - Numerical Experiments

Motivation





- Interesting results from Laplacian matrix.
- Develop new methods to obtain warm starting points.

Nonlinear Dimensionality Reduction

Dimensionality Reduction Problem

Given a set of n points $x_1, \ldots, x_n$ in $\mathbb{R}^l$, find a set of points $y_1, \ldots, y_n$ in $\mathbb{R}^m (m \ll l)$ such that y_i "represents" x_i.

Relation to DGP:

- In noisy case, the distances can not be precisely realized in low-dimensional space.
- If the distances in DGP do not contradict with each other, n points can be precisely realized in at most $n-1$ Euclidean space.

Algorithm framework (Belkin-Niyogi, 2002)

Step 1 Constructing the adjacency graph
(a) ϵ-neighborhood (b) k-nearest neighbors

Step 2 Choosing the weights
(a) Heat kernel (b) Simple-minded

Step 3 Eigenmaps: $x_i \to (f_1(i), \ldots, f_m(i))$, where $Lf_i = \lambda_i Df_i$,
$0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{n-1}$.

Laplacian Matrix

Given a graph $G = (V, E)$, the Laplacian matrix L is defined by $L = D - W$, where

$$W_{ij} = \begin{cases} e^{-\|x_i - x_j\|^2/t} & \textit{if } (i,j) \in E, \\ 0 & \textit{otherwise}. \end{cases}$$

and D is a diagonal matrix with

$$D_{ii} = \sum_j W_{ij}.$$

Note that W degenerate to the adjacent matrix when $t = \infty$.

Optimal Embedding

Problem: Map the weighted graph G to a line such that the adjacent vertices stay as close as possible.

Let $y = (y_1, y_2, \ldots, y_n)^T$ be such a map, we have

$$\frac{1}{2} \sum_{(i,j)\in E} (y_i - y_j)^2 W_{ij} = y^T L y$$

Therefore,

$$\min_{y^T D y = 1, Y^T D e = 0} y^T L y$$

gives us a reasonable solution.

Optimal Embedding

Problem: Map the weighted graph G to a line such that the adjacent vertices stay as close as possible.

Let $y = (y_1, y_2, \ldots, y_n)^T$ be such a map, we have

$$\frac{1}{2} \sum_{(i,j) \in E} (y_i - y_j)^2 W_{ij} = y^T L y$$

Therefore,

$$\min_{y^T D y = 1, Y^T D e = 0} y^T L y$$

gives us a reasonable solution.

General case: Embed the graph into m-dimensional Euclidean space.

Let $Y = [y_1 \;\; y_2 \;\; \ldots \;\; y_n]^T \in \mathbb{R}^{n \times m}$ that each row gives a coordinate. Similarity we need to minimize

$$\sum_{(i,j) \in E} \|y_i - y_j\|^2 W_{ij} = tr(Y^T L Y)$$

.

Courant-Fischer Theorem

Courant-Fischer Theorem

Let A be a symmetric $n \times n$ matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and corresponding eigenvectors $v_1, v_2, \ldots, v_n$. For $1 \leq k \leq n$, let S_k denote the span of $v_1, \ldots, v_k$ (with $S_0 = \{0\}$), and let $S_k^{\perp}$ denote the orthogonal complement of S_k. Then

$$\lambda_k = \min_{\|x\|=1, x \in S_{k-1}^{\perp}} x^T A x = \min_{x \in S_{k-1}^{\perp}} \frac{x^T A x}{x^T x}$$

Toy Example 1

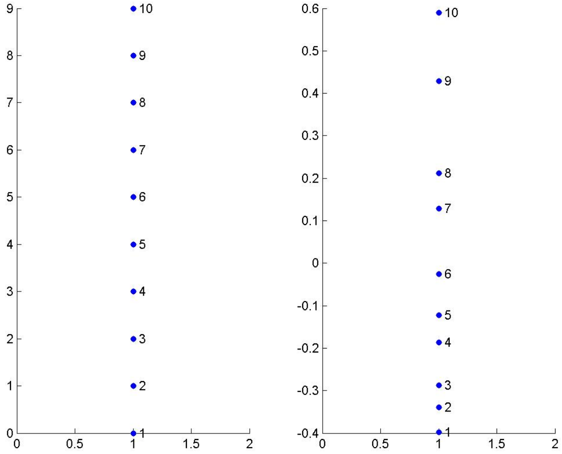


Figure 1 : cutoff=2, degree: [4,2,3.4], noise = 20%

Toy Example 1

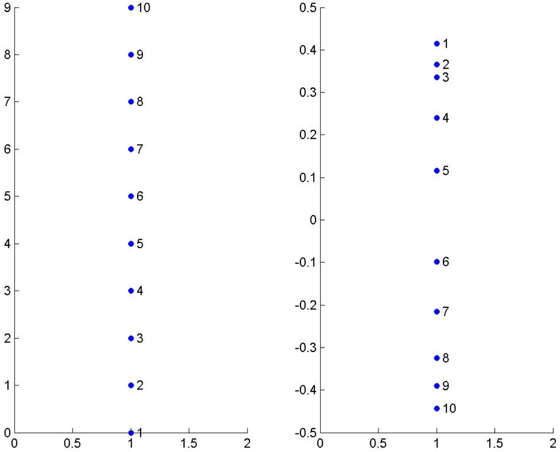
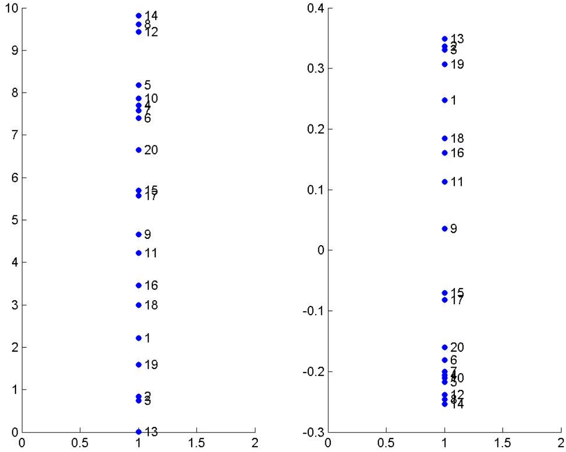
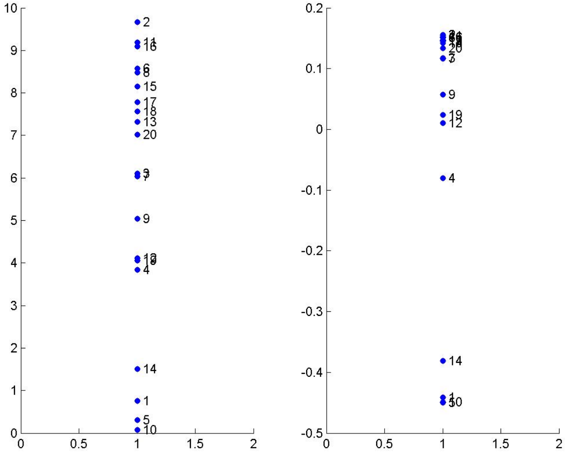


Figure 2 : cutoff=4, degree: [8,4,6], noise = 20%

Random Example 1



Random Example 2



Toy Example 2

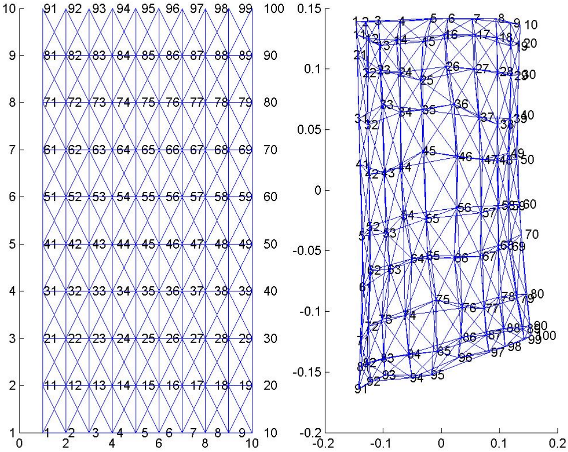


Figure 3 : cutoff=2, degree: [12,5,10], noise = 20%

Toy Example 2

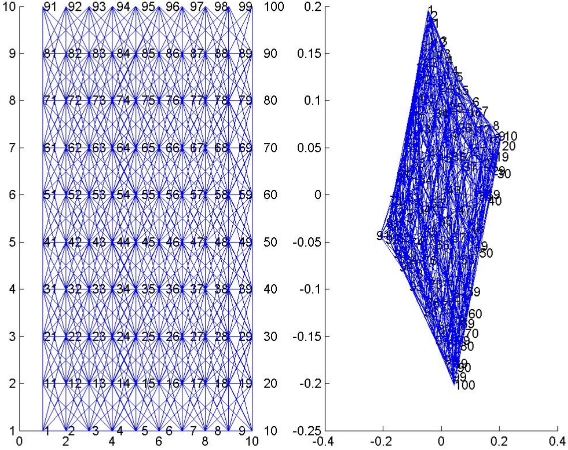
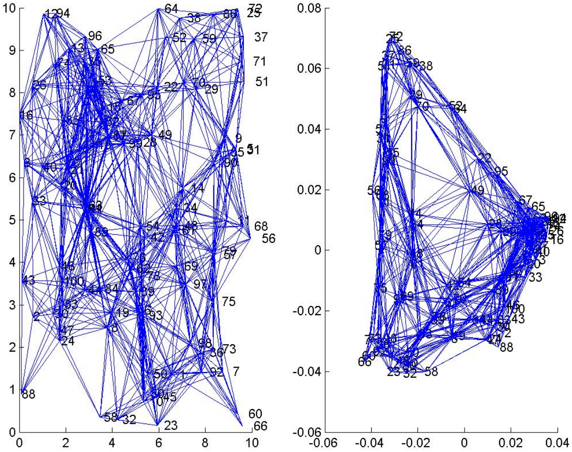
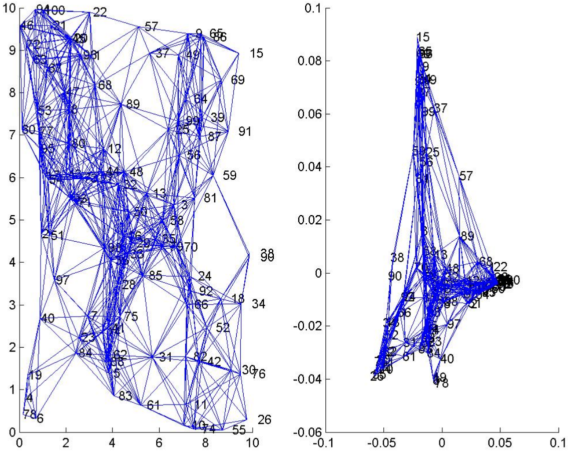


Figure 4 : cutoff=3, degree: [20,10,21.2], noise = 20%

Random Example 3



Random Example 4



Observations:

- With or without D, they make no big difference.
- The results vary slightly as $\epsilon/k, t$ change.
- The result is reasonably robust w.r.t the noise in distances.
- It performs better in "uniform" graphs.
- Sometimes several points collapse to stay together, which is due to the way we construct distances.
- It does not give the right coordinates, but meaningful topology information.

Algorithm framework

Laplacian Eigenmap-based Algorithm for DGP

- Step 1 Construct the Laplacian matrix L and calculate its eigenvalue decomposition $L = VDV^T$ such that the eigenvalues in D are in ascending order.
- Step 2 Set $X0 = V(:, 2 : m + 1)$ and scale it to the proper magnitude.
- Step 3 Using $X0$ as the initial point, apply error function minimization method to further improve the result.

Scale

Note that the coordinates from Eigenvalue decomposition only reveals the topology, but not in the right magnitude, therefore scaling the points is necessary.

1. Construct a new distance matrix $\tilde{D}$ according to $X0$ and E.
2. Calculate
$$ratio = \frac{sum(D)}{sum(\tilde{D})}.$$
3. Set $X0 = X0 * ratio$.

Toy Example 2

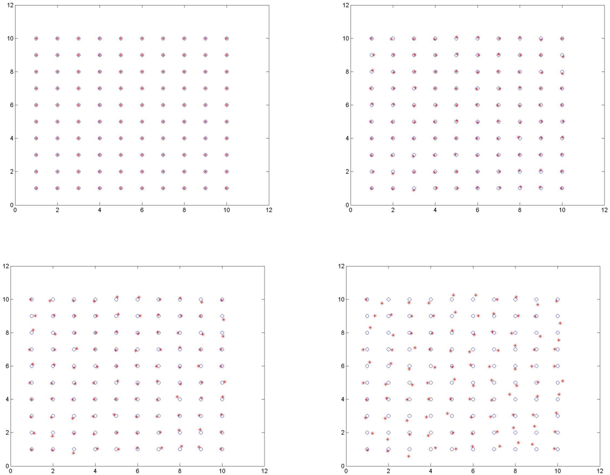


Figure 5 : cutoff = 2, noise = 0, 5%, 10%, 20%

Random Example

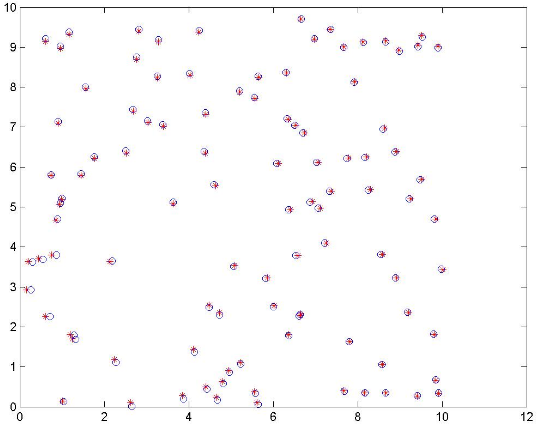


Figure 6 : cutoff = 3, noise = 0

Random Example

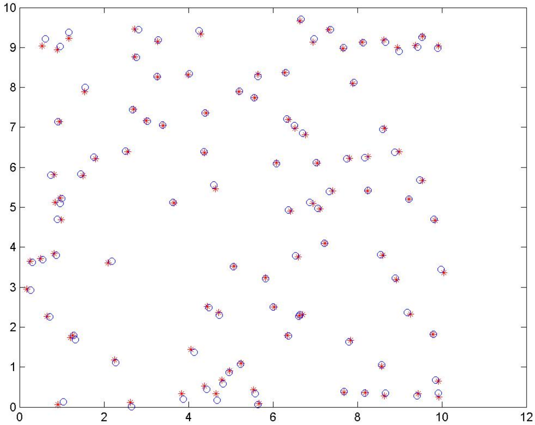


Figure 7 : cutoff = 3, noise = 5%

Random Example

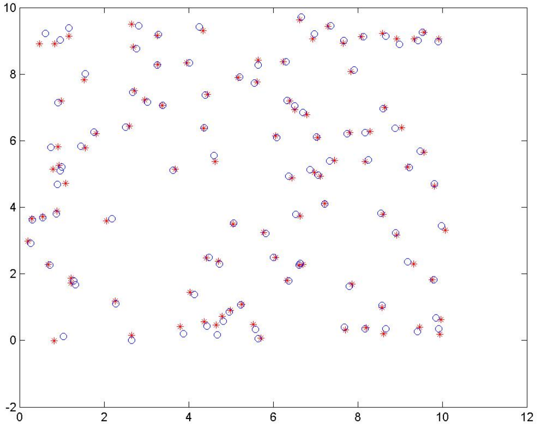


Figure 8 : cutoff = 3, noise = 10%

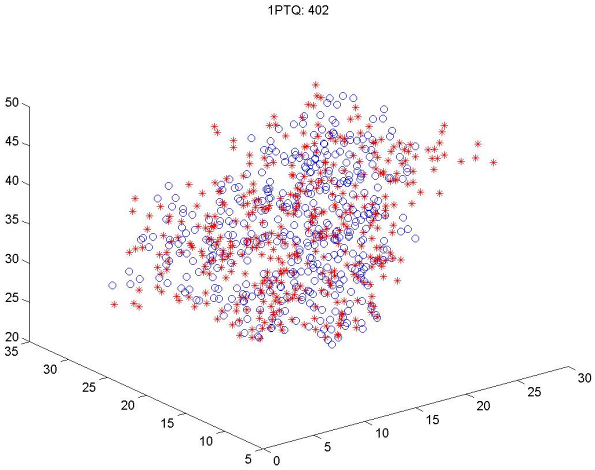


Figure 9 : 1PTQ:402, cutoff = 5, noise = 0

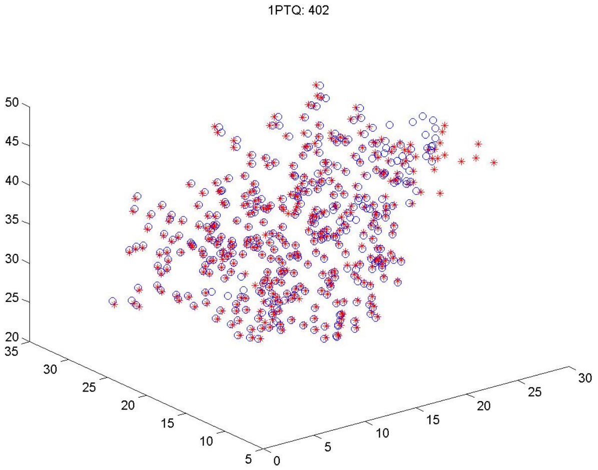


Figure 10 : 1PTQ:402, cutoff = 6, noise = 0

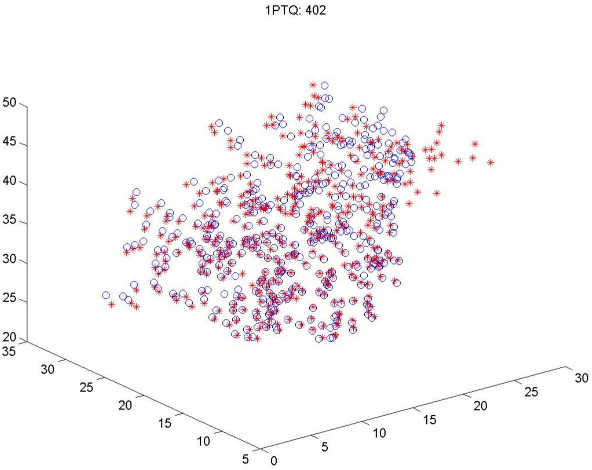


Figure 11 : 1PTQ:402, cutoff = 6, noise = 5%

Outline

3. Conclusion and Future Work

Conclusion and Future Work

- Conclusion
  - Proposed a Laplacian Eigenmap-based algorithm for DGP.
  - Finished some very preliminary numerical experiments.
- Future Work
  - Figure out the problem with protein computation.
  - Carefully study the parameters in Laplacian matrix.
  - Eliminate some edges to reduce the computational complexity in non-linear programming.
  - Develop new error functions to model the problem better.
  - Try to develop distributed algorithm based Laplacian Eigenmap.

Thank you for your attention!

szl@lsec.cc.ac.cn