Some Topics in Distance Geometry

Zhenli Sheng (盛镇醴)

Email: szl@lsec.cc.ac.cn

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

November 5th, 2013
seminar talk

Outline

Outline

Distance Geometry Problem

Given an integer d, find $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, such that

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

or

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

where S is the given index set.

- In general, it is a NP-hard problem.
- It has many applications:
  - protein structure determination
  - sensor network localization
  - multidimensional scaling
  - data mining: distance $\Leftrightarrow$ similarity
  - ......
- In most of the applications, we have local and sparse distances.

Outline

2. **Various Error Functions**

How to Compare Different Models?

Usually, application problems may have several different models, a natural question is: which model is the best?

How to Compare Different Models?

Usually, application problems may have several different models, a natural question is: which model is the best?

- How to define goodness?

$$RMSD(X,Y) = \min_{Q,T}\{\|X - YQ - T\|_F/\sqrt{n} : Q^{\mathrm{T}}Q = I\}$$

- result = model + algorithm

How to Compare Different Models?

Usually, application problems may have several different models, a natural question is: which model is the best?

- How to define goodness?

$$RMSD(X, Y) = \min_{Q,T}\{\|X - YQ - T\|_F/\sqrt{n} : Q^{\mathrm{T}}Q = I\}$$

- result = model + algorithm

In our case, some possible ways:

- Smoothness
- Number of "local minimizer"
- Numerical tests on the same instances

Numerical Tools

- Ipopt ((**I**nterior **P**oint **OPT**imizer) is a software package for large-scale non-linear optimization. It is designed to find (local) solutions of mathematical optimization problems of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_L \leq g(x) \leq g_U \\ & x_L \leq x \leq x_U \end{aligned} \tag{1}$$

  where $f(x)$ and $g(x)$ should be twice continuously differentiable.
- AMPL: A Modeling Language for Mathematical Programming. They can provide first-order and second-order information via automatic differential.

```
param n;
set Vertex := {1..n};
set Edge within Vertex cross Vertex;
param Dist {Edge} >= 0;
param startvar {1..n,1..3};
var x {1..n,1..3};
minimize Smoothed_Stress_function:
   sum{(i,j)in Edge} ((x[i,1]-x[j,1])^2 + (x[i,2]-x[j,2])^2 +
(x[i,3]-x[j,3])^2 - Dist[i,j]^2)^2;
```

Error Functions

- Stress function

$$Stress(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} (\|x_i - x_j\| - d_{ij})^2$$

- Smoothed Stress function

$$SStress(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} (\|x_i - x_j\|^2 - d_{ij}^2)^2$$

- Absolute Error function

$$AbsErr(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} \left|\|x_i - x_j\|^2 - d_{ij}^2\right|$$

Stress VS SStress

- *SStress* is smoother, thus converges fast

$$\frac{\partial Stress}{\partial x_i} = \sum_{j \in N(i)} 2\left(1 - \frac{d_{ij}}{\|x_i - x_j\|}\right)(x_i - x_j) \tag{2}$$

$$\frac{\partial SStress}{\partial x_i} = \sum_{j \in N(i)} 4(\|x_i - x_j\|^2 - d_{ij}^2)(x_i - x_j) \tag{3}$$

- Numerical tests: 100 random points in $[0,1]^3$, with different cutoffs, noiseless, random initial point.

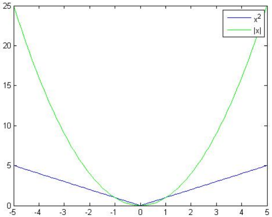
cutoff	per	SStress			Stress		
		iter	time[1]	RMSD	iter	time	RMSD
2	100	20	0.42	5.15-03	22	0.47	5.14-03
0.9	83.8	20	0.44	5.09-03	62	1.33	5.04-03
0.8	71.6	16	0.31	5.17-03	60	1.10	4.87-03
0.7	57.9	25	0.34	3.36-01	73	1.29	5.28-03
0.6	43.1	34	0.37	2.59-01	43	0.53	2.59-01
0.65	50.6	37	0.54	5.37-03	39	0.54	4.25-01

- We also have tested on some other random data and with different noise, it is very hard to say which model leads to better result, but *SStress* converges faster.

[1] ipopt

SStress VS AbsErr

- *AbsErr* is more robust with large outliers.



- *SStress* is twice continuously differentiable, while *AbsErr* not. Therefore *SStress* is used in direct error minimization.
- *AbsErr* is widely used in SDP-based models with techniques as below,
  - $\|x_i - x_j\|^2 = e_{ij}^T Y e_{ij}$, where $Y = X^T X$.
  - $|x| = x_+ - x_-$ with $x_+ \geq 0, x_- \geq 0$.

Two New Error Functions

We construct the following two new error functions to measure the **Quo**tient **Err**ors, rather than difference errors,

$$QuoErr(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} h\left(\frac{\|x_i - x_j\|^2}{d_{ij}^2}\right)$$

where

$$h(x) = \begin{cases} \frac{1}{2}(x-1)^2 & \text{if } x \geq 1, \\ x - 1 - log(x) & \text{if } 0 < x < 1. \end{cases}$$

or

$$h(x) = x - 1 - log(x) \quad \forall x > 0.$$

Here, we choose the second function, the gradient can be calculated as

$$\frac{\partial QuoErr}{\partial x_i} = \sum_{j\in N(i)} \left(\frac{1}{d_{ij}^2} - \frac{1}{\|x_i - x_j\|^2}\right)(x_i - x_j)$$

which is not continuous at $x_i = x_j$.

SStress VS QuoErr

- Random intial point, noiseless

cutoff	per	SStress			QuoErr		
		iter	time	RMSD	iter	time	RMSD
2	100	20	0.42	5.15-03	44	0.94	5.22-03
0.9	83.8	20	0.44	5.09-03	70	1.46	5.31-03
0.8	71.6	16	0.31	5.17-03	57	1.04	2.69-01
0.7	57.9	25	0.34	3.36-01	73	1.29	5.02-03
0.6	43.1	34	0.37	2.59-01	84	1.04	5.09-03
0.65	50.6	37	0.54	5.37-03	85	1.17	5.49-02

- Initial point: X_{true}*(1+0.5*(rand-0.5)*2), noise = 0.1

cutoff	per	SStress			QuoErr		
		iter	time	RMSD	iter	time	RMSD
2	100	8	0.15	3.48-02	13	0.29	1.81-02
0.9	83.8	8	0.16	3.29-02	21	0.45	1.86-02
0.8	71.6	8	0.18	2.93-02	15	0.27	1.93-02
0.7	57.9	10	0.15	3.00-02	23	0.35	2.20-02
0.6	43.2	16	0.20	2.89-02	21	0.25	2.17-02
0.5	28.8	10	0.08	3.25-02	28	0.23	2.58-02
0.4	16.4	16	0.07	3.99-02	48	0.21	6.57-02
0.3	7.6	45	0.10	1.66-01	39	0.09	1.62-02

- Usually, the QuoErr with not squared distance can get smaller RMSD error.

Short Summary

- *SStress* is fairly a good choice for postprocess procedure.
- *QuoErr* provides better results when the distances are contaminated by noise, but it takes more time usually.
- In successful cases, we can get smaller objective function value than putting the true positions into these functions.
  $\Rightarrow$ If we expect better RMSD result, new model is needed.
- AbsErr is used in SDP based models.
- All of the models succeed or fail at the similar level.

Hessian Matrix

Hessian Matrix of SStress

Let $x = (x_{11}, x_{12}, x_{13}, \ldots, x_{i1}, x_{i2}, x_{i3}, \ldots, x_{n1}, x_{n2}, x_{n3})$, then the Hessian Matrix $\in \mathbb{R}^{3n \times 3n}$ can be specified by

i) **(diagonal item)** for $i = 1 : n$, $p = 1 : 3$,

$$\frac{\partial^2 SStress}{\partial^2 x_{ip}} = \sum_{j \in N(i)} 4 \left(2(x_{ip} - x_{jp})^2 + \|x_i - x_j\|^2 - d_{ij}^2 \right) \tag{4}$$

ii) **(nondiagonal item of diagonal block)** for $i = 1 : n$, $p, q \in \{1, 2, 3\}$ with $p \neq q$,

$$\frac{\partial^2 SStress}{\partial x_{ip} \partial x_{iq}} = \sum_{j \in N(i)} 8(x_{ip} - x_{jp})(x_{iq} - x_{jq}) \tag{5}$$

iii) **(diagonal item of nondiagonal block)** for $i = 1 : n$, $k \in N(i)$, $p = 1 : 3$,

$$\frac{\partial^2 SStress}{\partial x_{ip} \partial x_{kp}} = -8(x_{ip} - x_{kp})^2 - 4(\|x_i - x_k\|^2 - d_{ik}^2) \tag{6}$$

iv) **(nondiagonal item of nondiagonal block)** for $i = 1 : n$, $k \in N(i)$, $p, q \in \{1, 2, 3\}$ with $p \neq q$,

$$\frac{\partial^2 SStress}{\partial x_{ip} \partial x_{kq}} = -8(x_{ip} - x_{kp})(x_{iq} - x_{kq}) \tag{7}$$

v) 0, otherwise.

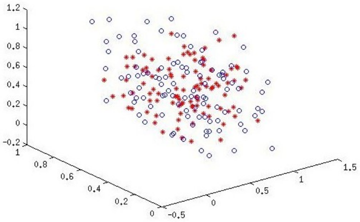
Picture of Hessian Matrix

$$\begin{pmatrix} \diamondsuit & & \square & & \square \\ & \diamondsuit & & & \square \\ \square & & \diamondsuit & & \\ \vdots & \vdots & & \ddots & \vdots \\ \square & \square & & & \diamondsuit \end{pmatrix}$$

- Hessian is a sparse symmetric matrix.
- Each $\diamondsuit, \square$ represents a 3×3 symmetric submatrix.
- The (i, j)-th block is zero matrix if $(i, j) \notin S$.
- The diagonal $\diamondsuit$ is the negative sum of $\square$ in the same row/column.

Regularization Term for Error Function

- "The points estimated from these models tend to crowd together towards the center of the configuration. "[2]



- They further proposed to add a regularization term to the objective function

$$\lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \|x_i - x_j\|^2$$

However, it is very difficult to choose the parameter λ.

[2] Biswas, P., Liang, T. C., Toh, K. C., Ye, Y., & Wang, T. C. (2006). *Semidefinite programming approaches for sensor network localization with noisy distance measurements*.

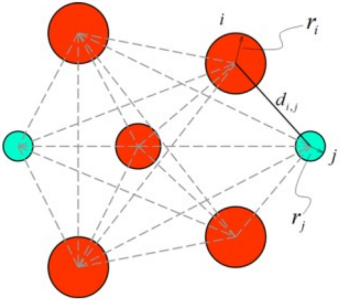
Outline

Generalized DG Model

Sit and Wu (2011) [3] proposed the following model to deal with distance bounds,

$$\begin{aligned} \max_{x_i, r_i} \quad & \sum_{i=1}^{n} r_i \\ \text{s.t.} \quad & \|x_i - x_j\| + r_i + r_j \le u_{i,j} \\ & \|x_i - x_j\| - r_i - r_j \ge l_{i,j}, \quad \forall (i,j) \in S \\ & r_i \ge 0, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{8}$$

where x_i is the coordinate, r_i is the fluctuation radius.



[3] Sit, A., & Wu, Z. (2011). *Solving a generalized distance geometry problem for protein structure determination.*

SDP Relaxation

Let $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{3\times n}$ and e_i be the i-th column of Identity matrix in appropriate dimensional space. Then, we have

$$\begin{aligned} \|x_i - x_j\|^2 &= \|X(e_i - e_j)\|^2 \\ &= e_{ij}^T X^T X e_{ij} \qquad (e_{ij} = e_i - e_j) \end{aligned} \tag{9}$$

Furthermore, let $R = (r_1, r_2, \ldots, r_n) \in \mathbb{R}^{1\times n}$ be a row vector,

$$\begin{aligned} (u_{ij} - r_i - r_j)^2 &= u_{ij}^2 - 2u_{ij}(r_i + r_j) + (r_i + r_j)^2 \\ &= u_{ij}^2 - 2u_{ij}R(e_i + e_j) + (R(e_i + e_j))^2 \\ &= u_{ij}^2 - 2u_{ij}Rf_{ij} + f_{ij}^T R^T R f_{ij} \qquad (f_{ij} = e_i + e_j) \\ &= u_{ij}^2 + \left\langle \begin{pmatrix} 0 & -u_{ij} f_{ij}^T \\ -u_{ij} f_{ij} & f_{ij} f_{ij}^T \end{pmatrix}, \begin{pmatrix} 1 & R \\ R^T & R^T R \end{pmatrix} \right\rangle \end{aligned} \tag{10}$$

To establish the associate SDP model, we let

$$Y = X^T X, \quad Z = \begin{pmatrix} 1 & R \\ R^T & R^T R \end{pmatrix},$$

and relax them to $Y \succeq 0, Z \succeq 0$. Therefore, by these notations, we can relax model (8) to

$$\begin{aligned}
\min_{Y,Z} \quad & -\left\langle \begin{pmatrix} 0 & e^T/2 \\ e/2 & 0 \end{pmatrix}, Z \right\rangle \\
\text{s.t.} \quad & \left\langle e_{ij} e_{ij}^T, Y \right\rangle + \left\langle \begin{pmatrix} 0 & -u_{ij} f_{ij}^T \\ -u_{ij} f_{ij} & f_{ij} f_{ij}^T \end{pmatrix}, Z \right\rangle \leq u_{ij}^2 \\
& \left\langle e_{ij} e_{ij}^T, Y \right\rangle - \left\langle \begin{pmatrix} 0 & l_{ij} f_{ij}^T \\ l_{ij} f_{ij} & f_{ij} f_{ij}^T \end{pmatrix}, Z \right\rangle \geq l_{ij}^2, \quad \forall (i,j) \in S \\
& \left\langle \begin{pmatrix} 0 & e_i^T/2 \\ e_i/2 & 0 \end{pmatrix}, Z \right\rangle \geq 0, \quad i = 1, 2, \ldots, n \\
& \left\langle e_1 e_1^T, Z \right\rangle = 1 \\
& Y \succeq 0, Z \succeq 0.
\end{aligned} \tag{11}$$

Outline

4 Ongoing works

Ongoing works

- Consider corresponding NSDP and ESDP model and facial reduction method to reduce the size of the problem.
- Exploit the extremely sparse structure of the SDP model to design a fast algorithm.

Thank you for your attention!

szl@lsec.cc.ac.cn