Fermat-Toricelli Problem: SDP formulation

Hon Leung Lee

May 21, 2013

1 Problem formulation

We describe here the idea of SDP formulation in [1]. Our problem is to minimize over $x \in \mathbb{R}^2$, $F(x) = \sum_{j=1}^{n} \eta_j \|x - a_j\|$ where all η_j is positive and $a_j \in \mathbb{R}^2$. Let x and a_j have coordinates (x, y) and (a_j, b_j) respectively (some abuse of notation here). Then the objective function becomes

$$F(x, y) = \sum_{j=1}^{n} \eta_j \sqrt{(x - a_j)^2 + (y - b_j)^2}.$$

Let the minimization problem be $(\mathcal{P})$.

Introduce slack variables $d_j = \sqrt{(x - a_j)^2 + (y - b_j)^2}$ for all j. Then $d_j^2 = (x - a_j)^2 + (y - b_j)^2$. One can use this to show that:

Theorem 1. *The problem $(\mathcal{P})$ is equivalent to the semidefinite program (SDP) $(\mathcal{P}_2)$:*

$$\text{minimize } F(x) = \sum_{j=1}^{n} \eta_j d_j \text{ subject to } \begin{bmatrix} d_j + x - a_j & y - b_j \\ y - b_j & d_j - x + a_j \end{bmatrix} \succeq 0$$

for all $j = 1, \cdots, n$

(Here $A \succeq 0$ means the matrix A is positive semidefinite; Notice the decision variables of $(\mathcal{P}_2)$ are $d_1, \cdots, d_n, x, y$).

Proof. For each j let $A_j = \begin{bmatrix} d_j + x - a_j & y - b_j \\ y - b_j & d_j - x + a_j \end{bmatrix}$. If $d_j = \sqrt{(x - a_j)^2 + (y - b_j)^2}$ then all principal minors of A_j are nonnegative:

$d_j \geq |x - a_j|$ and $\det(A) = d_j^2 - (x - a_j)^2 - (y - b_j)^2 = 0$. Thus $A_j \succeq 0$. So feasibility of $(\mathcal{P})$ implies the feasibility of $(\mathcal{P}_2)$, hence the minimum value in $(\mathcal{P}_2)$ cannot exceed that of $(\mathcal{P})$. On the other hand if $A_j \succeq 0$, then all the eigenvalues of A_j are nonnegative. By assessing the eigenvalues using quadratic formula one sees $d_j \geq \sqrt{(x - a_j)^2 + (y - b_j)^2} \geq$ the minimum value of $(\mathcal{P})$. This completes the other direction that the minimum value of $(\mathcal{P})$ is at most that of $(\mathcal{P}_2)$, and so the conclusion follows. $\square$

The above SDP $(\mathcal{P}_2)$ can be implemented using the SDP solver in `cvxopt` package in Python.

2 Discussion

We already implemented the following baby example using the `cvxopt` package.

Example 2. *If* $(a_0, b_0) = (0, 0)$, $(a_1, b_1) = (1, 0)$ *and* $(a_2, b_2) = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$ *then the Fermat point has to be the centroid* $\left(\frac{1}{2}, \frac{1}{2\sqrt{3}}\right)$, *whence the minimum distance is* $\sqrt{3}$.

The solver provided more information which we are also interested. For the above example it takes 6 iterations. The CPU time is 6.73 ms per loop. Also the convergence looks superlinear. This is nice.

3 Next step

1. Implement the SDP for general n points in $\mathbb{R}^2$ and general weights.
2. Perform numerical experiments through simple sampling. Fix $n = 5$ (or any nice integer). Generate 5 points in $\mathbb{R}^2$ using Gaussian distribution, and generate 500 such data. Run the algorithm and analyze the results.

References

[1] Nie, J., Parrilo, P. A..Sturmfels, B. : *Semidefinite Representation of the k-Ellipse. In Algorithms in Algebraic Geometry.* Springer New York (2008), 117-132.