Report of Project 1: Sensor Network Localization

Kuangyu Ding, Tianyun Tang

August 5, 2022

Overview

Sensor network localization

Given possible anchors $\{a_k : \ k \in [m]\} \subset \mathbb{R}^d$, distance information d_{ij}, $(i,j) \in N_x$ and $\hat{d}_{kj}$, $(k,j) \in N_a$, find $\{x_i : \ i \in [n]\} \subset \mathbb{R}^d$ for all i such that

$$\begin{aligned} \|x_i - x_j\|^2 &= d_{ij}^2, \ \forall \ (i,j) \in N_x, \ i < j, \\ \|a_k - x_j\|^2 &= \hat{d}_{kj}^2, \ \forall \ (k,j) \in N_a, \end{aligned} \tag{1}$$

where $(i,j) \in N_x$ ($(k,j) \in N_a$) connects points x_i and x_j (a_k and x_j) with an edge whose Euclidean length is d_{ij} ($\hat{d}_{kj}$). N_x and N_a denote the pairs of points whose distance are known.

- Define Root Mean Square Distance (RMSD):

$$RMSD = \frac{1}{\sqrt{n}} \left(\sum_{i=1}^{n} \|\hat{x}_i - x_i\|^2 \right)^{1/2}$$

SOCP relaxation

$$
\begin{aligned}
\min_{x_i} \ & \sum_i 0^\top x_i \\
\text{s.t. } & \|x_i - x_j\|^2 \leq d_{ij}^2, \ \forall\, (i,j) \in N_x, \ i < j, \\
& \|a_k - x_j\|^2 \leq \hat{d}_{kj}^2, \ \forall\, (k,j) \in N_a.
\end{aligned} \tag{2}
$$

Observation: Define $A := \operatorname{conv}(a_1, a_2, \ldots, a_m)$. Suppose (2) has a feasible solution, then there exists a feasible solution $\{x_i\}_{i \in [n]} \subset A$.

SDP relaxation

$$
\begin{aligned}
\min_{Z}\ & 0 \cdot Z \\
\text{s.t. } & Z_{1:d,1:d} = I, \\
& (0; e_i - e_j)(0; e_i - e_j)^\top \cdot Z = d_{ij}^2,\ \forall\ (i,j) \in N_x,\ \ i < j, \\
& (a_k; -e_j)(a_k; -e_j) \cdot Z = \hat{d}_{kj}^2,\ \forall\ (k,j) \in N_a, \\
& Z \succeq 0.
\end{aligned} \tag{3}
$$

Observation: (1) is feasible if and only if (3) has a feasible solution such that $\mathrm{rank}(Z) = d$.

Nonlinear least squares approach

$$\min_{x_i} \sum_{(i,j)\in N_x} \left(\|x_i - x_j\|^2 - d_{ij}^2\right)^2 + \sum_{(k,j)\in N_a} \left(\|a_k - x_j\|^2 - d_{kj}^2\right)^2, \tag{4}$$

which is an unconstraint nonlinear optimization problem.

Question 1

Question 1: Run some randomly generated problems in 2D with 3 or more anchors, respectively, and ten sensors to compare the three approaches. You may set up a threshold radius such that the distance between two points is known when the distance is below the threshold.

Solution: Choose anchors to be $(\pm 0.3, \pm 0.3)$. Randomly generate 10 sensors `X = rand(n,2)-0.5`.

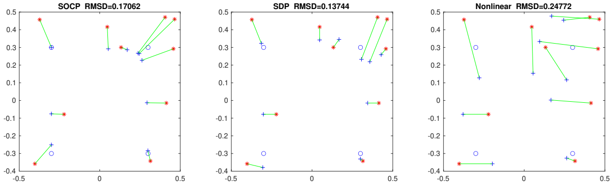


Figure: threshold radius $= 0.4$

Question 1

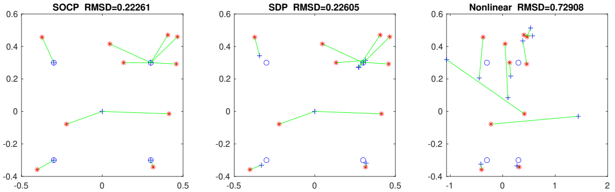


Figure: threshold radius = 0.2

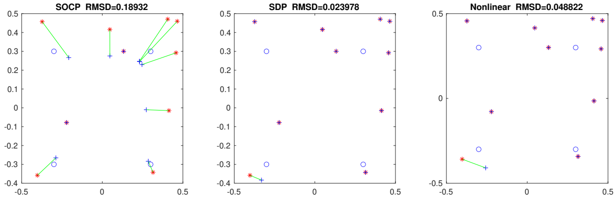


Figure: threshold radius = 0.6

Question 2

Question 2. Based on your comparison in Question 1, you might find that although the time complexity of the SOCP relaxation is fast, one drawback is that it might not be able to localize sensors that are not in the convex hull of anchors. To solve this problem, we can try an SOCP relaxation first and the steepest descent second strategy. That is, we use the SOCP solution of (2) as the initial feasible solution of problem (4). Then, we apply the steepest descent method for some steps to solve (4). Discuss the performance of this strategy and three previous approaches.

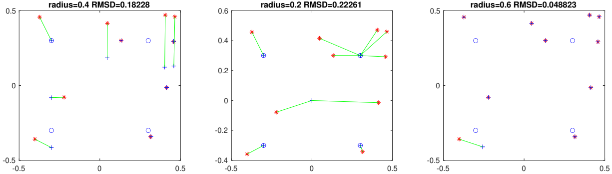


Figure: SOCP+Nonlinear least square

Question 1 and Question 2

Observations:

(1) The predicted sensors of SOCP is always inside the convex hull of anchors.

(2) The performance of Nonlinear least square model relies heavily on the radius threshold.

(3) The behaviour of SDP is the best among all these 3 models in Question 1.

(4) When we combine SOCP with nonlinear least square solver, the model becomes more accurate than using SOCP and Nonlinear least square respectively. However, it is still slightly worse than SDP model.

Question 3

Questions 3. Generate some random problems with slightly noisy sensor data. Use the SDP solution $\bar{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n]$ of (5) as the initial solution for solving model (4) by the Steepest Descent Method for a number steps. Are you able to estimate the position of the sensors well? Compare this to using Steepest Descent on (4) with random initialization.

Solution. We randomly generate 60 datas: `X = rand(60,2)-0.5`. Add noise: `dx = abs(1+ns*randn(n1,1)).*dx; da = abs(1+ns*randn(n2,1)).*da;`

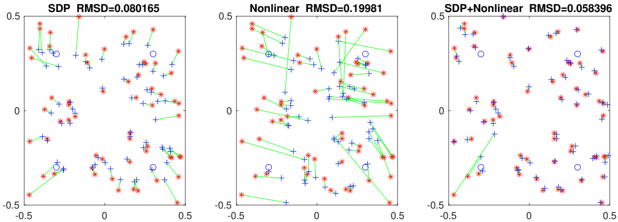


Figure: radius = 0.3, ns = 0.1

Question 3

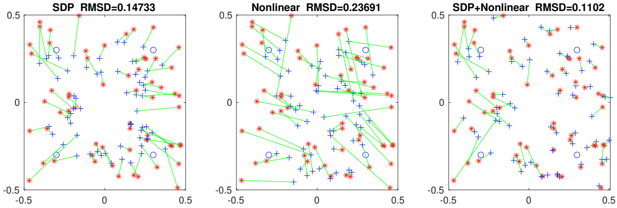


Figure: radius = 0.3, ns = 0.3

Observation SDP+Nonlinear least square is better than Nonlinear least square with random initialization. Nonlinear local search refinement can further improve the accuracy of SDP model.

Question 4

Question 4: Based on the idea in Question 2, develop an SOCP relaxation first and the steepest descent second method for SNL with Noisy Data. Compare this to the algorithm in Question 3.

Solution. The SOCP relaxation for SNL with Noisy data is as follows:

$$\begin{aligned}
\min_{x_j,\delta,\hat{\delta}} \quad & \sum_{(i,j)\in N_x} \delta_{ij} + \sum_{(k,j)\in N_a} \hat{\delta}_{kj} \\
\text{s.t.} \quad & \|x_i - x_j\|^2 - \delta_{ij} \le d_{ij}^2, \ \forall\ (i,j) \in N_x,\ i<j, \\
& \|a_k - x_j\|^2 - \hat{\delta}_{kj} \le \hat{d}_{kj}^2, \ \forall\ (k,j) \in N_a, \\
& \delta \ge 0, \hat{\delta} \ge 0.
\end{aligned} \tag{5}$$

Question 4

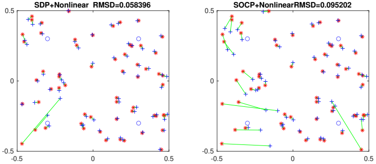


Figure: radius = 0.3, ns = 0.1

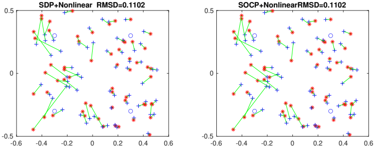


Figure: radius = 0.3, ns = 0.3

Observation. SOCP +Nonlinear is close to SDP + Nonlinear.

Question 5

The current available SDP solvers are still too time consuming for solving large-scale SDP problems. In this part, we implement first-order SDP methods to solve the SDP relaxation problem for SNL.

$$\min_{Z} \frac{1}{2}\left(\|Z_{1:d,1:d} - I_d\|^2 + \sum_{(i,j)\in Nx}((0; e_i - e_j)(0; e_i - e_j)^T \cdot Z - d_{ij}^2)^2 + \sum_{(k,j)\in Na}((a_k; -e_j)(a_k; -e_j)^T \cdot Z - \hat{d}_{ij}^2)^2\right) \tag{6}$$

$$\text{s.t. } Z \succeq 0,$$

which can be casted as

$$\begin{aligned} &\min f(Z) = \frac{1}{2}\|\mathcal{A}Z - b\|^2 \\ &\text{s.t. } Z \succeq 0, \end{aligned} \tag{7}$$

where $Z \in \mathbb{R}^{(d+n)\times(d+n)}$.

Question 5: Gradient Projection

GP for (6):
Step 1: $\hat{Z}^{k+1} = Z^k - \frac{1}{\beta}\nabla f(Z^k)$;
Step 2: $Z^{k+1} = \mathrm{Proj}_K(\hat{Z}^{k+1}) = V \max\{0, \Lambda\} V^T$.

Drawback: Eigendecomposition may be costly!

Question 5:

- Try just computing the few largest eigenpairs, say six largest λ_i with corresponding eigenvectors v_i and let: $Z^{k+1} = \sum_{i=1}^{6} \max\{0, \lambda_i v_i v_i^T.\}$ Typically, a few extreme eigenvalues of a symmetric matrix can be computed more efficiently. Here, we assume that the problem has only one anchor at the origin. One can find the true position later using two more anchor information.
- Any possible theoretical analysis of the projection algorithm?

Question 5

Note that

$$Z^{k+1} = \sum_{i=1}^{d} \max\{0, \lambda_i v_i v_i^T.\} \Longleftrightarrow Z^{k+1} \in \Pi_\Omega(\hat{Z}^{k+1}),\ \Omega = \{Z : Z \succeq 0, \mathrm{rank}(Z) \leq d\}.$$

Consider the following 2D-SNL model:

$$\begin{aligned} \min\ & f(Z) = \frac{1}{2}\|\mathcal{A}Z - b\|^2 \\ \text{s.t.}\ & Z \succeq 0,\ \mathrm{rank}(Z) \leq 2, \end{aligned} \tag{8}$$

$\Rightarrow$ Convex function + nonconvex constraint

Question 5: analysis for GP

- Consider the generic model:

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } x \in \mathcal{X}, \end{aligned} \tag{9}$$

- $\mathcal{X}$ is closed non-convex set; f is differentiable and L-Lipschitz smooth, i.e.,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2. \tag{10}$$

- GP method for (9):

$$x^{k+1} \in \Pi_{\mathcal{X}}(x^k - \frac{1}{\beta}\nabla f(x^k)),$$

Question 5: analysis for GP

Theorem

Let $\{x^k\}_k$ be the sequence generated by GP, ε is any given positive number. Suppose f is L-Lipschitz smooth, if $\beta > L$, then at most $N = \lfloor \frac{2(f(x^0) - f(x^*))}{(\beta - L)\varepsilon^2} \rfloor + 1$ steps, $\|x^N - x^{N-1}\| < \varepsilon$.

- Projection over $\mathcal{X}$ (non-convex):

$$\begin{aligned} &\min \ f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{\beta}{2}\|x - x^k\|^2 \\ &\text{s.t. } \ x \in \mathcal{X}. \end{aligned} \tag{11}$$

- x^k is feasible:

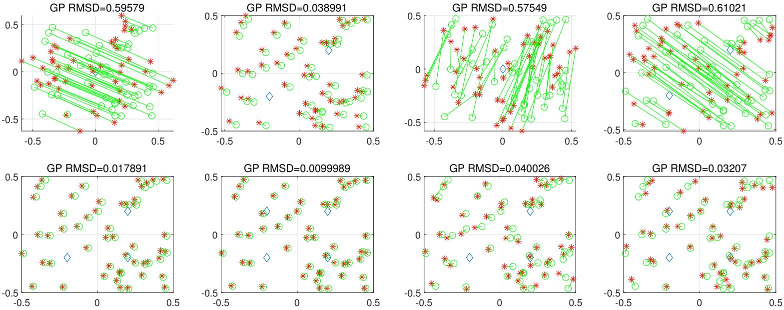
$$f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{\beta}{2}\|x^{k+1} - x^k\|^2 \leq f(x^k)$$

- L-Lipschitz smoothness

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2$$

$$\Rightarrow f(x^{k+1}) \leq f(x^k) - \frac{\beta - L}{2}\|x^{k+1} - x^k\|^2.$$

Question 5: numerical results



(a) Noise level: 0 (b) Noise level: 0.1

Figure: GP for SNL: $n = 50$. Radius $= 1.0$.

	n=20				n=50			
	m=1	m=2	m=3	m=4	m=1	m=2	m=3	m=4
CPU-time(s)	0.44	0.45	0.42	0.42	1.11	1.23	1.14	1.51
RMSD	6.16e-01	2.13e-02	1.21e-02	2.40e-03	5.96e-02	3.90e-02	1.79e-02	1.00e-02

Table: GP for SNL: noise level=0, Radius=0.8 for $n = 20$ and Radius=1.0 for $n = 50$.

Question 6: ADMM

Question 6: Another speed-up may be using ADMM approach. One can reformulate the nonlinear least squares model as

$$\begin{aligned} \min_{X,Y} \; & \sum_{(i,j)\in N_x} [(x_i - x_j)^T (y_i - y_j) - d_{ij}^2]^2 + \sum_{(k,j)\in N_a} [(a_k - x_j)^T (a_k - y_j) - \hat{d}_{kj}^2]^2 \\ \text{s.t. } & X = Y, \end{aligned} \tag{12}$$

where $X = (x_1; x_2; \ldots; x_n) \in \mathbb{R}^{dn}$ and $Y = (y_1; y_2; \ldots; y_n) \in \mathbb{R}^{dn}$.

- Augmented Lagrangian function:

$$\mathcal{L}_\beta(X, Y; Z) = \sum_{(i,j)\in N_x} [(x_i - x_j)^T (y_i - y_j) - d_{ij}^2]^2 + \sum_{(k,j)\in N_a} [(a_k - x_j)^T (a_k - y_j) - \hat{d}_{kj}^2]^2 - \langle Z, X - Y \rangle + \frac{\beta}{2} \|X - Y\|^2$$

Question 6: ADMM

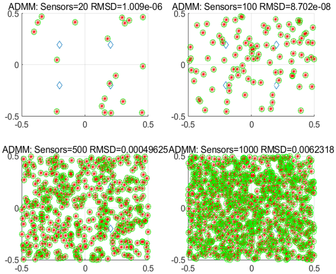
The ADMM scheme for (12):
Step 1: $X^{k+1} = \arg\min_X \ \mathcal{L}_\beta(X, Y^k; Z^k)$;
Step 2: $Y^{k+1} = \arg\min_Y \ \mathcal{L}_\beta(X^{k+1}, Y; Z^k)$;
Step 3: $Z^{k+1} = Z^k - \beta(X - Y)$.

- Solve linear system in Step 1 and Step 2. For Step 1:

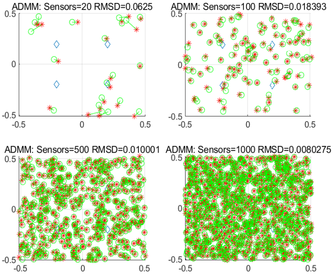
$$\begin{aligned} 0 &= \nabla_X \mathcal{L}_\beta(X^{k+1}, Y^k; Z^k) \\ &= \Big(\sum_{(i,j)\in N_x} G_{ij}^k + \sum_{(i,j)\in N_a} \hat{G}_{ij}^k + \beta I\Big)X + \Big(\sum_{(i,j)\in N_x} h_{ij}^k + \sum_{(i,j)\in N_a} \hat{h}_{ij}^k - Z - \beta Y\Big) \\ &=: G^k X + h^k \end{aligned}$$

- G^k tends to be sparse (sparse Cholesky factorization)
- when number of sensors is extremely large (apply CG method)

Question 6: numerical result



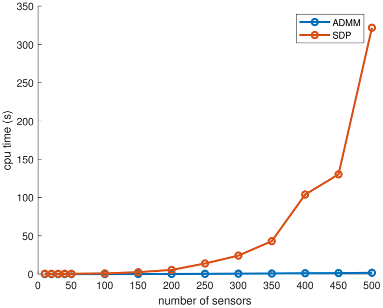
(a) Noise level=0, Radius=0.6



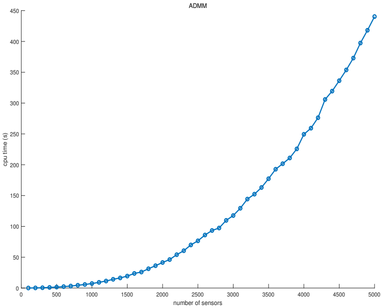
(b) Noise level=0.1, Radius=0.6

Figure: ADMM for SNL nonlinear model.

Question 6: numerical result



(a) Noise level=0, Radius=0.6



(b) Noise level=0, Radius=0.6

Figure: Compare ADMM and SDP (IPM) time: RMSD$< 1e-2$.

Conclusions

- Convex (SOCP, SDP) relaxation for SNL amy solve the original problem exactly under some conditions (e.g. SDP for universally-localizable networks);
- First-order methods (e.g. ADMM) is quite efficient to solve large-scale SNL problem;
- Lack of strong theoretical convergence guarantees for non-convex GP and non-ADMM for SNL.

Thanks for your listening.