IE 521 Convex Optimization

Lecture 19: Interior Point Method for Conic Programs

Niao He

15th April 2019
Outline

Recap

IPM for Conic Programs
Conic Programs
Self-concordant Barriers for LP, SOCP, SDP
IPM Complexity for LP
Primal-Dual Path Following IPM
Recap: Path Following Scheme

\[(P): \min_x c^T x \quad \Rightarrow \quad (P_t): \min_x \underbrace{t \cdot c^T x + F(x)}_{F_t(x)}\]

\[\text{S.t. } x \in X\]

- **Self-concordant Barrier:**

  \[|D^3 F(x)[h, h, h]| \leq 2(D^2 F(x)[h, h])^{3/2}\]

  \[|DF(x)[h]| \leq \sqrt{\nu} (D^2 F(x)[h, h])^{1/2}\]

- **Newton’s Method:**

  \[x_{k+1} = x_k - [\nabla^2 F(x_k)]^{-1}[t_{k+1} c + \nabla F(x_k)]\]

- **Update Policy:**

  \[t_{k+1} = t_k (1 + \frac{\gamma}{\sqrt{\nu}})\]

- **Initialization:**

  \[(x_0, t_0) \text{ such that } \lambda_{F_t_0}(x_0) \text{ is small.}\]
Recap: Path Following Scheme

0. Initialize \((x_0, t_0)\) with \(t_0 > 0\) and \(\lambda_{F_{t_0}}(x_0) \leq \beta \in (0, 1)\)

1. For \(k \geq 0\), do

\[
\begin{align*}
t_{k+1} &= t_k \left(1 + \frac{\gamma}{\sqrt{\nu}}\right) \\
x_{k+1} &= x_k - \left[\nabla^2 F(x_k)\right]^{-1} \left[ t_{k+1}c + \nabla F(x_k) \right]
\end{align*}
\]

Theorem. In the above scheme, one has

\[
c^T x_k - \min_{x \in X} c^T x \leq O(1) \frac{\nu}{t_0} \exp \left\{ -O(1) \frac{k}{\sqrt{\nu}} \right\}
\]

where the constant factor \(O(1)\) depends solely on \(\beta\) and \(\gamma\).
Recap: Path Following Scheme

- Number of Newton steps for initialization phase:
  \[ N_{\text{init}} \leq O\left(\sqrt{\nu} \log \nu\right) \]

- Number of Newton steps for main phase:
  \[ N_{\text{main}} \leq O\left(\sqrt{\nu} \log \frac{\nu}{\varepsilon}\right) \]

- Total arithmetic cost of finding an \(\varepsilon\)-solution:
  \[ O\left(M \sqrt{\nu} \log \left(\frac{\nu}{\varepsilon} + 1\right)\right) \]

where \(M\) is the arithmetic cost for a Newton’ step.
Conic Program

Primal Conic Program:

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax - b \in \mathcal{K} \\
\end{align*}
\]  
\hspace{1cm} (CP)

Dual Conic Program:

\[
\begin{align*}
\text{max} & \quad b^T y \\
\text{s.t.} & \quad A^T y = c \\
& \quad y \in \mathcal{K}_* \\
\end{align*}
\]  
\hspace{1cm} (CD)
LP, SOCP, SDP

- **Linear Program:**
  \[ \mathcal{K} = \mathbb{R}_+^m = \{ x \in \mathbb{R}^m : x_i \geq 0, i = 1, ..., m \} \]
  \[ \min_x \left\{ c^T x : a_i^T x - b_i \geq 0, i = 1, ..., m \right\} \quad (LP) \]

- **Second-order Conic Program:**
  \[ \mathcal{K} = \prod_{i=1}^m L^{n_i}, L^n = \{ x \in \mathbb{R}^n : x_n^2 \geq \sum_{i=1}^{n-1} x_i^2 \} \]
  \[ \min_x \left\{ c^T x : \|D_i x - d_i\|_2 \leq e_i^T x - f_i, i = 1, ..., m \right\} \quad (SOCP) \]

- **Semidefinite Program:**
  \[ \mathcal{K} = S_+^m = \{ X \in S^m : X \succeq 0 \} \]
  \[ \min_x \left\{ c^T x : \sum_{i=1}^n x_i A_i - B \succeq 0 \right\} \quad (SDP) \]
Self-concordant Barriers on Conic Programs

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax - b \in \mathcal{K} \quad \text{(CP)}
\end{align*}
\]

- Note that the feasible domain is
  \[ X := \{ x : Ax - b \in \mathcal{K} \} \]

- If \( F(y) \) is a self-concordant barrier for \( \mathcal{K} \), then
  \[
  \tilde{F}(x) := F(Ax - b)
  \]
  is a self-concordant barrier for \( X \). (why?)

- Q. How to construct barriers for \( \mathcal{K} = \mathbb{R}_+^m, L^n, S_+^m \)?
Self-concordant Barriers for $\mathbb{R}^m_+$

Example 1.

$$F(y) = -\sum_{j=1}^{m} \ln(y_j)$$ is m-s.c.b. on $\mathbb{R}^m_+$.

- $F(y)$ is convex
- $F(y)$ is standard self-concordant
- $F(y)$ is a barrier: $\nabla^2 F(x) \succeq \frac{1}{2} \nabla F(x)[\nabla F(x)]^T$
Self-concordant Barriers for $L^n$

Example 2.

$$F(y) = - \ln(y_n^2 - y_1^2 - \ldots - y_{n-1}^2)$$ is 2-s.c.b. on $L^n$.

- $F(y)$ is convex (why?)
- $F(y)$ is standard self-concordant (why?)
- $F(y)$ is a barrier: $\nabla^2 F(x) \succeq \frac{1}{2} \nabla F(x)[\nabla F(x)]^T$ (why?)
Self-concordant Barriers for $S^+_m$

Example 3.

$$F(X) = -\ln(\det(X)) = -\sum_{j=1}^{m} \ln(\lambda_j(X)) \text{ is } m\text{-s.c.b. on } S^+_m.$$  

- Given $X \in \text{int}(S^+_m)$ and $H \in S^m$, define 
  
  $$\phi(t) = F(X + tH) = -\ln(\det(X + tH))$$

  $$= -\sum_{j=1}^{m} \ln(1 + t\lambda_j(X^{-1/2}HX^{-1/2})) + \phi(0)$$

- $\phi(t)$ is a $m$-self-concordant barrier.

  $$\phi'(0) = -\sum_{j=1}^{m} \lambda_j, \quad \phi''(0) = \sum_{j=1}^{m} \lambda_j^2, \quad \phi'''(0) = -\sum_{j=1}^{m} \lambda_j^3$$

- $\phi(t)$ is convex since $\phi''(0) \geq 0$.
- $\phi(t)$ is standard self-concordant: $|\phi'''(0)| \leq 2[\phi''(0)]^{3/2}$
- $\phi(t)$ is a barrier: $|\phi'(0)^2| \leq n\phi''(0)$. 

Interior Point Method for Linear Program

\[
\min \quad c^T x \quad \text{s.t.} \quad a_j^T x \geq b_j, \quad j = 1, \ldots, m \quad (m > n) \quad (LP)
\]

- The barrier function

\[
\tilde{F}(x) = -\sum_{j=1}^{m} \ln(a_j^T x - b_j) \text{ is } m\text{-s.c.b.}
\]

- The gradient and Hessian

\[
\nabla \tilde{F}(x) = -\sum_{j=1}^{m} \frac{a_j}{a_j^T x - b_j}, \quad \nabla^2 \tilde{F}(x) = \sum_{j=1}^{m} \frac{a_j a_j^T}{(a_j^T x - b_j)^2}
\]

- Newton’s step:

\[
x_{k+1} = x_k - \left[\nabla^2 \tilde{F}(x_k)\right]^{-1}\left[t_{k+1} c + \nabla \tilde{F}(x_k)\right]
\]
Complexity of Solving Linear Programs

Interior Point Method

- Computing $\nabla \tilde{F}(x)$, $\nabla^2 \tilde{F}(x)$ require $O(mn)$, $O(mn^2)$
- Computing a Newton step requires $O(n^3)$
- Number of iterations is $O(\sqrt{m} \log(\frac{m}{\epsilon}))$
- The overall complexity of finding a $\epsilon$-solution is

$$O(mn^2)O(\sqrt{m} \log(\frac{m}{\epsilon})) = O(m^{3/2} n^2 \log(\frac{m}{\epsilon}))$$

Ellipsoid Method

- Separation oracle requires $O(mn)$
- Computing new ellipsoid requires $O(n^2)$
- Number of iterations is $O(n^2 \log(\frac{1}{\epsilon}))$
- The overall complexity of finding a $\epsilon$-solution is

$$O(mn + n^2)O(n^2 \log(\frac{1}{\epsilon})) = O(mn^3 \log(\frac{1}{\epsilon}))$$
Primal-Dual Path Following Schemes

When solving conic programs, ideally we would like to develop interior point methods that

▶ produce primal-dual pairs at each iteration
▶ handle equality constraints
▶ require no prior knowledge of a strictly feasible solution
▶ adjust penalty based on current solution

Key idea: To approximate the KKT conditions.

We will focus on the SDP case.
Primal and Dual SDP

Primal problem

\[
\begin{align*}
\text{min} & \quad \text{tr}(CX) \\
\text{s.t.} & \quad \text{tr}(A_iX) = b_i, \ i = 1, \ldots, m \\
& \quad X \succeq 0
\end{align*}
\quad (P)
\]

Dual problem

\[
\begin{align*}
\max_{y,Z} & \quad b^T y \\
\text{s.t.} & \quad \sum_{i=1}^{m} y_i A_i + Z = C \\
& \quad Z \succeq 0
\end{align*}
\quad (D)
\]

Assume \((P)\) and \((D)\) are strictly primal-dual feasible, so there is no duality gap.
Barrier Problems

Barrier problem of the primal:

\[
\min \quad \text{tr}(CX) - \mu \ln(\det(X))
\]
\[
\text{s.t.} \quad \text{tr}(A_i X) = b_i, \quad i = 1, \ldots, m \quad (BP)
\]

Barrier problem of the dual:

\[
\max_{y, Z} \quad b^T y + \mu \log(\det(Z))
\]
\[
\text{s.t.} \quad \sum_{i=1}^{m} y_i A_i + Z = C \quad (BD)
\]

These are indeed the Lagrange duals to each other, up to constant.
KKT Conditions

KKT conditions for (P) and (D):

\[
X^* \succeq 0, \quad Z^* \succeq 0 \\
\text{tr}(A_iX^*) = b_i, \quad i = 1, \ldots, m \\
\sum_{i=1}^m y_i^* A_i + Z^* = C \\
X^*Z^* = 0 \quad \text{(complementary slackness)}
\]

KKT conditions for (BP) and (BD):

\[
\text{tr}(A_iX^*(\mu)) = b_i, \quad i = 1, \ldots, m \\
\sum_{i=1}^m y_i^*(\mu)A_i + Z^*(\mu) = C \\
X^*(\mu)Z^*(\mu) = \mu I \quad \text{(complementary slackness)}
\]
Primal-Dual Central Path

Primal-dual central path: \{ (X^*(\mu), y^*(\mu), Z^*(\mu)) : \mu > 0 \}.

- The duality gap at \((X^*(\mu), y^*(\mu))\) is
  \[
  \text{tr}(CX^*(\mu)) - b^T y^*(\mu) = \text{tr}(Z^*(\mu)X^*(\mu)) = \mu n
  \]

- As \(\mu \to 0\), the duality gap is zero:
  \[
  (X^*(\mu), y^*(\mu), Z^*(\mu)) \to (X^*, y^*, Z^*)
  \]
Newton Step: Solving KKT Equations

Find direction \((\Delta X, \Delta y, \Delta Z)\) by solving the equations:

\[
\begin{align*}
\text{tr}(A_i(X + \Delta X)) &= b_i, \quad i = 1, \ldots, m \\
\sum_{i=1}^{m}(y_i + \Delta y_i)A_i + (Z + \Delta Z) &= C \\
(X + \Delta X)(Z + \Delta Z) &= \mu I
\end{align*}
\]

\[
\downarrow
\]

\[
\begin{align*}
\text{tr}(A_i\Delta X) &= 0, \quad i = 1, \ldots, m \\
\sum_{i=1}^{m}\Delta y_i A_i + \Delta Z &= 0 \quad (\star) \\
(X + \Delta X)(Z + \Delta Z) &= \mu I
\end{align*}
\]
Primal-Dual Path Following Scheme

0. Initialize $(X, y, Z) = (X_0, y_0, Z_0)$ with $X_0 > 0, Z_0 > 0$

1. For $k \geq 0$, do
   - compute $\mu = \frac{\text{tr}(XZ)}{n}$, $\mu \leftarrow \frac{\mu}{2}$
   - compute $(\Delta X, \Delta y, \Delta Z)$ by solving the equations $(\star)$
   - update $(X, y, Z) \leftarrow (X + \alpha \Delta X, y + \beta \Delta y, Z + \beta \Delta Z)$
     with proper $\alpha, \beta$ that preserves positivity of $(X, Z)$.

Remark. (Approximation of KKT equation). One can apply first-order approximation to the only nonlinear system:

$$\mu = (X + \Delta X)(Z + \Delta Z) \approx XZ + \Delta XZ + X \Delta Z$$

and solve the linearized KKT equations.
References

- Nesterov (2004), Introductory Lectures on Convex Optimization, Chapter 4.1.4-5
- Nemirovski (2004), Interior Point Polynomial Time Methods in Convex Programming, Chapter 3-4