IE 598: Big Data Optimization **Fall 2016**

Lecture 6: Conic Optimization – September 8

Lecturer: Niao He *Scriber: Juan Xu*

Overview In this lecture, we finish up our previous discussion on optimality conditions for constrained convex programs. We will then introduce an important class of well-structured convex optimization problems – conic programming.

6.1 Recap

In the previous lecture, we have discussed the optimality condition for simple constrained or unconstrained convex problems

- If $f(x)$ is a convex and differentiable function defined on a convex set X, then
$$x_* \in \arg\min_{x\in X} f(x) \quad \Leftrightarrow \quad \nabla f(x_*)^\top (x - x_*) \geq 0 \quad \forall x \in X,$$
- If $f(x)$ is a convex and differentiable function defined on $\mathbb{R}^n$, then
$$x_* \in \arg\min_{x\in \mathbb{R}^n} f(x) \quad \Leftrightarrow \quad \nabla f(x_*) = 0.$$

What about constrained convex problems?

$$\textbf{Primal:} \quad \mathrm{Opt(P)} = \begin{array}{l} \min f(x) \\ \text{s.t. } g_i(x) \leq 0 \quad \forall i = 1, \ldots, m, \\ x \in X. \end{array}$$

$$\textbf{Lagrangian dual:} \quad \mathrm{Opt(D)} = \begin{array}{l} \sup_{\lambda \geq 0} \underline{\mathrm{L}}(\lambda) \\ \text{where } \underline{\mathrm{L}}(\lambda) = \inf_{x\in X} \{f(x) + \sum_{i=1}^m \lambda_i g_i(x)\}. \end{array}$$

Recall the Lagrangian duality theorem,

- Weak duality: Opt(D)$\leq$Opt(P);
- Strong duality: Under slater condition, Opt(P)=Opt(D).

Solving the primal problem is essentially the same as solving its Lagrangian dual problem.

6.2 Constrained Convex Optimization (cont'd)

6.2.1 Saddle Point Formulation

Recall the Lagrangian function:

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x).$$

We can rewrite the primal and its Lagrangian dual problems in the following form.

$$\mathrm{Opt(D)} = \sup_{\lambda \geq 0} \underbrace{\inf_{x\in X} L(x,\lambda)}_{:=\underline{L}(\lambda)}$$

$$\mathrm{Opt(P)} = \inf_{x\in X} \underbrace{\sup_{\lambda \geq 0} L(x,\lambda)}_{:=\overline{L}(x)}$$

Note that $\overline{L}(x) = \begin{cases} f(x) & g_i(x) \leq 0, \forall i \\ +\infty & \text{otherwise} \end{cases}$. Hence, $\inf_{x\in X} \overline{L}(x) = \mathrm{Opt(P)}$.

Definition 6.1 ***(Saddle Point)***
We call (x_*, λ^*), where $x_* \in X$ and $\lambda^* \geq 0$ a saddle point of $L(x,\lambda)$ if and only if $L(x,\lambda^*) \geq L(x_*,\lambda^*) \geq L(x_*,\lambda), \forall x \in X, \forall \lambda \geq 0$.

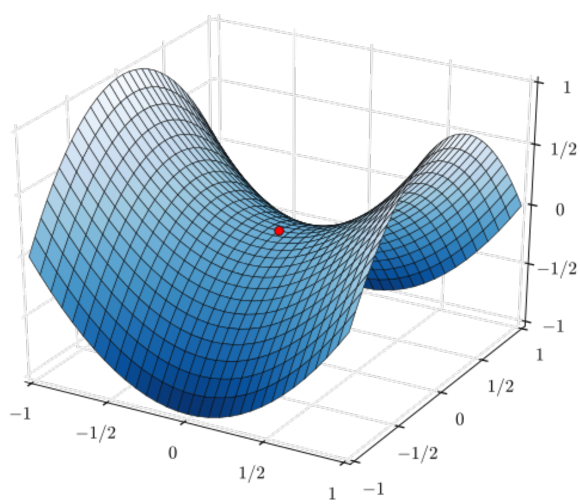

Figure 6.1: Saddle point (Source: https://en.wikipedia.org/wiki/Saddle_point)

Remark. From Figure (6.1), we can see that the saddle point (the red point) is the local minimum point in the x-axis direction, and the local maximum point in the y-axis direction.

Theorem 6.2 *(x_*, λ^*) is a saddle point of $L(x, \lambda)$ if and only if x_* is optimal for (P), λ^* is optimal for (D), and Opt(P)=Opt(D).*

Proof:

- $(\Rightarrow)$: On the one hand, we have

$$\mathrm{Opt}(D) = \sup_{\lambda \geq 0} \inf_{x \in X} L(x, \lambda) \geq \inf_{x \in X} L(x, \lambda^*) = L(x_*, \lambda^*).$$

The last equality is due to the fact that $L(x, \lambda^*) \geq L(x_*, \lambda^*), \forall x \in X$. On the other hand, we have

$$\mathrm{Opt}(D) = \inf_{x \in X} \sup_{\lambda \geq 0} L(x, \lambda) \leq \sup_{\lambda \geq 0} L(x_*, \lambda) = L(x_*, \lambda^*).$$

The last equality is due to the fact that $L(x_*, \lambda^*) \geq L(x_*, \lambda), \forall \lambda \geq 0$. We arrive at Opt(D)$\geq L(x_*, \lambda^*) \geq$ Opt(P), and by weak duality, Opt(D)$\leq$Opt(P)); it can then be shown that Opt(P)=Opt(D), x_* is optimal for (P), and λ^* is optimal for (D).

- $(\Leftarrow)$: Since x_* is optimal for (P), λ^* is optimal for (D), we have

$$\mathrm{Opt}(D) = \underline{L}(\lambda^*) = \inf_{x \in X} L(x, \lambda^*) \leq L(x_*, \lambda^*)$$

$$\mathrm{Opt}(P) = \overline{L}(x_*) = \sup_{\lambda \geq 0} L(x_*, \lambda) \geq L(x_*, \lambda^*)$$

Now that $\mathrm{Opt}(P) = \mathrm{Opt}(D)$, we must have $\mathrm{Opt}(P) = L(x_*, \lambda^*) = \mathrm{Opt}(D)$. That is,

$$\inf_{x \in X} L(x, \lambda^*) = L(x_*, \lambda^*) = \sup_{\lambda \geq 0} L(x_*, \lambda)$$

which implies that (x_*, λ^*) is a saddle point.

■

6.2.2 Optimality conditions for constrained convex problem

Theorem 6.3 ***(Saddle point condition)***
If (P) is a convex optimization problem, and x_* is a feasible solution to (P), then

$$x_* \text{ is optimal for (P)} \xLeftrightarrow{\text{under slater condition}} \exists (x_*, \lambda^*) \text{ is a saddle point of } L(x, \lambda) \tag{6.1}$$

Proof: $(\Rightarrow)$: Under the slater condition, if x_* is the optimal solution of (P), we can find an optimal solution λ^* for the Lagrangian dual problem such that Opt(P)=Opt(D). By Theorem (6.2), (x_*, λ^*) is a saddle point of $L(x, \lambda)$. $(\Leftarrow)$: By Theorem (6.2), if (x_*, λ^*) is a saddle point of $L(x, \lambda)$, then x_* is optimal for (P). ■

Theorem 6.4 ***(Karush-Kuhn-Tucker (KKT) condition)***
If (P) is convex, x_* is a feasible solution to (P), and $f(x)$, $g_i(x)$'s are all differentiable at x_*, then

$$x_* \text{ is optimal for (P)} \xLeftrightarrow{\text{under slater condition}} \exists \lambda^* \geq 0 \text{ s.t. } \begin{array}{ll} (a) & \nabla f(x_*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x_*) \in N_X(x_*); \\ (b) & \lambda_i^* g_i(x_*) = 0, \forall i = 1, \ldots, m \end{array} \tag{6.2}$$

where $N_X(x_*)$ represents the normal cone of x_* ($N_X(x_*) = \{g : g^\top (x - x_*) \geq 0, \forall x \in X\}$).

We refer to (a) as the Lagrangian stationarity condition, (b) the complementary slackness condition. Both (a) and (b) consist of the KKT condition.

Proof: The proof follows directly from the previous theorem. First,

$$L(x,\lambda^*) \geq L(x_*,\lambda^*), \quad \forall x \in X \quad \Leftrightarrow \quad \frac{\partial L}{\partial x}(x_*,\lambda^*)^\top (x - x_*) \geq 0, \quad \forall x \in X \Leftrightarrow \text{ condition } (a)$$

This is because $\frac{\partial L}{\partial x}(x_*,\lambda^*) = \nabla f(x_*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x_*))$. Second,

$$\begin{aligned} L(x_*,\lambda^*) \geq L(x_*,\lambda), \quad \forall \lambda \geq 0 \quad &\Leftrightarrow \quad f(x_*) + \sum_{i=1}^m \lambda_i^* g_i(x_*) \geq f(x_*) + \sum_{i=1}^m \lambda_i g_i(x_*) \quad \forall \lambda \geq 0 \\ &\Leftrightarrow \quad \sum_{i=1}^m \lambda_i^* g_i(x_*) \geq \sum_{i=1}^m \lambda_i g_i(x_*) \quad \forall \lambda \geq 0 \\ &\Leftrightarrow \quad \text{condition } (b) \quad (\text{because } g_i(x_*) \leq 0,\ \forall i = 1,\ldots,m, \text{ and } \lambda \geq 0). \end{aligned}$$

Therefore, (x_*,λ^*) is a saddle point if and only if (a) and (b) hold true. ■

6.3 Conic Optimization

6.3.1 A bit of history

Mid-1940s	Dantzig
	Simplex method for solving linear programming
	Worst cast exponential time (however, fast when implemented)
Mid-1970s	Shor, Khachiyan, Nemirovski, Yudin
	Ellipsoid method for linear programming and convex optimization
	Polynomial time (however, very slow for large problems)
	Complexity: $O(n^2 \log(\frac{1}{\epsilon})) \cdot O(n^2)$
Mid-1980s	Karmarkar, Nemirovski, Nesterov
	Interior point method for linear programming and well-structured convex problem
	Polynomial time
	Complexity: $O(\nu \cdot \log(\frac{\nu}{\epsilon})) \cdot O(n^3)$

The poor performance of some algorithms (e.g. Ellipsoid method) despite of polynomial time stems from their black box oriented nature - these algorithms do not adjust themselves to the structure of the problem and only utilize local information, e.g. the values and (sub)gradients of the objective and the constraints at query points. While in contrast, interior point method is designed to take advantage of the problem's structure and is much more efficient.

6.3.2 From linear programming to conic programming

When passing from a linear program to a nonlinear convex program, we can act as follows

$$\begin{aligned}
&\text{(Linear Programming)} && \min\{c^\top x : Ax - b \geq 0\} \quad (\text{“}\geq\text{” is component-wise}) \\
& &=\;& \min\{c^\top x : Ax - b \in \mathbb{R}^n_+\}.
\end{aligned}$$

$$\begin{aligned}
&\text{(Conic Programming)} && \min\{c^\top x : Ax - b \geq_{\mathbf{K}} 0\} \quad (\text{“}\geq_{\mathbf{K}}\text{” is some order}) \\
& &=\;& \min\{c^\top x : Ax - b \in \mathbf{K}\},
\end{aligned}$$

where $\mathbf{K}$ is a regular (closed, pointed, nonempty interior) cone. The order “$\geq_{\mathbf{K}}$” is defined as $x \geq_{\mathbf{K}} y$ if and only if $x - y \in \mathbf{K}$. It is easy to see that linear programming is a special case of conic programming which will be shown below.

6.3.3 Three typical conic problems

1. **Linear programming (LP)**
   $\mathbf{K} = \mathbb{R}^{\mathbf{n}}_+ = \mathbb{R}_+ \times \cdots \times \mathbb{R}_+$.

   $$\textbf{(LP):} \quad \min\{c^\top x : a_i^\top x - b_i \geq 0, i = 1, \ldots, k\}$$

2. **Second-order cone programming (SOCP, also called conic quadratic programming)**
   $\mathbf{K} = \mathbf{L}^{\mathbf{n_1}} \times \cdots \times \mathbf{L}^{\mathbf{n_k}}$, where $\mathbf{L}^{\mathbf{n}} = \{(\nu_{\mathbf{1}}, \ldots, \nu_{\mathbf{n}}),\ \nu_{\mathbf{n}} \geq \sqrt{\nu_{\mathbf{1}}^{\mathbf{2}} + \ldots \nu_{\mathbf{n-1}}^{\mathbf{2}}}\}$.

   $$\textbf{(SOCP):} \quad \min\{c^\top x : ||A_i x - b_i||_2 \leq c_i^\top x - d_i, i = 1, \ldots, k\}$$

   One can treat $Ax - b = \left[A_1 x - b_1, c_1^\top x - d_1; \cdots; A_k x - b_k, c_k^\top x - d_k\right]$.

3. **Semi-definite programming (SDP)**
   $\mathbf{k} = \mathbf{S}^{\mathbf{n_1}}_+ \times \cdots \times \mathbf{S}^{\mathbf{n_k}}_+$, where $\mathbf{S}^{\mathbf{n}}_+ = \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{X} = \mathbf{X}^\top\}$.

   $$\textbf{(SDP):} \quad \min\{c^\top x : A_i x - b_i \succeq 0, i = 1, \ldots, k\};$$

   This can be simplified as

   $$\textbf{(SDP):} \quad \min\{c^\top x : \mathcal{A}x - b \succeq 0\},$$

   because

   $$\mathcal{A}x - b := \begin{bmatrix} A_1 x - b_1 & & \\ & \ddots & \\ & & A_k x - b_k \end{bmatrix} \succeq 0 \Leftrightarrow A_i x - b_i \succeq 0,\ \forall i = 1, \ldots, k.$$

Lemma. (Schur complement for symmetric positive semidefinite matrices)
If $S = \begin{bmatrix} P & R^\top \\ R & Q \end{bmatrix}$ is symmetric with $Q \succ 0$, then $S \succeq 0 \Leftrightarrow P - R^\top Q^{-1} R \succeq 0$.

Remark. We show that **(LP)**$\subseteq$**(SOCP)**$\subseteq$**(SDP)**.

- **(LP)⊆(SOCP)**

$$a_i^\top x - b_i \geq 0 \Leftrightarrow \begin{bmatrix} 0 \\ a_i^\top x - b_i \end{bmatrix} \geq_{\mathbf{L^2}} 0.$$

- **(SOCP)⊆(SDP)**

$$\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \in \mathbf{L^m} \Leftrightarrow \begin{bmatrix} x_m & x_1 & \dots & x_{m-1} \\ x_1 & x_m & & \\ \vdots & & \ddots & \\ x_{m-1} & & & x_m \end{bmatrix} \succeq 0 \xLeftrightarrow{\textbf{by Schur lemma}} x_m \geq \sqrt{x_1^2 + \cdots + x_{m-1}^2}.$$

Note that the above matrix is linear in x and can be written as $\sum_i x_i A_i$ with properly defined A_i.

6.3.4 Calculus of Conic Programming

Let $\mathcal{K}$ be a family of regular cones, e.g., $\mathbb{R}^n_+, \mathbf{L^n}, \mathbf{S^n_+}$, or their inner points.

Definition 6.5 *X is $\mathcal{K}$-representable if $\exists \mathbf{K} \in \mathcal{K}$, s.t. $X = \{x : \exists u, Ax + Bu - b \in \mathbf{K}\}$.*

Definition 6.6 *$f(x)$ is $\mathcal{K}$-representable if epi(f) is $\mathcal{K}$-representable, i.e., $\exists \mathbf{K} \in \mathcal{K}$, s.t.,*

$$epi(f) = \{(x,t) : \exists v, Cx + dt + Dv - e \in \mathbf{K}\}.$$

Remark It can be shown that many convexity-preserving operations preserve $\mathcal{K}$-representability, e. g

- taking intersections, direct product, affine image of $\mathcal{K}$-representable sets;
- taking conic combination, affine composition, partial minimization of $\mathcal{K}$-representable functions.

Here we provide some examples:

Examples (SOCP-r functions)

- $f(x) = ||x||_2$
  $epi(f) = \{(x,t) : t \geq ||x||_2\} \Leftrightarrow \{(x,t) : \begin{bmatrix} x \\ t \end{bmatrix} \in \mathbf{L^{n+1}}\}.$
- $f(x) = x^\top x$
  $epi(f) = \{(x,t) : t \geq x^\top x\} \Leftrightarrow \{(x,t) : (t+1)^2 \geq (t-1)^2 + 4x^\top x\} \Leftrightarrow \{(x,t) : \begin{bmatrix} 2x \\ t-1 \\ t+1 \end{bmatrix} \in \mathbf{L^{n+2}}\}.$
- $f(x) = x^\top Qx + q^\top x + t$, where $Q = L \cdot L^\top$
  $epi(f) = \{(x,t) : t \geq x^\top Qx + q^\top x + t\} \Leftrightarrow \{(x,t) : \begin{bmatrix} 2L^\top x \\ t - q^\top x - r - 1 \\ t - q^\top x - r + 1 \end{bmatrix} \in \mathbf{L^{n+2}}\}.$

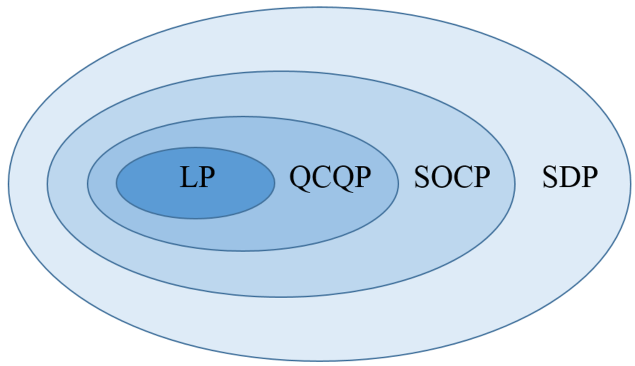

Figure 6.2: Relationships among LP, QCQP, SOCP, SDP

(QCQP refers to quadratically constrained quadratic programming.)

References

[BV04] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization.* Cambridge university press.

[DN01] Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, (Vol. 2). SIAM.