An Efficient Steepest Descent Projection Method for Sensor Network Tracking

Yinyu Ye
Joint Work with Andrew Naber

Stanford University

18 May 2021
Mini-symposium on Sensor Network Localization and Dynamical Distance Geometry, Fields Institute, Canada

Outline

Sensor network localization

Steepest descent projection method

SNL algorithm and analysis

Numerical results

Sensor network localization (SNL) problem

Suppose that there are n different sensors in our network and that the noisy Euclidean distance d_{ij} between sensors i and j is known if $(i, j) \in N$. We want find $x_1, \ldots, x_n \in \mathbf{R}^r$ that solve the problem:

Anchor-Free

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in N} l(\|x_i - x_j\|_2^2 - d_{ij}^2) \\ \text{subject to} & x_1 + \cdots + x_n = 0 \end{array} \tag{1}$$

Anchored

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in N} l(\|x_i - x_j\|_2^2 - d_{ij}^2) \\ \text{subject to} & x_1 = a_1, \cdots, x_r = a_r \end{array} \tag{2}$$

for some loss function $l(\cdot)$ (assumed smooth).

The problem is generally non-convex even though $l(\cdot)$ is convex.

SNL semidefinite programming (SDP) representation

Lift the quadratic expression $\|x_i - x_j\|_2^2$ to the equivalent linear expression $U_{ii} - U_{ij} - U_{ji} + U_{jj}$ or $(e_i - e_j)(e_i - e_j)^T \cdot U$, where U is the positive semidefinite matrix with entries $U_{ij} = x_i^T x_j$.

In other words, $U = XX^T$ for $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T$.

Problem (1) is then equivalent to

$$\begin{array}{ll} \text{minimize} & \sum_{(i,j)\in N} l(U_{ii} - U_{ij} - U_{ji} + U_{jj} - d_{ij}^2) \\ \text{subject to} & U \succeq 0 \\ & U\mathbf{1} = 0 \\ & \textbf{rank}(U) \leq r. \end{array} \tag{3}$$

Problem (2) can be represented in a similar fashion.

References

SDP based Sensor Network Localization:

Pratik Biswas, Y. Y, "Semidefinite programming for ad hoc wireless sensor network localization," *Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004.

Pratik Biswas, T-C Liang, K-C Toh, Y. Y, T-C Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE transactions on automation science and engineering*, 3(4), 2006

Nathan Krislock and Henry Wolkowicz, "Explicit Sensor Network Localization using Semidefinite Representations and Facial Reductions", *SIAM Optimization*, 20(5), 2010.

Alfakih and Y.Y, "On affine motions and bar frameworks in general position," Linear Algebra and its Applications, 438, 2013
and many more...

Simplify constraints in the representation

We can eliminate the equality constraints in problem (3) with the reparameterization $U = CVC^T$ for

$$C = \begin{bmatrix} -\frac{1}{\sqrt{n}}\mathbf{1}_{n-1}^T \\ I - \frac{1}{n+\sqrt{n}}\mathbf{1}_{n-1}\mathbf{1}_{n-1}^T \end{bmatrix} \in \mathbf{R}^{n\times(n-1)}$$

to obtain a problem of the form

$$\begin{array}{ll} \text{minimize} & g(V) \\ \text{subject to} & V \succeq 0 \\ & \mathbf{rank}(V) \leq r. \end{array} \tag{4}$$

Unfortunately, problem (4) remains non-convex, but we can project any symmetric matrix A onto the feasible set easily, meaning that the following minimization problem can be solved easily

$$\begin{array}{ll} \text{minimize} & \|V - A\|_f^2 \\ \text{subject to} & V \succeq 0 \\ & \mathbf{rank}(V) \leq r. \end{array} \tag{5}$$

Steepest descent projection method (SDPM)

Taking a step back, we want to solve the general problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

where f is differentiable and satisfies the Lipschitz condition:

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} \|y - x\|_2^2$$

for all $x, y \in \mathcal{X}$. Let $x^\star$ denote an optimal solution.
$\mathcal{X}$ is a general set.

Key assumption

We can project any point y onto $\mathcal{X}$ exactly and easily by solving the problem

$$\begin{array}{ll} \text{minimize} & \|x - y\|_2 \\ \text{subject to} & x \in \mathcal{X} \end{array}$$

Projection may not be unique!

Sets that are easy to project onto

- Nonnegative orthant: $x = \max\{0,\ y\}$
- $s(< n)$-sparse vectors in $\mathbf{R}^n$ or $\mathbf{R}^n_+$: keep the largest s absolute-value entries and zero the rest
- SDP cone $\mathbf{S}^n_+$: keep the positive eigenvalue part
- r-rank matrices in $\mathbf{R}^{m\times n}$ or $\mathbf{S}^n_+$
- some lattices (like $\mathbf{Z}^n$)
- unit sphere
- orthogonal matrices

Steepest descent projection method

Start from $x^k \in \mathcal{X}$, then let

$$x^{k+1} = \mathbf{proj}_{\mathcal{X}} \left(x^k - \frac{1}{2L} \nabla f(x^k) \right)$$

Standard steepest descent step: $\hat{x}_{k+1} = x^k - \frac{1}{L} \nabla f(x^k)$ but we take a half of the standard step-size.

$\hat{x}^{k+1}$ may be infeasible so that we project it back to the feasible set.

It is a "Descent-First and Feasible-Second" approach.

References

Steepest descent projection method (convex objective, convex feasible set):

A. A. Goldstein, "Convex programming in Hilbert space," *Bulletin of the American Mathematical Society*, vol. 70, no. 5, pp. 709–710, 1964.

E. S. Levitin and B. T. Polyak, "Constrained minimization methods," *USSR Computational mathematics and mathematical physics*, vol. 6, no. 5, pp. 1–50, 1966.

J. C. Dunn, "Global and asymptotic convergence rate estimates for a class of projected gradient processes," *SIAM Journal on Control and Optimization*, vol. 19, no. 3, pp. 368–400, 1981.

More references

Projected gradient method under assumptions about the nature of the feasible set:

R. F. Barber and W. Ha, “Gradient descent with non-convex constraints: local concavity determines convergence,” *Information and Inference: A Journal of the IMA*, vol. 7, no. 4, pp. 755–806, 2018.

M. V. Balashov, “About the gradient projection algorithm for a strongly convex function and a proximally smooth set,” *J. of Convex Analysis*, vol. 24, no. 2, pp. 493–500, 2017.

M. Balashov, B. Polyak, and A. Tremba, “Gradient projection and conditional gradient methods for constrained nonconvex minimization,” *Numerical Functional Analysis and Optimization*, pp. 1–28, 2020.

All of these papers make some relatively strong assumptions about the nature of the constraint set and objective function...

Convergence analysis

In each iteration, the method solves the convex QP problem

$$\begin{array}{ll} \text{minimize} & \nabla f(x^k)^T(x - x^k) + L\|x - x^k\|_2^2 \\ \text{subject to} & x \in \mathcal{X}. \end{array}$$

Because x^k is feasible,

$$\nabla f(x^k)^T(x^{k+1} - x^k) + L\|x^{k+1} - x^k\|_2^2 \leq 0.$$

The L-Lipschitz assumption and the previous inequality imply

$$\begin{aligned} f(x^{k+1}) - f(x^k) &\leq \nabla f(x^k)^T(x^{k+1} - x^k) + \frac{L}{2}\|x^{k+1} - x^k\|_2^2 \\ &\leq -L\|x^{k+1} - x^k\|_2^2 + \frac{L}{2}\|x^{k+1} - x^k\|_2^2 \\ &= -\frac{L}{2}\|x^{k+1} - x^k\|_2^2. \end{aligned}$$

The sequence is descent unless $d = x^{k+1} - x^k = 0$.

Convergence rate

With this guaranteed decrease at each iteration, it is simple to show that in at most

$$\frac{2(f(x_0) - f(x^\star))}{L\epsilon^2}$$

iterations, the method generates points such that

$$\|x^{k+1} - x^k\|_2 \leq \epsilon.$$

Interpretation of $d = x^{k+1} - x^k$

Consider the case $\mathcal{X}$ being the nonnegative orthant. We have the following cases:

$$
\begin{aligned}
\nabla f(x^k)_j < 0 \quad & \Rightarrow d_j = -\tfrac{1}{2L}\nabla f(x^k)_j \\
\nabla f(x^k)_j \geq 0 \ \&\ x_j > \tfrac{1}{2L}\nabla f(x^k)_j \quad & \Rightarrow d_j = -\tfrac{1}{2L}\nabla f(x^k)_j \\
\nabla f(x^k)_j \geq 0 \ \&\ x_j \leq \tfrac{1}{2L}\nabla f(x^k)_j \quad & \Rightarrow d_j = -(x^k)_j
\end{aligned}
$$

Therefore d represents the complementary slackness residuals at step k.

In general, d is a feasible direction in the convex hull of $\mathcal{X}$. Thus, $d = 0$ implies that x^k is a first-order stationary solution.

Star-convex feasible set and convex objective

$$
\begin{aligned}
f(x^{k+1}) - f(x^k) &\le \nabla f(x^k)^T(x^{k+1}-x^k) + \tfrac{L}{2}\|x^{k+1}-x^k\|^2 \\
&\le \nabla f(x^k)^T(x^{k+1}-x^k) + L\|x^{k+1}-x^k\|^2.
\end{aligned}
$$

Let $\mathcal{X}$ be x^*-star convex, that is, from any feasible solution $x \in \mathcal{X}$, the convex combination of $\alpha x^* + (1-\alpha)x \in \mathcal{X}$ for any $0 \le \alpha \le 1$ where x^* is a minimum of the problem. Then we must have

$$
f(x^{k+1}) - f(x^k) \le \alpha \nabla f(x^k)^T(x^*-x^k) + \alpha^2 L\|x^*-x^k\|^2,\ \forall 0 \le \alpha \le 1.
$$

We like to find α to minimize the right-hand-side. If f is convex then $0 \ge f(x^*) - f(x^k) \ge \nabla f(x^k)^T(x^*-x^k)$,

$$
\alpha^* = \min\left\{1,\ \frac{|\nabla f(x^k)^T(x^*-x^k)|}{2L\|x^*-x^k\|^2}\right\}
$$

$$
f(x^{k+1}) - f(x^k) \le -\frac{|\nabla f(x^k)^T(x^*-x^k)|^2}{4L\|x^*-x^k\|^2} \le -\frac{(f(x^k)-f(x^*))^2}{4L\|x^*-x^k\|^2}.
$$

Convergence rate

Then it is simple to show that in at most

$$\frac{4L\Delta^2}{\epsilon} \log\left(\frac{f(x^0) - f(x^*)}{\epsilon}\right)$$

iterations, the method generates points such that

$$f(x^k) - f(x^*) \leq \epsilon.$$

Here, Δ if the diameter of the level set $\{x \in \mathcal{X} : \ f(x) \leq f(x^0)\}$.

Star-Convex Example I: non-convex cones are all 0-star convex.

Star-Convex Example II: $x^* \in \mathbf{R}^n$ is a $s(< n)$-sparse optimal solution and $\mathcal{X}$ is the set of all d-sparse solutions x, $s \leq d < n$, such that $\mathrm{supp}(x^*) \subset \mathrm{supp}(x)$.

Suppose further that f is strongly convex. Then,

$$f(x^k) - f(x^*) \geq \nabla f(x^k)^T (x^k - x^*) + \lambda \Delta_k^2 \geq \lambda \Delta_k^2$$

for some $\lambda \leq L$, so

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\lambda}{4L}\right) (f(x^k) - f(x^*))$$

which gives a linear rate of convergence.

There is an early result in compressed sensing which proves a linear rate for the method if $\frac{L}{\lambda} < 2$.

Reformulation of SNL problem (4)

For notational convenience, we rewrite it as

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}(CVC^T) - b) \\ \text{subject to} & V \succeq 0 \\ & \mathbf{rank}(V) \leq r, \end{array} \tag{6}$$

where $f : \mathbf{R}^m \to \mathbf{R}$ is defined by $f(z) = \sum_{k=1}^m l(z_k)$, $\mathcal{A} : \mathbf{S}^n \to \mathbf{R}^m$ is defined entrywise by

$$\mathcal{A}(U)_k = U_{i_k i_k} - U_{i_k j_k} - U_{j_k i_k} + U_{j_k j_k},$$

$b \in \mathbf{R}^m$ is defined entrywise by $b_k = d_{i_k j_k}^2$, and we have indexed the m elements of N with k.

The adjoint of the linear operator $\mathcal{A}$ is defined for all $z \in \mathbf{R}^m$ by

$$\mathcal{A}^*(z) = \sum_{k=1}^{m} z_k (E_{i_k i_k} - E_{i_k j_k} - E_{j_k i_k} + E_{j_k j_k}).$$

The gradient of the objective at some V is given by

$$C^T \mathcal{A}^*(\nabla f(\mathcal{A}(CVC^T) - b))C$$

and that, depending on m, multiplying it with a vector can be done efficiently.

Algorithm 1 SDPM applied to solving problem (4)

determine Lipschitz constant L for the gradient of the objective
choose initialization $Y_0 \in \mathbf{R}^{(n-1)\times r}$
for $t = 0, 1, 2, \ldots$ **do**
    find top r eigenvalues $\lambda_1, \ldots, \lambda_r$, eigenvectors $Q \in \mathbf{R}^{(n-1)\times r}$ of the linear operator

$$Y_t Y_t^T - \frac{1}{2L} C^T \mathcal{A}^*(\nabla f(\mathcal{A}(C Y_t Y_t^T C^T) - b)) C$$

    $Y_{t+1} = Q\,\mathbf{diag}((\lambda_1, \ldots, \lambda_r)_+)^{1/2}$
    if $\|Y_{t+1}^T Y_{t+1} - Y_t^T Y_t\|_F^2 \leq \epsilon^2 \|Y_t^T Y_t\|_F^2$ **then**
        find $s = \mathrm{argmin}_{s\geq 0} f(s\mathcal{A}(C Y_t Y_t^T C^T) - b)$
        break with $X_\epsilon = \sqrt{s} C Y_t$
    end if
end for

Convex relaxation of problem (6)

Remark: we add a scaling step in the algorithm to improve practical performance.

Drop the rank constraint would

$$\begin{array}{ll} \text{minimize} & f(\mathcal{A}(CVC^T) - b) \\ \text{subject to} & V \succeq 0 \end{array} \tag{7}$$

Then, this is a convex-objective and convex feasible set case so that the method would converge to the (global) optimal solution. This is guaranteed if the SNL framework is universally rigid (So & Y, "Theory of Semidefinite programming for sensor-network localization")

In practice, we may keep the solution at most rank-d, $n >> d > r$ to save memory space and computation time. Then gradually reduce d to r.

Analysis of the scaling

Suppose that V is an approximate stationary point. The rescaling in step 7 means that $\mathbf{tr}(\nabla g(sV)(sV)) = 0$ because either $s = 0$ or

$$\frac{d}{ds} g(sV) = s\,\mathbf{tr}(\nabla g(sV)V) = 0.$$

Trivially, $sV \succeq 0$ because $V \succeq 0$.

The positive semidefiniteness of $\nabla g(sV)$ cannot be guaranteed in general, but if the number of positive eigenvalues of $sV - \frac{2}{L}\nabla g(sV)$ is less than r, then $\nabla g(sV) \succeq 0$.

To see this, first let $\lambda_i = \lambda_i(sV - \frac{2}{L}\nabla g(sV)$ and q_i denote the corresponding normalized eigenvector for $i = 1, \ldots, n$. Then,

$$\begin{aligned} sV - \frac{2}{L}\nabla g(sV) &= \sum_{\lambda_i > 0} \lambda_i q_i q_i^T + \sum_{\lambda_i \le 0} \lambda_i q_i q_i^T \\ &= sV + \sum_{\lambda_i \le 0} \lambda_i q_i q_i^T, \end{aligned}$$

which implies that $\nabla g(sV) \succeq 0$.

Numerical results

We generate instances of the SNL problem by choosing $n = 100$ points uniformly at random in $[-1,1] \times [-1,1]$. We add small random displacements of approximately 0.1 to each sensor and attempt to recover these new positions from noisy distance measurements using the algorithm.

We use RMS error as the metric:

$$\left(\frac{1}{n} \sum_{i=1}^{n} \|x_i - x_i^{\text{recovered}}\|_2^2 \right)^{1/2}.$$

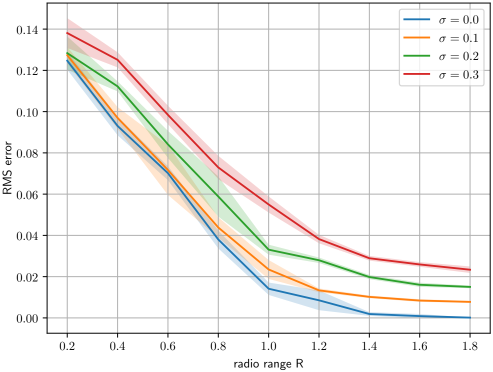

Figure 1: RMS error for varying levels of additive zero-mean Gaussian noise.

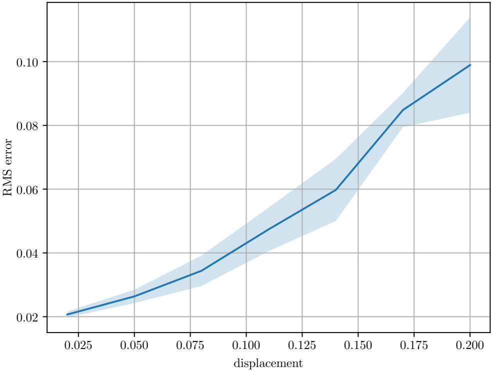

Figure 2: RMS error against different displacements used for initialization ($R = 0.8$, $\sigma = 0.1$).

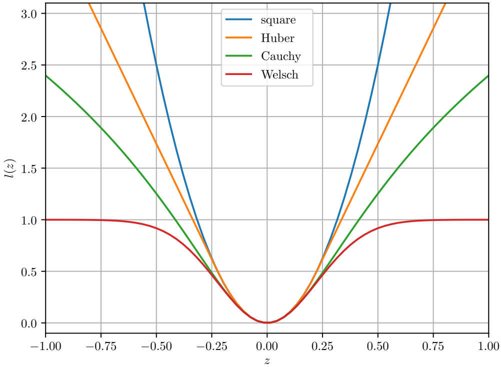

Figure 3: Robust loss functions with unit shape parameter.

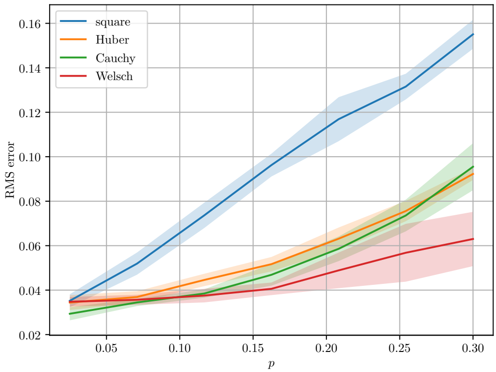

Figure 4: RMS error at $R = 0.8$, $\sigma = 0.1$, and displacement of about 0.1 for the four different robust loss functions against the probability p of a gross measurement corruption

The method is particularly effective as a post-solver for the SDP relaxation without rank constraint.

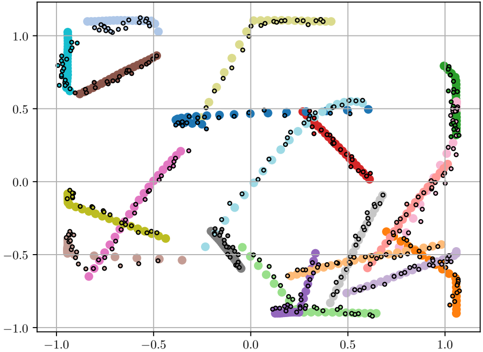

Figure 5: A sensor network tracking problem solved using our algorithm with 20 of 100 trajectories shown. The large circles are true sensor positions and small circles with a black outline are estimated.

Satellite cloud tracking

Video

Remarks and Questions

Questions?