Mathematical Optimization

IMA Summer Course: Conic/Semidefinite Optimization

Shuzhong Zhang

Department of Industrial & Systems Engineering
University of Minnesota

zhangs@umn.edu

August 3, 2016

Let us start by considering the rationalized *preferences* choices. In the traditional economics theory, the rationality of the preferences lies in the fact that they pass the following consistency test:

1. Any commodity is always considered to be preferred over itself.
2. If commodity A is preferred over commodity B, while B is preferred over A at the same time, then the decision-maker is clearly *indifferent* between A and B, i.e., they are identical to the decision-maker.
3. If commodity A is preferred over commodity B, then any positive multiple of A should also be preferred over the same amount of multiple of B.
4. If commodity A is preferred over commodity B, while commodity B is preferred over commodity C, then the decision-maker would prefer A over C.

In mathematical terms, the analog is an object known as the pointed convex cone. We shall confine ourself to a finite dimensional Euclidean space here, to be denoted by $\mathbf{R}^n$. A subset $\mathcal{K}$ of $\mathbf{R}^n$ is called a *pointed convex cone* in $\mathbf{R}^n$ if the following conditions are satisfied:

1. The origin of the space – vector 0 – belongs to $\mathcal{K}$.
2. If $x \in \mathcal{K}$ and $-x \in \mathcal{K}$ then $x = 0$.
3. If $x \in \mathcal{K}$ then $tx \in \mathcal{K}$ for all $t > 0$.
4. If $x \in \mathcal{K}$ and $y \in \mathcal{K}$ then $x + y \in \mathcal{K}$.

If the objects in question reside in $\mathbf{R}^n$, then a preference ordering is rational if and only if there is a convex cone $\mathcal{K}$, such that x is preferred over y is signified by $x - y \in \mathcal{K}$. In this context, the order-defining convex cone may not be a *closed* set. Consider the lexicographic ordering in $\mathbf{R}^2$. That is, for any two points in $\mathbf{R}^2$, the preferred choice is the one with greater first coordinate; in case a tie occurs then the point with larger second coordinate is preferred; in case the second coordinate is also a tie, then the two points are identical. Now, the underlying order-defining cone can be explicitly written as

$$\mathcal{K} = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \, x_1 > 0 \right\} \bigcup \left(\mathbf{R}_+ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right).$$

In general, the preference ordering may not necessarily be complete: there can be incomparable objects. For convenience, from now on we shall consider *closed pointed convex cones*, i.e., $\mathrm{cl}\,\mathcal{K} = \mathcal{K}$ and $\mathcal{K} \cap (-\mathcal{K}) = 0$. The ordering defined by a closed pointed convex cone is necessarily partial.

Cone $\mathcal{K}$ is called *proper* if

- $\mathcal{K}$ is convex;
- $\mathcal{K}$ is solid;
- $\mathcal{K}$ is pointed.

Many decision problems can be formulated using a chosen proper cone as a preference ordering.

A most famous example is linear programming:

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{subject to} & Ax = b \\ & x \geqslant 0, \end{array}$$

where A is a matrix, and b and c are vectors. The last constraint $x \geqslant 0$ is understood to be a componentwise relation, which can as well be written as $x \in \mathbf{R}^n_+$.

Conic Optimization

In this light, linear programming is a special case of the following *conic optimization* model

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array}$$

where $\mathcal{K} \subseteq \mathbf{R}^n$ is a prescribed closed convex cone. We can always assume that the rows in A are all linearly independent.

In practice, the following three convex cones are most popular in conic optimization:

- $\mathcal{K} = \mathbf{R}^n_+$.
- $\mathcal{K}$ is a Cartesian product of Lorentz cones; that is,

$$\begin{aligned}\mathcal{K} \quad = \quad & \left\{ \left(\begin{array}{c} t_1 \\ x_1 \end{array} \right) \middle| \, t_1 \in \mathbf{R}, \, x_1 \in \mathbf{R}^{d_1}, \, t_1 \geqslant \|x_1\| \right\} \\ & \times \cdots \times \left\{ \left(\begin{array}{c} t_m \\ x_m \end{array} \right) \middle| \, t_m \in \mathbf{R}, \, x_m \in \mathbf{R}^{d_m}, \, t_m \geqslant \|x_m\| \right\}.\end{aligned}$$

For convenience, let us denote the standard Lorentz cone as follows:

$$\mathrm{SOC}(n+1) = \left\{ \left(\begin{array}{c} t \\ x \end{array} \right) \,\middle|\, t \in \mathbf{R},\ x \in \mathbf{R}^n,\ t \geqslant \|x\| \right\}.$$

In this notation, the previous cone is

$$\mathcal{K} = \mathrm{SOC}(d_1+1) \times \cdots \times \mathrm{SOC}(d_m+1)$$

Semidefinite Programming

- $\mathcal{K}$ is the cone of positive semidefinite matrices either in $\mathcal{S}^{n\times n}$ (n by n real symmetric matrices) or in $\mathcal{H}^{n\times n}$ (n by n complex Hermitian matrices); that is, $\mathcal{K} = \mathcal{S}_+^{n\times n}$ or $\mathcal{K} = \mathcal{H}_+^{n\times n}$.

Specifically, the standard conic optimization model in this case is:

$$\begin{array}{ll} \text{minimize} & C \bullet X \\ \text{subject to} & A_i \bullet X = b_i,\ i = 1, ..., m \\ & X \succeq 0 \end{array}$$

where

$$X \bullet Y \equiv \langle X, Y \rangle \equiv \sum_{i,j} X_{ij} Y_{ij} \equiv \text{Tr}\, XY.$$

The first choice of the cone is also known as *polyhedral cone* and the corresponding optimization problem is *Linear Programming* (LP);

The second choice of the cone corresponds to *Second Order Cone Programming* (SOCP);

The third choice of the cone corresponds to *Semidefinite Programming* (SDP).

As an appetizer we shall introduce some examples leading to SOCP and SDP.

Example 1: The Weber problem

The first traceable problem of SOCP is perhaps the following problem posed by Pierre de Fermat in the 17th century. Given three points a, b and c on the plane, find the point in the plane that minimizes the total distance to the three given points. The solution was found by Torricelli, hence known as the Torricelli point, and the method was published by Viviani, a pupil of Torricelli, in 1659. The problem can be formulated as SOCP:

$$\begin{array}{ll} \text{minimize} & t_1 + t_2 + t_3 \\ \text{subject to} & u = x - a,\ v = x - b,\ w = x - c \\ & \left(\begin{array}{c} t_1 \\ u \end{array}\right) \in \mathrm{SOC}(3),\ \left(\begin{array}{c} t_2 \\ v \end{array}\right) \in \mathrm{SOC}(3),\ \left(\begin{array}{c} t_3 \\ w \end{array}\right) \in \mathrm{SOC}(3). \end{array}$$

Problems of such type later had gained a renewed interest in management science. In 1909 the German economist Alfred Weber introduced the problem of finding a best location for the warehouse of a company, in such a way that the total transportation cost to serve the customers is minimum. This is known as the Weber problem. Again, the problem can be formulated as SOCP. Suppose that there are m customers needing to be served. Let the location of customer i be a_i, $i = 1, ..., m$. Suppose that customers may have different demands, to be translated as weight w_i for customer i, $i = 1, ..., m$. Denote the desired location of the warehouse to be x. Then, the optimization problem is

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{m} w_i t_i \\ \\ \text{subject to} & \left(\begin{array}{c} t_i \\ x - a_i \end{array} \right) \in \mathrm{SOC}(3),\ i = 1, ..., m. \end{array}$$

Example 2: Convex Quadratic Programming

The popularity of SOCP is also due to that it is a generalized form of convex QCQP (Quadratically Constrained Quadratic Programming). To be precise, consider the following QCQP:

$$\begin{array}{ll} \text{minimize} & x^\top Q_0 x + 2b_0^\top x \\ \text{subject to} & x^\top Q_i x + 2b_i^\top x + c_i \leqslant 0,\ i = 1, ..., m, \end{array} \tag{1}$$

where $Q_i \succeq 0,\ i = 0, 1, ..., m$.

Observe that $t \geqslant x^\top x \Longleftrightarrow \left\| \begin{pmatrix} \frac{t-1}{2} \\ x \end{pmatrix} \right\| \leqslant \frac{t+1}{2}$. Therefore, (1) can be equivalently written as

$$
\begin{array}{ll}
\text{minimize} & x_0 \\
\text{subject to} & \begin{pmatrix} \frac{-2b_0^\top x + x_0 + 1}{2} \\ \frac{-2b_0^\top x + x_0 - 1}{2} \\ Q_0^{\frac{1}{2}} x \end{pmatrix} \in \mathrm{SOC}(n+2) \\
& \begin{pmatrix} \frac{-2b_i^\top x - c_i + 1}{2} \\ \frac{-2b_i^\top x - c_i - 1}{2} \\ Q_i^{\frac{1}{2}} x \end{pmatrix} \in \mathrm{SOC}(n+2),\ i = 1, ..., m.
\end{array}
$$

The versatility of the SOCP modelling goes beyond quadratic models. For the illustrating purpose, let us consider an example of stochastic queue location problem.

Suppose that there are m potential customers to serve in the region. Customers' demands are random, and once a customer calls for service, then the server in the service center will need to go to the customer to provide the required service. In case the server is occupied, then the customer will have to wait. The goal is to find a good location for the service center so as to minimize the expected waiting time of the service.

Suppose that the service calls from the customers are i.i.d., and the demand process follows the Poisson distribution with overall arrival rate λ, and the probability that the lth service call is from customer i is assumed to be p_i, where $l = 1, 2, ...$ and $i = 1, ..., m$. The queueing principle is *First Come First Serve*, and there is only one server in the service center. This model can be regarded as M/G/1 queue, and the expected service time, including waiting and travelling, can be explicitly computed. To this end, denote the velocity of the server to be v, and the location of customer i to be a_i, $i = 1, ..., m$, and the location of the service center to be x.

The expected waiting time for customer i is given by

$$w_i(x) := \frac{(2\lambda/v^2)\sum_{i=1}^{m} p_i\|x-a_i\|^2}{1-(2\lambda/v)\sum_{i=1}^{m} p_i\|x-a_i\|} + \frac{1}{v}\|x-a_i\|, \tag{2}$$

where the first term is the expected waiting time for the server to be free, and the second term is the waiting time for the server to show up at the door, after his departure at the service center.

A first glance at (2) may not suggest that it can be modelled by SOCP, due to the fractional term. One can find the connection, however, by observing the fact that $\|x\|^2/s \leqslant t,\ s \geqslant 0$ is equivalent to $\left\| \begin{pmatrix} \frac{t-s}{2} \\ x \end{pmatrix} \right\| \leqslant \frac{t+s}{2}$. In view of (2), to minimize the total waiting time the optimal location of the service center is formulated as

$$\text{minimize}_x \quad \frac{(2m\lambda/v^2)\sum_{i=1}^m p_i\|x-a_i\|^2}{1-(2\lambda/v)\sum_{i=1}^m p_i\|x-a_i\|} + (1/v)\sum_{i=1}^m \|x-a_i\|$$

or, equivalently,

$$
\begin{array}{ll}
\text{minimize} & (2m\lambda/v^2)\sum_{i=1}^{m} p_i t_i + (1/v)\sum_{i=1}^{m} t_{0i} \\
\text{subject to} & \begin{pmatrix} t_{0i} \\ x - a_i \end{pmatrix} \in \mathrm{SOC}(3),\ i = 1, ..., m \\
& \begin{pmatrix} \frac{t_i + s}{2} \\ \frac{t_i - s}{2} \\ x - a_i \end{pmatrix} \in \mathrm{SOC}(4),\ i = 1, ..., m \\
& s \leqslant 1 - (2\lambda/v)\sum_{i=1}^{m} p_i s_i \\
& \begin{pmatrix} s_i \\ x - a_i \end{pmatrix} \in \mathrm{SOC}(3),\ i = 1, ..., m.
\end{array}
$$

If the objective is to minimize the worst response time, i.e. minimizing $\max_{1\leqslant i\leqslant m} w_i(x)$, then the problem can be formulated as

$$
\begin{array}{ll}
\text{minimize} & t_0 \\
\text{subject to} & t_0 \geqslant (2\lambda/v^2)\sum_{i=1}^m p_i t_i + t_{0i}/v,\ i = 1, ..., m \\
& \left(\begin{array}{c} t_{0i} \\ x - a_i \end{array} \right) \in \mathrm{SOC}(3),\ i = 1, ..., m \\
& \left(\begin{array}{c} \frac{t_i + s}{2} \\ \frac{t_i - s}{2} \\ x - a_i \end{array} \right) \in \mathrm{SOC}(4),\ i = 1, ..., m \\
& s \leqslant 1 - (2\lambda/v)\sum_{i=1}^m p_i s_i \\
& \left(\begin{array}{c} s_i \\ x - a_i \end{array} \right) \in \mathrm{SOC}(3),\ i = 1, ..., m.
\end{array}
$$

Departure from Linear Programming

All is fine and familiar, just like in the case of LP. But is that so? Let us consider some serious discrepancy between SOCP and LP.

A famous result regarding linear programming asserts that a linear programming problem can only be in one of the three states: (1) the problem is infeasible, and any slight perturbation of the problem data will keep the problem infeasible; (2) the problem is feasible but there is no finite optimal value; (3) the problem is feasible and has an optimal solution. Does the same hold true for SOCP? The answer is a definite No!

Consider

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 + x_3 \\ \text{subject to} & x_1 - x_2 = 0 \\ & x_3 = 1 \\ & \left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right) \in \mathrm{SOC}(3). \end{array}$$

The conic constraint requires that $x_1 \geqslant \sqrt{x_2^2 + x_3^2}$. The above problem is clearly infeasible.

For any positive $\epsilon > 0$ the following perturbed version is however always feasible

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 + x_3 \\ \text{subject to} & x_1 - x_2 = \epsilon \\ & x_3 = 1 \\ & \left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right) \in \mathrm{SOC}(3), \end{array}$$

with a feasible solution e.g. $(x_1, x_2, x_3) = (\frac{1}{2\epsilon} + \epsilon, \frac{1}{2\epsilon}, 1)$.

In the same vein, it is possible for SOCP to be *not* infeasible, *not* unbounded in the objective value, yet there is *no* optimal solution. Consider

$$\begin{array}{ll} \text{minimize} & x_1 - x_2 \\ \text{subject to} & x_3 = 1 \\ & \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathrm{SOC}(3). \end{array}$$

In this example, for any $\epsilon > 0$ the solution $(x_1, x_2, x_3) = (\frac{1}{2\epsilon} + \epsilon, \frac{1}{2\epsilon}, 1)$ yields an objective value ϵ. However, there is no attainable solution with value exactly equal to 0. In this sense, it is a bit of abuse to use the term ‘minimize’ in the objective. It might be more appropriate to replace ‘minimize’ by ‘infimum’. In order not to create too many new symbols, we will still use the old notation, as a suboptimal compromise.

The choice of $\mathcal{K}$ being the cone of positive matrices leads to many interesting consequences. Later, we will focus on a number of selected applications for SDP. As an appetizer, here we shall introduce SDP and consider a few examples. First of all, the second order cone can be easily embedded into the cone of positive matrices, by observing the fact that

$$\begin{pmatrix} t \\ x \end{pmatrix} \in \mathrm{SOC}(n+1) \Longleftrightarrow \begin{bmatrix} t, & x^\top \\ x, & tI_n \end{bmatrix} \in \mathcal{S}_+^{(n+1)\times(n+1)}.$$

In SDP, the decision variables are in the matrix form. In the real domain, the standard SDP is

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} C_{ij}X_{ij} \\ \text{subject to} & \sum_{i,j=1}^{n} A_{ij}^{(k)}X_{ij} = b_k,\ k = 1, ..., m \\ & X \in \mathcal{S}_{+}^{n\times n}, \end{array}$$

where, as problem data, $C, A^{(k)} \in \mathcal{S}^{n\times n}$, for $k = 1, ..., m$, and $b \in \mathbf{R}^m$. Observe that $\text{Tr}\,(CX) = \sum_{i,j=1}^{n} C_{ij}X_{ij}$, which is an inner product between the two matrices. To unify the notation, we use $\langle x, y\rangle$ to denote the inner product between x and y, which can be either in the vector form or in the matrix form. In case of SDP, it is often convenient to denote $X \bullet Y$ as the inner-product between X and Y.

Example 3: the Eigenvalue Problems

The most obvious application of SDP is perhaps the problem of finding the largest eigenvalue of a given n by n symmetric matrix A, which can be cast as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - A \in \mathcal{S}_+^{n\times n}. \end{array}$$

If the matrix A itself is a result of design, say $A = A_0 + \sum_{j=1}^m x_j A_j$, then designing the matrix so as to yield the smallest eigenvalue is again an SDP problem:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - A_0 - \sum_{j=1}^m x_j A_j \in \mathcal{S}_+^{n\times n}. \end{array}$$

Moreover, the sum (or the weighted sum) of the first k largest eigenvalues of a matrix can also be expressed by *Linear Matrix Inequalities*. Let $\lambda_1(A) \geqslant \lambda_2(A) \geqslant \cdots \geqslant \lambda_n(A)$ be the n eigenvalues of the symmetric matrix A in descending order of the symmetric matrix A, and let $f_k(A) = \sum_{i=1}^k \lambda_i(A)$. Then,

$$t \geqslant f_k(A) \Longleftrightarrow \exists s \in \mathbf{R},\, Z \in \mathcal{S}^{n\times n} : t - ks - \mathrm{Tr}\,(Z) \geqslant 0,\, Z \succeq 0,\, Z - A + sI \succeq 0.$$

Hence, the problem of designing a matrix so as to minimize the sum of the k largest eigenvalues can be cast as SDP:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & t - ks - \mathrm{Tr}\,(Z) \geqslant 0 \\ & Z + sI - A_0 - \sum_{j=1}^m x_j A_j \in \mathcal{S}_+^{n\times n} \\ & Z \in \mathcal{S}_+^{n\times n}. \end{array}$$

Example 4: Polynomial Optimization

Consider the problem of finding the minimum of a univariate polynomial of degree $2n$:

$$\text{minimize}_x \quad x^{2n} + a_1 x^{2n-1} + \cdots + a_{2n-1}x + a_{2n}.$$

The problem can be cast equivalently as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & x^{2n} + a_1 x^{2n-1} + \cdots + a_{2n-1}x + a_{2n} - t \geqslant 0 \text{ for all } x \in \mathbf{R}. \end{array} \tag{3}$$

It is well known that a univariate polynomial is nonnegative over the real domain if and only if it can be written as a *sum of squares* (SOS), which is equivalent to saying that there must be a positive semidefinite matrix $Z \in \mathcal{S}_+^{(n+1)\times(n+1)}$ such that

$$x^{2n} + a_1 x^{2n-1} + \cdots + a_{2n-1} x + a_{2n} - t = (1, x, x^2, \cdots, x^n) Z (1, x, x^2, \cdots, x^n)^\top.$$

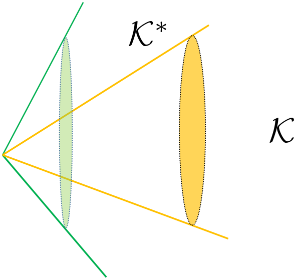
Thus, (3) can be equivalently written as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & a_{2n} - t = Z_{1,1} \\ & a_{2n-k} = \displaystyle\sum_{i+j=k+2} Z_{i,j},\ k = 1, ..., 2n-1 \\ & Z_{(n+1),(n+1)} = 1 \\ & Z \in \mathcal{S}_+^{(n+1)\times(n+1)}. \end{array}$$

Dual Cones

If $\mathcal{K}$ is a convex cone, then its dual cone is defined as

$$\mathcal{K}^* = \{s \mid s^\top x \geqslant 0, \ \forall x \in \mathcal{K}\}.$$



It is easy to verify that: (1) $(\mathbf{R}^n_+)^* = \mathbf{R}^n_+$; (2) $(\mathrm{SOC}(n))^* = \mathrm{SOC}(n)$; (3) $(\mathcal{S}^{n\times n}_+)^* = \mathcal{S}^{n\times n}_+$.

Duality in Conic Optimization

Recall the conic optimization

$$
\begin{array}{lll}
(P) & \text{minimize} & c^\top x \\
 & \text{subject to} & Ax = b \\
 & & x \in \mathcal{K}.
\end{array}
$$

Upper bounding the optimal value yields its dual

$$
\begin{array}{lll}
(D) & \text{maximize} & b^\top y \\
 & \text{subject to} & A^\top y + s = c \\
 & & s \in \mathcal{K}^*.
\end{array}
$$

Theorem 1

(Weak duality theorem) $v(D) \leqslant v(P)$.

The Strong Duality Theorem

Theorem 2

If the primal conic program and its dual conic program both satisfy the Slater condition, then the optimal solution sets for both problems are non-empty and compact. Moreover, the optimal solutions are complementary to each other with zero duality gap.

Barrier Function

Consider a convex barrier function $F(x)$ for $\mathcal{K}$:

- $F(x) < \infty$ for all $x \in \text{int}\, \mathcal{K}$;
- $F(x^k) \to \infty$ as $x^k \to x$ where x is on the boundary of $\mathcal{K}$.

Definition 3

Let $\mathcal{K} \subseteq \mathbf{R}^n$ be a given solid, closed, convex cone, and F be a barrier function defined in $\operatorname{int} \mathcal{K}$. We call F to be a self-concordant function if for any $x \in \operatorname{int} \mathcal{K}$ and any direction $h \in \mathbf{R}^n$ the following two properties are satisfied:

- $|\nabla^3 F(x)[h,h,h]| \leqslant 2(\nabla^2 F(x)[h,h])^{3/2}$;
- $|\nabla F(x)[h]| \leqslant \theta(\nabla^2 F(x)[h,h])^{1/2}$.

Then, F is called a *self-concordant* barrier function for the cone $\mathcal{K}$ with θ as the *complexity value.*

Self-Concordant Barrier for SDP

Consider

$$F(X) = -\log \det X.$$

For any direction $H \in \mathcal{S}^{n\times n}$ we have

$$\nabla F(X)[H] = -\mathrm{Tr}\,(X^{-1}H)$$

and

$$\nabla^2 F(X)[H,H] = \mathrm{Tr}\,(X^{-1}H)^2$$

and

$$\nabla^3 F(X)[H,H,H] = -2\mathrm{Tr}\,(X^{-1}H)^3.$$

One can easily show that it is self-concordant with $\theta = \sqrt{n}$.

Local Geometry

A geometry is associated with a local inner product system. Suppose that $F(x)$ is a strictly convex barrier function for the cone $\mathcal{K}$. Consider

$$\langle u, v \rangle = u^\top \nabla^2 F(x) v.$$

The above inner product is coordinate-free, i.e., if we let $y = A^{-1}x$ then the inner product remains invariant.

To be specific about the locality of the inner product, let us denote

$$\langle u, v \rangle_x := u^\top \nabla^2 F(x) v.$$

The norm induced by the inner product is $\|u\|_x := \sqrt{\langle u, u \rangle_x}$.

Penalized Problem

For

$$
\begin{array}{lll}
(P) & \text{minimize} & c^\top x \\
 & \text{subject to} & x \in a + \mathcal{L} \\
 & & x \in \mathcal{K},
\end{array}
$$

where $\mathcal{L} = \{x \in \mathbf{R}^n \mid Ax = 0\}$, let

$$F_\mu(x) = \frac{1}{\mu} c^\top x + F(x).$$

Obviously, $F_\mu(x)$ is also a barrier function for (P).

For any $0 < \mu' < \mu$, we have

$$
\begin{array}{rcl}
\nabla^2 F_{\mu'}(x) & = & \nabla^2 F_\mu(x) = \nabla^2 F(x) \\
\nabla F_{\mu'}(x) & = & \dfrac{\mu}{\mu'} \nabla F_\mu(x) + \dfrac{\mu' - \mu}{\mu'} \nabla F(x).
\end{array}
$$

Short-step central path following

Let

$$n(\mu; x) = -(\nabla^2 F_\mu(x))^{-1} \nabla F_\mu(x),$$

and

$$p(\mu; x) := \|n(\mu; x)\|_x,$$

namely, the local norm of Newton direction of F_μ.

Newton step. Let $x^{i+1} = x^i + n(\mu^i; x^i)$.

Target shifting. Let μ^{i+1} be so that $p(\mu^{i+1}; x^{i+1}) = 1/4$.

Theorem 4

Suppose that $\mu^0 = O(1)$. Then, in $O(\theta \log \frac{1}{\epsilon})$ Newton steps we will reach a point x with $\mu < \epsilon$ and $p(\mu; x) \leqslant 1/4$.

Long-step central path following

Newton step.

- Let $y = x^i$.
- While $p(\mu^i; y) \geqslant 1/8$, find the Newton direction $n(\mu^i; y)$ and do line minimization

$$t := \text{argmin } F_{\mu^i}(y + tn(\mu^i; y)),$$

and

$$y := y + tn(\mu^i; y)$$

and return to while.

- *Update the iterate*: Let $x^{i+1} = y$.

Target shifting. Let $\mu^{i+1} = \mu^i/2$.

By construction, in the while loop, it holds that $p(\mu^i; y) \geqslant 1/8$ and $p(\mu^i; x^{i+1}) < 1/8$.

Theorem 5

Suppose that $\mu^0 = O(1)$. Then, in $O(\log \frac{1}{\epsilon})$ number of target shifting we will reach a point x with $\mu < \epsilon$ and $p(\mu; x) \leqslant 1/8$. Between each target shifting it takes at most $O(\theta^2)$ numbers of Newton steps.

References

Software tools:

- CVX for *disciplined convex optimization*, developed by Michael Grant and Stephen Boyd:

  http://cvxr.com/cvx/

- YALMIP, developed by Johan Löfberg:

  http://users.isy.liu.se/johanl/yalmip/

More materials for general reading:

- S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.
- J.F. Sturm, *Theory and Algorithms for Semidefinite Programming*. In *High Performance Optimization*, eds. H. Frenk, K. Roos, T. Terlaky, and S. Zhang, Kluwer Academic Publishers, 1999.
- Yu. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, 13, 1994.
- J. Renegar, *A Mathematical View of Interior Point Methods in Convex Optimization*, MPS-SIAM Series on Optimization, 2001.
- Yu. Nesterov, *Introductory Lectures on Convex Optimization*, Kluwer Academic Publishers, 2004.