Project 9

João Xavier
Instituto Superior Técnico

1 Introduction

Setup. Consider K targets and N sensors deployed in a given area. Let $t_k \in \mathbf{R}^2$ be the position of target k ($k = 1, \ldots, K$) and $s_n \in \mathbf{R}^2$ be the position of sensor n ($n = 1, \ldots, N$).

Each sensor reports its distance to the closest target plus some noise. More specifically, sensor n ($n = 1, \ldots, N$) reports the measurement

$$d_n = \|s_n - x_n\| + \text{noise}, \tag{1}$$

where $x_n \in \mathbf{R}^2$ is the position of the target closest to sensor n (thus, x_n is an element of set of the targets' positions $\{t_1, \ldots, t_K\}$). Note that, for a vector $v = (v_1, \ldots, v_d) \in \mathbf{R}^d$, the symbol $\|v\|$ denotes its Euclidean norm,

$$\|v\| = \left(v_1^2 + \cdots + v_d^2\right)^{1/2}.$$

We will start with $T = 2$ targets and $N = 64$ sensors. The sensors are arranged in an 8×8 grid, with the grid points separated by one unit both in the horizontal and vertical directions. We say that sensor n is a neighbor of sensor m if they are neighbors in the grid, that is, if $\|s_n - s_m\| = 1$. We use the notation $n \sim m$ to signal that sensors n and m are neighbors. Note that most sensors have four neighbors, but some have only two or three neighbors.

Goal. Given the sensors' positions s_n ($n = 1, \ldots, N$) and their respective measurements d_n ($n = 1, \ldots, N$), we aim to find the targets' positions t_k ($k = 1, \ldots, K$).

2 Step 1

As explained in the meeting, we will start by solving the convex optimization problem

$$\begin{array}{ll} \underset{x_1,\ldots,x_N,y_1,\ldots,y_N}{\text{minimize}} & \sum_{n=1}^N \left(\alpha_n^T \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \beta_n\right)^2 + \rho \sum_{1 \le n \le N} \sum_{n+1 \le m \le N, m \sim n} \left\| \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \end{bmatrix} \right\| \\ \text{subject to} & \|x_n\|^2 \le y_n, \quad n = 1, \ldots, N, \end{array} \tag{2}$$

where

$$\alpha_n := \begin{bmatrix} -2s_n \\ 1 \end{bmatrix} \in \mathbf{R}^3, \quad (n = 1, \ldots, N),$$

$\beta_n := d_n^2 - \|s_n\|^2$ $(n = 1, \ldots, N)$, and $\rho > 0$ is a given positive constant—in this project, use $\rho = 10$. The optimization variables are $x_n \in \mathbf{R}^2$ and $y_n \in \mathbf{R}$ for $n = 1, \ldots, N$. Note that, in the second sum of the cost function in (2), we sum only over neighbor sensors.

Implementation. In the MATLAB file you received, the positions of the sensors are the columns of the matrix `sensors` $\in \mathbf{R}^{2\times N}$; thus, the position of the nth sensor (denoted a_n above) is the nth column of the matrix `sensors`. The measurements are given in the vector `d` $\in \mathbf{R}^{1\times N}$; thus, the measurement of sensor n (denoted d_n above) is the nth entry of the vector `d`.

After you solve problem (2), you should check if your solution matches the one given in the matrix `X` $\in \mathbf{R}^{2\times N}$ and vector `y` $\in \mathbf{R}^{1\times N}$: your optimal x_n should be the nth column of matrix `X`, and your optimal y_n should be the nth entry of vector `y`.

3 Step 2

The solution to problem (2), denoted $(x_n^\star, y_n^\star)$ for $n = 1, \ldots, N$, typically consists of different target estimates for different sensors. That is, we often have $x_n^\star \neq x_m^\star$ for $n \sim m$, meaning that we have a surplus of guesses for the target positions (ideally, we should have only $T = 2$ distinct $(x_n^\star, y_n^\star)$s).

To produce fewer guesses, instead of solving problem (2) we could solve

$$\begin{array}{ll} \underset{x_1,\ldots,x_N,y_1,\ldots,y_N}{\text{minimize}} & \sum_{n=1}^N \left(\alpha_n^T \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \beta_n \right)^2 + \rho \sum_{1\le n\le N} \sum_{n+1\le m\le N, m\sim n} \log \left\| \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \end{bmatrix} \right\| \\ \text{subject to} & \|x_n\|^2 \le y_n, \quad n = 1, \ldots, N. \end{array} \tag{3}$$

Formulation (3) exploits the fact that the log function descends aggressively to $-\infty$ near zero. Thus, since there is a huge payoff in bringing the arguments of the log's to zero, we expect formulation (3) to produce solutions with $x_n = x_m$ for most pairs $n \sim m$. In other words, we expect formulation (3) to generate fewer target estimates by merging the target estimates of most neighbors.

Unfortunately, the optimization problem (3) is nonconvex (why?). However, using the Majorization-Minorization (MM) framework, we can commonly find a local minimum. As discussed in the meeting, application of the MM framework to problem (3) leads to the iterative scheme

$$\left(x_n^{(r+1)}, y_n^{(r+1)}\right)_{n=1}^N = \begin{array}{ll} \underset{x_1,\ldots,x_N,y_1,\ldots,y_N}{\text{argmin}} & f^{(r)}(x_1, y_1, \ldots, x_N, y_N) \\ \text{subject to} & \|x_n\|^2 \le y_n, \quad n = 1, \ldots, N, \end{array} \tag{4}$$

where

$$f^{(r)}\left(x_1, y_1, \ldots, x_N, y_N\right) := \sum_{n=1}^{N}\left(\alpha_n^T\begin{bmatrix}x_n\\ y_n\end{bmatrix} - \beta_n\right)^2 + \rho \sum_{1\leq n\leq N}\ \sum_{n+1\leq m\leq N, m\sim n} \omega_{n,m}^{(r)}\left\|\begin{bmatrix}x_n\\ y_n\end{bmatrix} - \begin{bmatrix}x_m\\ y_m\end{bmatrix}\right\|$$

and

$$\omega_{n,m}^{(r)} := \left(\left\|\begin{bmatrix}x_n^{(r)}\\ y_n^{(r)}\end{bmatrix} - \begin{bmatrix}x_m^{(r)}\\ y_m^{(r)}\end{bmatrix}\right\| + \epsilon\right)^{-1},$$

with ϵ being a tiny positive constant. Thus, we solve a sequence of problems of the form (4) for $r = 0, 1, 2, \ldots$. Note that each problem (4) is convex (why?). We initialize the iterative scheme with the solution of problem (2):

$$\left(x_n^{(0)}, y_n^{(0)}\right) := \left(x_n^\star, y_n^\star\right), \qquad \text{for } n = 1, \ldots, N.$$

4 Step 3

Let $\mathcal{T} := \left\{\left(\overline{x}_p, \overline{y}_p\right) : p = 1, \ldots, P\right\}$ be the set of distinct target estimates produced by the MM-based iterative scheme. Typically, after $r = 30$ iterations of (4), we have $4 \leq P \leq 8$ target estimates, which is still too high because our setup features only $T = 2$ targets.

We now improve each pair of target estimates $\left(\left(\overline{x}_p, \overline{y}_p\right), \left(\overline{x}_q, \overline{y}_q\right)\right)$ to a polished pair of target estimates $\left(\left(\widehat{x}_p, \widehat{y}_p\right), \left(\widehat{x}_q, \widehat{y}_q\right)\right)$.

Polishing the target estimates. Take a pair $\left(\left(\overline{x}_p, \overline{y}_p\right), \left(\overline{x}_q, \overline{y}_q\right)\right)$ from the set $\mathcal{T}$ and define Ω_p to be the subset of indexes of the sensors closer to target estimate p than to target estimate q:

$$\Omega_p := \left\{n = 1, \ldots, N : \|a_n - \overline{x}_p\| < \|a_n - \overline{x}_q\|\right\}. \tag{5}$$

Since all the sensors in Ω_p have the same nearest target, we can improve the preliminary target estimate $\left(\overline{x}_p, \overline{y}_p\right)$ to the polished version

$$\begin{array}{rll}\left(\widehat{x}_p, \widehat{y}_p\right) := & \underset{x,y}{\operatorname{argmin}} & \sum_{n=1, n\in\Omega_p}^{N}\left(\alpha_n^T\begin{bmatrix}x\\ y\end{bmatrix} - \beta_n\right)^2 \\ & \text{subject to} & \|x\|^2 \leq y.\end{array} \tag{6}$$

Note that the formulation (6) involves just the sensors in Ω_p and forces them to agree on a common target estimate.

Similarly, the preliminary target estimate $\left(\overline{x}_q, \overline{y}_q\right)$ is improved to the polished version

$$\begin{array}{rll}\left(\widehat{x}_q, \widehat{y}_q\right) := & \underset{x,y}{\operatorname{argmin}} & \sum_{n=1, n\in\Omega_q}^{N}\left(\alpha_n^T\begin{bmatrix}x\\ y\end{bmatrix} - \beta_n\right)^2 \\ & \text{subject to} & \|x\|^2 \leq y.\end{array} \tag{7}$$

where

$$\Omega_q := \left\{n = 1, \ldots, N : \|a_n - \overline{x}_p\| \geq \|a_n - \overline{x}_q\|\right\}. \tag{8}$$

Generating the final estimates. After polishing each pair of target estimates from $\mathcal{T}$, we end up with several pairs of polished target estimates $((\widehat{x}_p, \widehat{y}_p), (\widehat{x}_q, \widehat{y}_q))$, where $1 \leq p < q \leq P$; more precisely, we end up with $P(P-1)/2$ of such pairs. Which pair (p, q) should we pick to be our final estimate for the target positions?

Answer: The pair that best fits the sensor measurements in (1). To find the best pair, we first evaluate the fitting error for each (p, q) as

$$e_{(p,q)} := \sum_{n=1, n\in\Omega_p}^{N} \left(\|a_n - \widehat{x}_p\| - d_n\right)^2 + \sum_{n=1, n\in\Omega_q}^{N} \left(\|a_n - \widehat{x}_q\| - d_n\right)^2 , \tag{9}$$

where, for a given (p, q), the subsets Ω_p and Ω_q are given by (5) and (8). Finally, we pick the pair (p, q) with the smallest fitting error $e_{(p,q)}$.