Based on
Atilla Sit, solving distance geometry problems for protein structure determination
Lit Review 3.3, Chapter 4, Chapter 5.2

MXB 361 Week 4 Notes

Given set of distances find the nodes

1. Consider 1 D problem
   Find $\{x_i\}$ s.t $|x_i - x_j| = d_{ij}$, $i,j \in S$

Use this to show distance problem NP hard if sparse data.
- non deterministic polynomial time
- It is suspected no polynomial time algorithms for NP hard problems. Not proved.

Integer partition problem
$A = (s_1, \dots, s_n)$, $s_i > 0$ integers
Find 2 subsets S_1, S_2 of A s.t
$$\sum_{j \in S_1} s_j = \sum_{j \in S_2} s_j$$

Theorem: 1 D problem $\Leftrightarrow$ integer set partition problem (NP hard)
Proof: $n+1$ nodes, $d_{j\,j+1} = s_j$ $j = 1, \dots, n$, $d_{1\,n+1} = 0 \Rightarrow x_{n+1} = x_1$
$$\sum_{j=1}^{n} (x_{j+1} - x_j) = x_{n+1} - x_1 = 0$$
But $|x_{j+1} - x_j| = s_j$
$$S_1 = \{j : x_{j+1} - x_j = s_j\}, \quad S_2 = \{j : x_{j+1} - x_j = -s_j\}$$
$$\sum_{j \in S_1} s_j - \sum_{j \in S_2} s_j = \sum_{j=1}^{n} (x_{j+1} - x_j) = 0$$

ε Approximation problem
$$\big|\, \|x_i - x_j\| - \delta_{ij} \big| \le \varepsilon_{ij}$$

2. Build up algorithm: one node at a time LLS (Linear Least Squares)
   Find $k+1$ nodes not in same hyperplane
   Determine coords using our previous SVD (use x_n pinned)
   Repeat
   For each undetermined node if it has k distances to k determined nodes not in same plane, find node by LLS (modified)
   end
   end

Suppose k nodes determined
d_{ij} distances from $i = 1, \dots, k$ to undetermined j then
$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{ij}^2, \quad i = 1, \dots, k$$

Subtract eqn i from eqn $i+1$ $i=1,\ldots,l-1$ to eliminate quadratic terms for x_j

$$-2(x_{i+1}-x_i)^T x_j = (d^2_{i+1\,j}-d^2_{ij})-(\|x_{i+1}\|^2-\|x_i\|^2),\ i=1,\ldots,l-1$$

or $A x_j = b_j$

$$A=-2\begin{bmatrix}(x_2-x_1)^T\\ \vdots\\ (x_l-x_{l-1})^T\end{bmatrix},\quad b=\begin{bmatrix}d^2_{2j}-d^2_{1j}-(\|x_2\|^2-\|x_1\|^2)\\ \vdots\\ d^2_{lj}-d^2_{l-1\,j}-(\|x_l\|^2-\|x_{l-1}\|^2)\end{bmatrix}$$

or $A^TA x_j = A^Tb$ (LLS)

Note i) $A=(l-1)\times k$, $A^TA=k\times k$

ii) Solve $\approx n$ linear problems of dimension k

iii) whereas SVD approach $O(n^2)$

iv) if A full rank, A^TA nonsingular

v) This holds if at least $k+1$ of l determined nodes not in same hyperplane

3. Noisy problem

$$d_{ij}=d_{ij}(1+2\theta(\tfrac{1}{2}-\text{rand})),\quad \text{rand}\in U[0,1]$$

4. Build up algorithm with NLS

LLS Buildup may not be stable, as errors may grow.

Idea: Use not only l distances from l nodes to unknown node but use all distances among l determined nodes.

5. Another approach

- Find $k+1$ nodes not in same hyperplane, $x_1,\ldots x_{k+1}$.

Want to find x_j.

$$\|x_i\|^2-2x_i^Tx_j+\|x_j\|^2=d_{ij}^2,\quad i=1,\ldots,k+1$$

- Subtract eqn i from eqn $i+1$, $i=1,\ldots,k$
- Write as a linear system, similar to before

$$Ax_j=b$$

$$A=-2\begin{bmatrix}(x_2-x_1)^T\\(x_3-x_2)^T\\ \vdots\\(x_{k+1}-x_k)^T\end{bmatrix}\qquad b=\begin{bmatrix}d_{2j}^2-d_{1j}^2-(\|x_2\|^2-\|x_1\|^2)\\ \vdots\\ d_{k+1\,j}^2-d_{kj}^2-(\|x_{k+1}\|^2-\|x_k\|^2)\end{bmatrix}$$

- Since $x_1, \dots x_{k+1}$ not in same hyperplane
  A is nonsingular, $\therefore x_j$ is unique.
- Repeat - ie for each undetermined node if the node has $k+1$ distances to $k+1$ determined nodes not on same hyperplane solve $Ax_{new} = b$, until no node can be determined.
- Cheap has $n-(k+1)$ solves of linear system dimension k. So $O(n)$ algorithm.
- BUT errors build up and later nodes may be inconsistent.