A regularized SDP based approach for anchor free 3D graph localization
Kim-Chuan TOH
Department of Mathematics, NUS

http://www.math.nus.edu.sg/~mattohkc

joint work with:
Pratik Biswas (Stanford U.)
Yinyu Ye (Stanford U.)
Outline

- Graph localization problem
- Geometric build-up algorithm
- Semidefinite programming (SDP) based algorithm
- Performance
- Regularized SDP
- Distributed algorithm
- Performance on protein molecules
- Work in progress
3D graph localization problem (GLP)

Input: Given $n$ atoms $x_1, \ldots, x_n \in \mathbb{R}^3$, and Euclidean distances $d_{ij}$ between $x_i, x_j$ for some pairs $(i, j) \in \mathcal{E}$.

GLP: Find the positions $x_1, \ldots, x_n \in \mathbb{R}^3$ such that

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

Aim: Find a robust, accurate, and efficient algorithm to solve the problem.

Example. In molecular conformation of a molecule, some inter-atomic distances are found from X-ray diffraction or NMR experiments, re-construct the 3D coordinates of the atoms satisfying the distance constraints. Here, $\mathcal{E} = \{(i, j) : \|x_i - x_j\| \leq R\}$, where $R = 6\text{Å}$.

Similar set up in ad-hoc wireless sensor network.

Related problems: Metric distance embedding, dimensionality reduction in machine learning, kernel estimation in statistics, fastest mixing Markov chains, Euclidean ball packing, ...
**Complexity**

- If all pairwise distances are given, i.e., $\mathcal{E}$ is complete $= \{(i, j) : 1 \leq i < j \leq n\}$, then GLP is solvable in $O(n^3)$ using eigenvalue decomposition as follows.

  WLOG, assume $x_1 = 0$. Then $\|x_j\| = d_{j1}$, $j = 2 : n$, and

  $$x_i^T x_j = \frac{1}{2}(\|x_i\|^2 + \|x_j\|^2 - d_{ij}^2) =: H_{ij}, \quad 2 \leq i, j \leq n.$$

  Let $X = [x_2, \ldots, x_n] \in \mathbb{R}^{3 \times (n-1)}$. Then

  $$X^T X = H.$$

  Compute the eigenvalue decomposition $H = U \text{diag}(\sigma) U^T$, where $U \in \mathbb{R}^{(n-1) \times 3}$, $\sigma \in \mathbb{R}^3$.

  Set $X = \text{diag}(\sqrt{\sigma}) U^T$. This technique is called multi-dimensional scaling (MDS).

  **Note:** the configuration is determined up to translation, rotation, and reflection.

- If $\mathcal{E}$ is incomplete, GLP is NP-hard [Saxe '79]

  Many early algorithms are based on completing the partial distance matrix $D = (d_{ij})$ and then estimate $x_1, \ldots, x_n$ via MDS. Some complete the missing distances randomly; others complete $D$ by shortest-path distances, etc.
Partial Literature on molecular conformation

- EMBED algorithm by Crippen and Havel (’88); complete $D +$ MDS, followed by minimizing the sum of distance errors: $\sum_{(i,j) \in E} \|x_i - x_j\|^2 - d_{ij}^2$

- Partial metrization by Brunger et al (’93, ’98)

- Glunt and Hayden (’01); similar to EMBED but complete $D$ by nearest Euclidean distance matrix via alternating projection

- Alfakih et al, Trosset, also complete $D$ by nearest Euclidean distance matrix

- Graph reduction by Hendrickson (’91, ’95)

- Global optimization by Moré and Wu (’96,’97); more refined one by Williams et al (’01) using domain knowledge such as minimum separation distances between atoms

- Geometric build-up algorithm by Dong and Wu (’03,’05)
Partial literature in ad-hoc wireless sensor network

Here the networks have a small number of known points called anchors. Besides Euclidean distances, sensors may collect angle information from some triplets.

• There is an explosion of research on sensor network localization.

See Hightower and Boriello ('01); Ganesan et al ('02); Bulusu et al ('00); Doherty et al ('01); Howard et al ('01); Niculescu and Nath ('01); Savarese et al ('02); Shang and Ruml ('03); Biswas et al ('03'); Tseng ('04'); ...

Note: In sensor networks, distances are always quite noisy
Geometric build up algorithm of Dong and Wu

Assume 2D for convenience. A point can be determined by its distance to 3 known points that are not on a line.

Suppose the pairwise distance between atom $j$ and the first 3 atoms are given.

\[
\begin{align*}
\|x_j\|² - 2x_1^T x_j + \|x_1\|² &= d_{j1}² \\
\|x_j\|² - 2x_2^T x_j + \|x_2\|² &= d_{j2}² \\
\|x_j\|² - 2x_3^T x_j + \|x_3\|² &= d_{j3}².
\end{align*}
\]

(2)−(1) and (3)−(2) give

\[
\begin{align*}
2(x_1 - x_2)^T x_j + \|x_2\|² - \|x_1\|² &= d_{j2}² - d_{j1}² \\
2(x_2 - x_3)^T x_j + \|x_3\|² - \|x_2\|² &= d_{j3}² - d_{j2}²
\end{align*}
\]

In matrix form:

\[
\begin{bmatrix}
(x_1 - x_2)^T \\
(x_2 - x_3)^T
\end{bmatrix}
\begin{bmatrix}
x_j
\end{bmatrix} = \text{rhs}
\]

\[
A
\]

The $2 \times 2$ matrix $A$ is nonsingular if $x_1, x_2, x_3$ are not on a line ⇒ $x_j$ is uniquely determined.
Non-convex quadratic model

Not easy to adapt geometric based algorithms for GLP with noisy distances. Easier to adapt optimization based approach.

Let \( X = [x_1, \ldots, x_n] \).

\[
\min_{X \in \mathbb{R}^{3 \times n}} \left\{ \sum_{(i,j) \in \mathcal{E}} \left| \|x_i - x_j\|^2 - d_{ij}^2 \right| \right\} \quad \text{(non-convex!)}
\]

Important to eliminate the translational degree of freedom by adding the constraint: \( \|Xe\| = 0 \) (center of gravity at origin).

Most previous approaches try to use some form of global optimization methods to solve variants of the above problem.

Can write (NCQP) as:

\[
\min_{X \in \mathbb{R}^{3 \times n}} \left\{ \sum_{(i,j) \in \mathcal{E}} \left| \alpha_{ij} - d_{ij}^2 \right| : \|x_i - x_j\|^2 = \alpha_{ij} \right\}
\]

Doherty et al., and Tseng relax "\( = \)" to "\( \leq \)" to get a convex quadratic problem, but relaxation can be weak.
Matrix representation and SDP relaxation

Recall: $X = [x_1, \ldots, x_n]$. Let $e_{ij} = e_i - e_j$. Then $x_i - x_j = X e_{ij}$,

$$\|x_i - x_j\|^2 = e_{ij}^T X^T X e_{ij} = \langle e_{ij} e_{ij}^T, X^T X \rangle = \langle A_{ij}, Y \rangle,$$

Can rewrite (NCQP) as:

$$\min_{X \in \mathbb{R}^{3 \times n}} \left\{ \sum_{(i,j) \in E} | \langle A_{ij}, Y \rangle - d_{ij}^2 | : Y = X^T X \right\}$$

Relax $Y = X^T X$ to $Y \succeq 0$:

$$(SDP) \quad \min \left\{ \sum_{(i,j) \in E} | \langle A_{ij}, Y \rangle - d_{ij}^2 | : Y \succeq 0 \right\}$$

- If the SDP solution has rank 3, then GLP is solved! [So, Ye]
- (SDP) is convex! $Y \in \mathbb{R}^{n \times n}$, number of constraints = $|E| + 1$.
- Can be efficiently solved by interior-point methods for problems with $n \leq 100, |E| \leq 2000$. 
SDP based algorithm

- Solve (SDP). Let $Y$ be the SDP solution.
- Compute eigenvalue decomposition $Y = U \text{diag}(\sigma) U^T$. Set $X^0 = \text{diag}(\sqrt{\sigma_3}) U_3^T$.
- Use SDP estimated solution $X^0$ as the starting iterate, refine it by applying a gradient descent method to minimize ($NCQP$).

Computational cost: $10 \sim 20 |\mathcal{E}|^3$ operations and $|\mathcal{E}|^2$ memory space.
**Performance**

Assume 2D for convenience. Experimental set up:

- Generate \( n \) random points \( \hat{X} = [\hat{x}_1, \ldots, \hat{x}_n] \) in the unit square \([-0.5, 0.5] \times [-0.5, 0.5]\).
- Set \( \hat{d}_{ij} = \|\hat{x}_i - \hat{x}_j\| \) if \( \leq 0.3 \).
- Add noise: \( d_{ij} = \hat{d}_{ij}(1 + z \ast nf) \), where \( z \sim N(0, 1) \), \( nf \in [0, 1] \) is the noise factor.
- Apply the SDP based algorithm to the distance data \( D = (d_{ij}) \). Let \( X = [x_1, \ldots, x_n] \) be the estimated positions.
- Accuracy measure:

\[
RMSD = \frac{1}{\sqrt{n}} \min \{\|\hat{X} - QX\|_F : Q \in IR^{2 \times 2 \text{ orthogonal}}\}
\]
Performance continued

Distance data without noise: $n_f = 0$.

![Graph showing distance data without noise and refinement results with RMSD values.](image)
Performance continued

Distance data with 30% noise: $nf = 0.3$.

Note: estimated points tend to crowd together.
**Regularization**

To prevent crowding, we maximize the pair-wise separations by adding the following regularization term to (NCQP):

\[-\frac{\lambda}{2n} \sum_{j=1}^{n} \sum_{i=1}^{n} \|x_i - x_j\|^2 = -\lambda \langle X^T X, I - ee^T / n \rangle.\]

The corresponding term to add to (SDP) is \(-\lambda \langle Y, I - ee^T / n \rangle\).

Distance data with 30% noise: with regularization.
Eigenvalues of $Y$ versus $\lambda$

80 random points on unit square: $nf = 0.3$
RMSD versus $\lambda$

Work in progress: automatic selection of $\lambda$.

One possible criterion:

$$\lambda = \frac{\sum_{(ij) \in \mathcal{E}} | \langle A_{ij}, Y \rangle - d_{ij}^2 |}{\langle Y, I - ee^T/n \rangle}$$
**A distributed algorithm**

When $|\mathcal{E}|$ is large, (SDP) is too large to be solved at one go.

(1) Break up the graph into several overlapping subgraphs (via symmetric re-ordering)
(2) Solve the GLP for each subgraphs
(3) Stitch the subconfigurations together by matching the overlapping components.

Main obstacles:
(a) how to handle a subgraph that is not uniquely localizable in 3D
(b) how to decide whether a point is accurately estimated to act as anchor for stitching
(c) how to minimize propagation of errors
Performance on protein molecules

Distance data without noise.

\[ \mathcal{E} = \{(i, j) : \|\widehat{x}_i - \widehat{x}_j\| \leq 6\text{\AA}\} \]

Protein molecule 1PTQ from Protein Data Bank: \( n = 402, |\mathcal{E}| = 7085 \)
Comparison using PDBviewer
### Results on protein molecules

| PDB ID | no. of atoms | % of pairwise dist. given | |ε| | RMSD(Å) | CPU time (secs) |
|--------|--------------|---------------------------|------|-----|---------|-----------------|
| 1PTQ   | 402          | 8.79                      | 7.1e3| 2.2e-6| 121.3   |
| 1HOE   | 558          | 6.55                      | 1.1e4| 4.2e-5| 318.9   |
| 1TJO   | 5459         | 0.72                      | 2.1e5| 1.6e-5| 3019.6  |
| 1NFB   | 5666         | 0.65                      | 2.1e5| 2.3e-4| 3010.5  |
| 1MQQ   | 5681         | 0.75                      | 2.4e5| 9.2e-6| 3149.7  |
| 1NF7   | 6880         | 0.54                      | 2.6e5| 1.0e-2| 3360.6  |
| 1HMV   | 7398         | 0.52                      | 2.8e5| 1.8e-2| 3946.5  |
| 1l7W   | 8629         | 0.47                      | 3.5e5| 3.5e-4| 4664.3  |
CPU time

The diagram shows the relationship between CPU time (in seconds) and the number of atoms. The data points form a linear trend, indicating that CPU time increases proportionally with the number of atoms.
Distributed algorithm on distance data with noise

1PTQ: same set up as before, but add 10% noise to distance data.
1MQQ: $n = 5681$, add 10% noise to distance data.
Work in progress and future work

(1) Automatic selection of regularization parameter

(2) Robust distributed algorithm for 3D GLP with noisy data

(3) How does the SDP estimated solution depend on the noise in $d_{ij}$?

We derived a simple upper bound (UB) on expected optimal SDP objective value. When noises $\sim N(0, \tau^2)$, $UB = (1.6\tau + \tau^2) \sum \hat{d}_{ij}^2$. Empirical results: RMSD $\sim \sqrt{\tau}$.

(4) Solving large SDPs from kernel estimation [Wahba et al] and nonlinear dimensionality reduction [Saul et al]; fastest mixing Markov chains [Boyd, Xiao]

$$
\min \left\{ \sum_{(i,j) \in E} | \langle A_{ij}, Y \rangle - d_{ij}^2 | \pm \lambda \langle I, Y \rangle : Y \succeq 0 \right\}
$$

Here $d_{ij}$ measures the dissimilarity between objects $i$ and $j$. Dimension of localization is not determined a priori, so distributed algorithm may not be suitable. Krylov-subspace methods based interior-point algorithms seem viable.