

CME 307 / MS&E 311: Optimization

Interior Point Methods

Professor Udell

Management Science and Engineering
Stanford

March 1, 2024

with Prof. Luiz-Rafael Santos, UFSC <https://lrsantos11.github.io/>

Convex optimization

Convex optimization problem

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & Ax = b\end{array}$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $g : \mathbf{R}^n \rightarrow \mathbf{R}^p$ are smooth and convex, $A \in \mathbf{R}^{m \times n}$ is full rank.

Convex optimization

Convex optimization problem

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \\ & Ax = b\end{array}$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $g : \mathbf{R}^n \rightarrow \mathbf{R}^p$ are smooth and convex, $A \in \mathbf{R}^{m \times n}$ is full rank.

KKT conditions:

Convex optimization

Convex optimization problem

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \quad s \in \mathbf{R}^p \\ & Ax = b \quad y \in \mathbf{R}^m\end{array}$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $g : \mathbf{R}^n \rightarrow \mathbf{R}^p$ are smooth and convex, $A \in \mathbf{R}^{m \times n}$ is full rank.

KKT conditions:

Convex optimization

Convex optimization problem

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0 \quad s \in \mathbf{R}^p \\ & Ax = b \quad y \in \mathbf{R}^m\end{array}$$

where $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $g : \mathbf{R}^n \rightarrow \mathbf{R}^p$ are smooth and convex, $A \in \mathbf{R}^{m \times n}$ is full rank.

KKT conditions:

$$\begin{aligned}\nabla f(x) + A^T y + (\nabla g(x))^T s &= 0 \\ Ax &= b \\ g(x) &\leq 0 \\ s &\geq 0 \\ s_j g_j(x) &= 0, \quad j = 1, \dots, p\end{aligned}$$

Outline

IPM for linear and quadratic programs

IPM for Convex nonlinear programming

IPM for Conic Optimization

IPM for linear and quadratic programs

Linear/Quadratic Program

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \\ & x \geq 0,\end{array}$$

where $Q \in \mathbf{S}_+^n$, and $A \in \mathbf{R}^{m \times n}$ is full-rank.

IPM for linear and quadratic programs

Linear/Quadratic Program

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \\ & x \geq 0,\end{array}$$

where $Q \in \mathbf{S}_+^n$, and $A \in \mathbf{R}^{m \times n}$ is full-rank.

- ▶ $\mathcal{P} := \{x \in \mathbf{R}^n \mid Ax = b, x \geq 0\}$ is a polyhedron.

IPM for linear and quadratic programs

Linear/Quadratic Program

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \\ & x \geq 0,\end{array}$$

where $Q \in \mathbf{S}_+^n$, and $A \in \mathbf{R}^{m \times n}$ is full-rank.

- ▶ $\mathcal{P} := \{x \in \mathbf{R}^n \mid Ax = b, x \geq 0\}$ is a polyhedron.
- ▶ If $Q = 0$, problem is a linear program.

IPM for linear and quadratic programs

Linear/Quadratic Program

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \\ & x \geq 0,\end{array}$$

where $Q \in \mathbf{S}_+^n$, and $A \in \mathbf{R}^{m \times n}$ is full-rank.

- ▶ $\mathcal{P} := \{x \in \mathbf{R}^n \mid Ax = b, x \geq 0\}$ is a polyhedron.
- ▶ If $Q = 0$, problem is a linear program.

How to solve LP/QP problems?

IPM for linear and quadratic programs

Linear/Quadratic Program

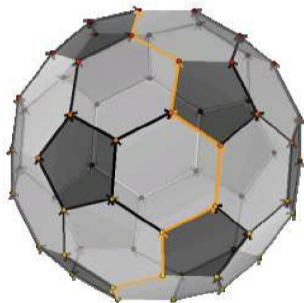
$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \\ & x \geq 0,\end{array}$$

where $Q \in \mathbf{S}_+^n$, and $A \in \mathbf{R}^{m \times n}$ is full-rank.

- ▶ $\mathcal{P} := \{x \in \mathbf{R}^n \mid Ax = b, x \geq 0\}$ is a polyhedron.
- ▶ If $Q = 0$, problem is a linear program.

How to solve LP/QP problems?

Simplex: vertex to vertex
IPM: go through the middle!



IPM for linear and quadratic programs

Linear/Quadratic Program

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b, \\ & x \geq 0,\end{array}$$

where $Q \in \mathbf{S}_+^n$, and $A \in \mathbf{R}^{m \times n}$ is full-rank.

- ▶ $\mathcal{P} := \{x \in \mathbf{R}^n \mid Ax = b, x \geq 0\}$ is a polyhedron.
- ▶ If $Q = 0$, problem is a linear program.

How to solve LP/QP problems?

Advantages of vertex solution vs interior solution?

Simplex: vertex to vertex
IPM: go through the middle!



Building blocks of IPM

Ingredients for Interior Point Method

- ▶ Duality theory: Lagrangian function; KKT (first order optimality) condition.
- ▶ Barrier function: logarithmic barrier.
- ▶ Newton's method (and a good linear solver)

Building blocks of IPM

Ingredients for Interior Point Method

- ▶ Duality theory: Lagrangian function; KKT (first order optimality) condition.
- ▶ Barrier function: logarithmic barrier.
- ▶ Newton's method (and a good linear solver)

The reward: fantastic convergence properties!

- ▶ Theoretical: $O(\sqrt{n} \log(1/\varepsilon))$ iterations
- ▶ Practical: $O(\log n \log(1/\varepsilon))$ iterations

(but the per-iteration cost may be high due to the Newton solve: often $O(n^3)$)

IPM: algorithmic template

IPM procedure

- ▶ replace inequalities with log barriers;
- ▶ form the Lagrangian;
- ▶ write down the KKT conditions of the perturbed problem;
- ▶ find one (or more) directions using [Newton's method](#) on the KKT system;
- ▶ (decide how to combine the directions and) compute a stepsize.

Duality and KKT conditions

Primal-dual QPs

Primal problem

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b \\ & x \geq 0\end{array}$$

Dual problem

$$\begin{array}{ll}\text{maximize} & b^\top y - \frac{1}{2}x^\top Qx \\ \text{subject to} & A^\top y + s - Qx = c \\ & s \geq 0\end{array}$$

Duality and KKT conditions

Primal-dual QPs

Primal problem

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b \\ & x \geq 0\end{array}$$

Dual problem

$$\begin{array}{ll}\text{maximize} & b^\top y - \frac{1}{2}x^\top Qx \\ \text{subject to} & A^\top y + s - Qx = c \\ & s \geq 0\end{array}$$

KKT conditions

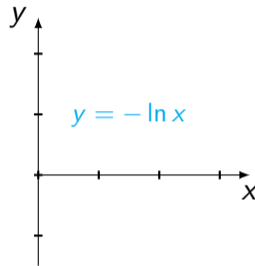
$$\begin{aligned}Ax &= b \\ A^\top y + s - Qx &= c \\ XSe &= 0 \\ (x, s) &\geq 0\end{aligned}$$

▷ complementarity: $x \cdot s = 0$

where $X = \mathbf{diag}(x_1, \dots, x_n)$, $S = \mathbf{diag}(s_1, \dots, s_n) \in \mathbf{R}^{n \times n}$, and $e = (1, \dots, 1) \in \mathbf{R}^n$.

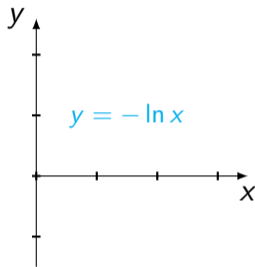
Logarithmic barrier

$-\ln x_j$
replaces the inequality
 $x_j \geq 0$



Logarithmic barrier

$-\ln x_j$
replaces the inequality
 $x_j \geq 0$



$$\text{minimize} \quad -\sum_{j=1}^n \ln x_j \quad \Longleftrightarrow \quad \text{maximize} \quad \prod_{1 \leq j \leq n} x_j$$

\Rightarrow keeps every entry of x away from 0.

Barrier primal QP

Step 1: replace inequality constraints by barrier

Replace the primal QP

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx \\ \text{subject to} & Ax = b \\ & x \geq 0\end{array}$$

with the barrier primal QP

$$\begin{array}{ll}\text{minimize} & c^\top x + \frac{1}{2}x^\top Qx - \mu \sum_{j=1}^n \ln x_j \\ \text{subject to} & Ax = b\end{array}$$

Logarithmic barrier and stationarity

Step 2: remove equality constraints using Lagrangian

$$\mathcal{L}(x, y, \mu) = c^\top x + \frac{1}{2} x^\top Q x - y^\top (Ax - b) - \mu \sum_{j=1}^n \ln x_j$$

Logarithmic barrier and stationarity

Step 2: remove equality constraints using Lagrangian

$$\mathcal{L}(x, y, \mu) = c^\top x + \frac{1}{2}x^\top Qx - y^\top (Ax - b) - \mu \sum_{j=1}^n \ln x_j$$

A stationary point (x, y, μ) of the Lagrangian satisfies

$$\nabla_x \mathcal{L}(x, y, \mu) = 0 \qquad = c + Qx - A^\top y - \mu X^{-1}e$$

with $X^{-1} = \mathbf{diag}(x_1^{-1}, \dots, x_n^{-1}) \in \mathbf{R}^{n \times n}, (x_j > 0)$.

KKT conditions for barrier problem

- Define $s := \mu X^{-1}e$, which implies $XSe = \mu e$, to get

KKT _{μ}

$$Ax = b$$

$$A^T y + s - Qx = c$$

$$XSe = \mu e$$

$$(x, s) > 0$$

KKT conditions for barrier problem

- Define $s := \mu X^{-1}e$, which implies $XSe = \mu e$, to get

KKT _{μ}

$$\begin{aligned}Ax &= b \\ A^\top y + s - Qx &= c \\ XSe &= \mu e \\ (x, s) &> 0\end{aligned}$$

KKT

$$\begin{aligned}Ax &= b \\ A^\top y + s - Qx &= c \\ XSe &= 0 \\ (x, s) &\geq 0\end{aligned}$$

KKT conditions for barrier problem

- Define $s := \mu X^{-1}e$, which implies $XSe = \mu e$, to get

KKT _{μ}

$$\begin{aligned}Ax &= b \\ A^\top y + s - Qx &= c \\ XSe &= \mu e \\ (x, s) &> 0\end{aligned}$$

KKT

$$\begin{aligned}Ax &= b \\ A^\top y + s - Qx &= c \\ XSe &= 0 \\ (x, s) &\geq 0\end{aligned}$$

KKT _{μ} \rightarrow KKT as $\mu \rightarrow 0$.

Central path (LP case)

- ▶ Parameter μ controls the distance to optimality

$$c^\top x - b^\top y = c^\top x - x^\top A^\top y = x^\top s = n\mu$$

Central path (LP case)

- ▶ Parameter μ controls the distance to optimality

$$c^\top x - b^\top y = c^\top x - x^\top A^\top y = x^\top s = n\mu$$

- ▶ Analytic center (μ -center): unique point

$$(x(\mu), y(\mu), s(\mu)), \quad x(\mu) > 0, \quad s(\mu) > 0$$

that satisfies the KKT_μ conditions.

Central path (LP case)

- ▶ Parameter μ controls the distance to optimality

$$c^\top x - b^\top y = c^\top x - x^\top A^\top y = x^\top s = n\mu$$

- ▶ Analytic center (μ -center): unique point

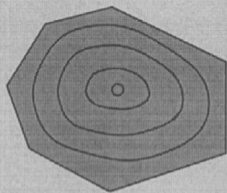
$$(x(\mu), y(\mu), s(\mu)), \quad x(\mu) > 0, \quad s(\mu) > 0$$

that satisfies the KKT_μ conditions.

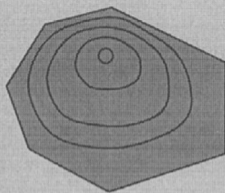
- ▶ The curve

$$\mathcal{C}_\mu = \{(x(\mu), y(\mu), s(\mu)) \mid \mu > 0\}$$

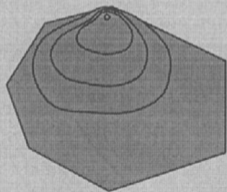
is called the primal-dual central path.



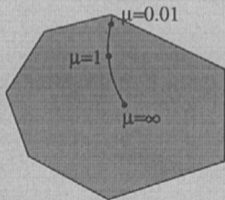
(a) $\mu = \infty$



(b) $\mu = 1$



(c) $\mu = 0.01$



(d) central path

Recall Newton's method for nonlinear equation

- For $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$ smooth, solve $F(x) = 0$.

Recall Newton's method for nonlinear equation

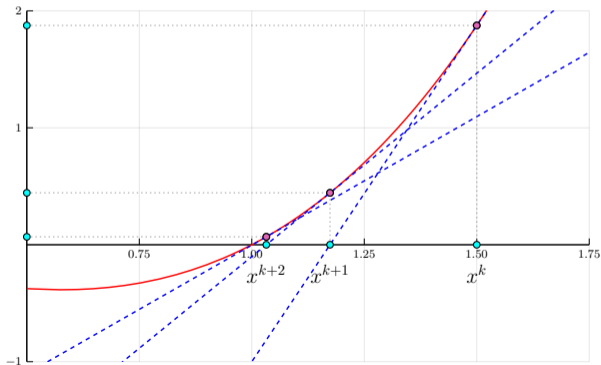
- ▶ For $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$ smooth, solve $F(x) = 0$.
- ▶ Newton's method: define Jacobian $J_F(x)$ so $J_F(x)_{ij} = \frac{\partial F_i}{\partial x_j}$, and iterate

$$x^{k+1} = x^k - \alpha_k J_F(x^k)^{-1} F(x^k)$$

Recall Newton's method for nonlinear equation

- ▶ For $F : \mathbf{R}^n \rightarrow \mathbf{R}^n$ smooth, solve $F(x) = 0$.
- ▶ Newton's method: define Jacobian $J_F(x)$ so $J_F(x)_{ij} = \frac{\partial F_i}{\partial x_j}$, and iterate

$$x^{k+1} = x^k - \alpha_k J_F(x^k)^{-1} F(x^k)$$



Apply Newton Method to KKT_μ

The first order optimality conditions for the barrier problem form a large system of nonlinear equations:

$$F(x, y, s) = 0,$$

where $F : \mathbf{R}^{2n+m} \mapsto \mathbf{R}^{2n+m}$ is defined as

$$F(x, y, s) = \begin{bmatrix} Ax & -b \\ A^\top y + s - Qx & -c \\ XSe & -\mu e \end{bmatrix}$$

Apply Newton Method to KKT_μ

The first order optimality conditions for the barrier problem form a large system of nonlinear equations:

$$F(x, y, s) = 0,$$

where $F : \mathbf{R}^{2n+m} \mapsto \mathbf{R}^{2n+m}$ is defined as

$$F(x, y, s) = \begin{bmatrix} Ax & -b \\ A^\top y + s - Qx & -c \\ XSe & -\mu e \end{bmatrix}$$

- ▶ The first two blocks are **linear**.
- ▶ The last block, corresponding to the complementarity condition, is **nonlinear**.
- ▶ Jacobian is

$$J_F(x, y, s) = \begin{bmatrix} A & 0 & 0 \\ -Q & A^\top & I \\ S & 0 & X \end{bmatrix}$$

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

- ▶ Given (x_0, y_0, s_0) feasible, $\mu_0 = \frac{1}{n} \cdot (x^0)^\top s^0$

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

- ▶ Given (x_0, y_0, s_0) feasible, $\mu_0 = \frac{1}{n} \cdot (x^0)^\top s^0$
- ▶ For $k = 1, 2, \dots$

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

- ▶ Given (x_0, y_0, s_0) feasible, $\mu_0 = \frac{1}{n} \cdot (x^0)^\top s^0$
- ▶ For $k = 1, 2, \dots$
 - ▶ $\mu_k = \sigma \mu_{k-1}$, where $\sigma \in (0, 1)$

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

- ▶ Given (x_0, y_0, s_0) feasible, $\mu_0 = \frac{1}{n} \cdot (x^0)^\top s^0$
- ▶ For $k = 1, 2, \dots$
 - ▶ $\mu_k = \sigma \mu_{k-1}$, where $\sigma \in (0, 1)$
 - ▶ Find Newton direction $(\Delta x^k, \Delta y^k, \Delta s^k)$ by solving

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^\top & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} b - Ax^k \\ c - A^\top y^k - s^k + Qx^k \\ \mu_k e - X^k S^k e \end{bmatrix}$$

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

- ▶ Given (x_0, y_0, s_0) feasible, $\mu_0 = \frac{1}{n} \cdot (x^0)^\top s^0$
- ▶ For $k = 1, 2, \dots$
 - ▶ $\mu_k = \sigma \mu_{k-1}$, where $\sigma \in (0, 1)$
 - ▶ Find Newton direction $(\Delta x^k, \Delta y^k, \Delta s^k)$ by solving

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^\top & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} b - Ax^k \\ c - A^\top y^k - s^k + Qx^k \\ \mu_k e - X^k S^k e \end{bmatrix}$$

- ▶ Find step length α_k so $(x^k + \alpha_k \Delta x^k, y^k + \alpha_k \Delta y^k, s^k + \alpha_k \Delta s^k)$ is feasible.

Interior-point QP Algorithm

IPM Framework

Fix the barrier parameter μ and make *one* (damped) Newton step towards the solution of FOC. Then reduce the barrier parameter μ and repeat.

- ▶ Given (x_0, y_0, s_0) feasible, $\mu_0 = \frac{1}{n} \cdot (x^0)^\top s^0$
- ▶ For $k = 1, 2, \dots$
 - ▶ $\mu_k = \sigma \mu_{k-1}$, where $\sigma \in (0, 1)$
 - ▶ Find Newton direction $(\Delta x^k, \Delta y^k, \Delta s^k)$ by solving

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^\top & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} b - Ax^k \\ c - A^\top y^k - s^k + Qx^k \\ \mu_k e - X^k S^k e \end{bmatrix}$$

- ▶ Find step length α_k so $(x^k + \alpha_k \Delta x^k, y^k + \alpha_k \Delta y^k, s^k + \alpha_k \Delta s^k)$ is feasible.
- ▶ Make step $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k + \alpha_k \Delta x^k, y^k + \alpha_k \Delta y^k, s^k + \alpha_k \Delta s^k)$.

Path-following algorithm

- ▶ **Short-step path-following method:** $\mathcal{O}(\sqrt{n})$ complexity result

Path-following algorithm

- **Short-step path-following method:** $\mathcal{O}(\sqrt{n})$ complexity result

Theorem ([Gondzio, 2012, Thm. 3.1])

Given $\epsilon > 0$, suppose that a feasible starting point $(x^0, y^0, s^0) \in \mathcal{N}_2(0.1)$ satisfies

$$(x^0)^\top s^0 = n\mu^0, \text{ where } \mu^0 \leq 1/\epsilon^\kappa,$$

for some positive constant κ . Then for some $K = \mathcal{O}(\sqrt{n} \ln(1/\epsilon))$ such that

$$\mu^k \leq \epsilon, \quad \forall k \geq K$$

Path-following algorithm

- ▶ **Short-step path-following method:** $\mathcal{O}(\sqrt{n})$ complexity result

Theorem ([Gondzio, 2012, Thm. 3.1])

Given $\epsilon > 0$, suppose that a feasible starting point $(x^0, y^0, s^0) \in \mathcal{N}_2(0.1)$ satisfies

$$(x^0)^\top s^0 = n\mu^0, \text{ where } \mu^0 \leq 1/\epsilon^\kappa,$$

for some positive constant κ . Then for some $K = \mathcal{O}(\sqrt{n} \ln(1/\epsilon))$ such that

$$\mu^k \leq \epsilon, \quad \forall k \geq K$$

- ▶ θ -neighborhood of the central path:

$$\mathcal{N}_2(\theta) := \{(x, y, s) \in \mathcal{F}^0 \mid \|XSe - \mu e\| \leq \theta\mu\}, \text{ with } \mu = \frac{1}{n}x^\top s.$$

Path-following algorithm

- ▶ **Short-step path-following method:** $\mathcal{O}(\sqrt{n})$ complexity result

Theorem ([Gondzio, 2012, Thm. 3.1])

Given $\epsilon > 0$, suppose that a feasible starting point $(x^0, y^0, s^0) \in \mathcal{N}_2(0.1)$ satisfies

$$(x^0)^\top s^0 = n\mu^0, \text{ where } \mu^0 \leq 1/\epsilon^\kappa,$$

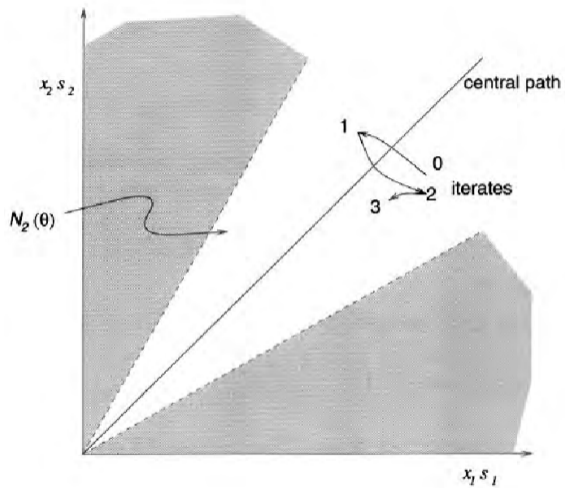
for some positive constant κ . Then for some $K = \mathcal{O}(\sqrt{n} \ln(1/\epsilon))$ such that

$$\mu^k \leq \epsilon, \quad \forall k \geq K$$

- ▶ θ -neighborhood of the central path:

$$\mathcal{N}_2(\theta) := \{(x, y, s) \in \mathcal{F}^0 \mid \|XSe - \mu e\| \leq \theta\mu\}, \text{ with } \mu = \frac{1}{n}x^\top s.$$

- ▶ Slow progress towards optimality



Augmented system

Newton direction

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^\top & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^\top y - s + Qx \\ \mu_k e - XSe \end{bmatrix} =: \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix}$$

use last (complementarity) block to solve for Δs as a function of Δx .

Augmented system

Define $\Theta = XS^{-1}$ (ill-conditioned!). Then Δx and Δy solve the Newton system

\Longleftrightarrow

$$\begin{bmatrix} -Q - \Theta^{-1} & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}$$

- ▶ Newton system is nonsymmetric.
- ▶ Augmented system is symmetric but indefinite.

Normal equations

Augmented system

$$\begin{bmatrix} -\Theta^{-1} & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix} =: \begin{bmatrix} g \\ \xi_p \end{bmatrix}$$

Normal equations

Eliminate Δx to arrive at the *Normal equations*

$$(A\Theta A^\top)\Delta y = A\Theta g + \xi_p$$

Normal equations

Augmented system

$$\begin{bmatrix} -\Theta^{-1} & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix} =: \begin{bmatrix} g \\ \xi_p \end{bmatrix}$$

Normal equations

Eliminate Δx to arrive at the *Normal equations*

$$(A\Theta A^\top)\Delta y = A\Theta g + \xi_p$$

- ▶ $A\Theta A^\top$ is symmetric and positive semidefinite. (Finally!)
- ▶ Normal equations in QP ($A(Q + \Theta)A^\top$) $\Delta y = g$ are generally nearly dense, even when A and Q are sparse.
- ▶ LP: Normal equations are often used.
- ▶ QP: usually use the indefinite augmented system.

Outline

IPM for linear and quadratic programs

IPM for Convex nonlinear programming

IPM for Conic Optimization

IPM for NLP

► Convex NLP

$$\text{minimize } f(x) \quad \text{subject to } g(x) \leq 0$$

$$\text{minimize } f(x) \quad \text{subject to } g(x) + z = 0, z \geq 0$$

IPM for NLP

- Convex NLP

$$\text{minimize } f(x) \quad \text{subject to } g(x) \leq 0$$

$$\text{minimize } f(x) \quad \text{subject to } g(x) + z = 0, z \geq 0$$

- Replace inequality $z \geq 0$ with logarithmic barrier

$$\text{minimize } f(x) - \mu \sum_{i=1}^m \ln(z_i) \quad \text{subject to } g(x) + z = 0$$

IPM for NLP

- Convex NLP

$$\text{minimize } f(x) \quad \text{subject to } g(x) \leq 0$$

$$\text{minimize } f(x) \quad \text{subject to } g(x) + z = 0, z \geq 0$$

- Replace inequality $z \geq 0$ with logarithmic barrier

$$\text{minimize } f(x) - \mu \sum_{i=1}^m \ln(z_i) \quad \text{subject to } g(x) + z = 0$$

- Write out Lagrangian

$$L(x, y, z, \mu) = f(x) + y^\top (g(x) + z) - \mu \sum_{i=1}^m \ln(z_i)$$

IPM for NLP

- Write conditions for stationary point

$$\nabla_x L(x, z, y) = \nabla f(x) + J_g(x)^\top y = 0$$

$$\nabla_y L(x, z, y) = g(x) + z = 0$$

$$\nabla_z L(x, z, y) = y - \mu Z^{-1} e = 0$$

IPM for NLP

- Write conditions for stationary point

$$\nabla_x L(x, z, y) = \nabla f(x) + J_g(x)^\top y = 0$$

$$\nabla_y L(x, z, y) = g(x) + z = 0$$

$$\nabla_z L(x, z, y) = y - \mu Z^{-1} e = 0$$

- Write KKT system

$$\nabla f(x) + J_g(x)^\top y = 0,$$

$$g(x) + z = 0$$

$$YZe = \mu e$$

Newton for KKT of NLP

- ▶ Apply Newton method for KKT system

Newton for KKT of NLP

- ▶ Apply Newton method for KKT system
- ▶ Jacobian matrix of KKT system

$$J_F(x, z, y) = \begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix}$$

where $Q(x, y) = \nabla^2 f(x) + \sum_{i=1}^m y_i \nabla^2 g_i(x)$ is the Hessian of L

Newton for KKT of NLP

- ▶ Apply Newton method for KKT system
- ▶ Jacobian matrix of KKT system

$$J_F(x, z, y) = \begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix}$$

where $Q(x, y) = \nabla^2 f(x) + \sum_{i=1}^m y_i \nabla^2 g_i(x)$ is the Hessian of L

- ▶ Newton step for KKT system

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - z \\ \mu e - YZe \end{bmatrix}$$

From QP to NLP

- Newton direction for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - z \\ \mu e - YZe \end{bmatrix}$$

From QP to NLP

- ▶ Newton direction for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - z \\ \mu e - YZe \end{bmatrix}$$

- ▶ Augmented system for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top \\ J_g(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - \mu Y^{-1} e \end{bmatrix}$$

From QP to NLP

- ▶ Newton direction for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - z \\ \mu e - YZe \end{bmatrix}$$

- ▶ Augmented system for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top \\ J_g(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - \mu Y^{-1} e \end{bmatrix}$$

- ▶ Need to compute $Q(x, y)$ and $J_g(x)$ at each iteration

From QP to NLP

- ▶ Newton direction for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top & 0 \\ J_g(x) & 0 & I \\ 0 & Z & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - z \\ \mu e - YZe \end{bmatrix}$$

- ▶ Augmented system for NLP

$$\begin{bmatrix} Q(x, y) & J_g(x)^\top \\ J_g(x) & -ZY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) - J_g(x)^\top y \\ -g(x) - \mu Y^{-1} e \end{bmatrix}$$

- ▶ Need to compute $Q(x, y)$ and $J_g(x)$ at each iteration
- ▶ Caveat: use trust region method to choose stepsize.

Outline

IPM for linear and quadratic programs

IPM for Convex nonlinear programming

IPM for Conic Optimization

Self-concordant function

Definition

Function f is *self-concordant* if for some constant $M_f \geq 0$, the inequality

$$\nabla^3 f(x)[u, u, u] \leq M_f \|u\|_{\nabla^2 f(x)}^{3/2}$$

holds for any $x \in \text{dom } f$ and $u \in R^n$.

- ▶ A self-concordant function is always well approximated by a quadratic model because the error of such an approximation can be bounded by the $\|u\|_{\nabla^2 f(x)}^{3/2}$

Self-concordant function

Definition

Function f is *self-concordant* if for some constant $M_f \geq 0$, the inequality

$$\nabla^3 f(x)[u, u, u] \leq M_f \|u\|_{\nabla^2 f(x)}^{3/2}$$

holds for any $x \in \text{dom } f$ and $u \in R^n$.

- ▶ A self-concordant function is always well approximated by a quadratic model because the error of such an approximation can be bounded by the $\|u\|_{\nabla^2 f(x)}^{3/2}$

Self-concordant function

Definition

Function f is *self-concordant* if for some constant $M_f \geq 0$, the inequality

$$\nabla^3 f(x)[u, u, u] \leq M_f \|u\|_{\nabla^2 f(x)}^{3/2}$$

holds for any $x \in \text{dom } f$ and $u \in R^n$.

- ▶ A self-concordant function is always well approximated by a quadratic model because the error of such an approximation can be bounded by the $\|u\|_{\nabla^2 f(x)}^{3/2}$

Theorem ([Boyd and Vandenberghe, 2004, Section 11.5])

Newton's method with line search finds an ε approximate solution in less than $T := \text{constant} \times (f(x_0) - f^*) + \log_2 \log_2 \frac{1}{\varepsilon}$ iterations.

Log-barrier is self-concordant

Theorem

The barrier function $-\ln(x)$ is self-concordant in \mathbf{R}_+ .

Log-barrier is self-concordant

Theorem

The barrier function $-\ln(x)$ is self-concordant in \mathbf{R}_+ .

Proof.

Consider $f(x) = -\ln(x)$, then

$$f'(x) = -\frac{1}{x}, \quad f''(x) = \frac{1}{x^2}, \quad f'''(x) = -\frac{2}{x^3}$$

Log-barrier is self-concordant

Theorem

The barrier function $-\ln(x)$ is self-concordant in \mathbf{R}_+ .

Proof.

Consider $f(x) = -\ln(x)$, then

$$f'(x) = -\frac{1}{x}, \quad f''(x) = \frac{1}{x^2}, \quad f'''(x) = -\frac{2}{x^3}$$

Compute and check that self-concordance holds with $M_f = 2$. □

► $-\ln(1/x^\alpha)$, with $\alpha \in (0, \infty)$ is not self-concordant in \mathbf{R}_+ .

Log-barrier is self-concordant

Theorem

The barrier function $-\ln(x)$ is self-concordant in \mathbf{R}_+ .

Proof.

Consider $f(x) = -\ln(x)$, then

$$f'(x) = -\frac{1}{x}, \quad f''(x) = \frac{1}{x^2}, \quad f'''(x) = -\frac{2}{x^3}$$

Compute and check that self-concordance holds with $M_f = 2$. □

- ▶ $-\ln(1/x^\alpha)$, with $\alpha \in (0, \infty)$ is not self-concordant in \mathbf{R}_+ .
- ▶ $\exp(1/x)$ is not self-concordant in \mathbf{R}_+ .

Conic optimization

- Consider the optimization problem

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

where K is a convex closed cone.

Conic optimization

- Consider the optimization problem

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

where K is a convex closed cone.

- The associated dual is

$$\begin{array}{ll}\text{maximize} & b^\top y \\ \text{subject to} & A^\top y + s = c \\ & x \in K^* \text{ (Dual cone)}\end{array}$$

Conic optimization

- Consider the optimization problem

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

where K is a convex closed cone.

- The associated dual is

$$\begin{array}{ll}\text{maximize} & b^\top y \\ \text{subject to} & A^\top y + s = c \\ & x \in K^* \text{ (Dual cone)}\end{array}$$

- Weak duality

$$c^\top x - b^\top y = x^\top (c - A^\top y) = x^\top s \geq 0$$

Conic optimization

- Consider the optimization problem

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

where K is a convex closed cone.

- The associated dual is

$$\begin{array}{ll}\text{maximize} & b^\top y \\ \text{subject to} & A^\top y + s = c \\ & x \in K^* \text{ (Dual cone)}\end{array}$$

- Weak duality

$$c^\top x - b^\top y = x^\top (c - A^\top y) = x^\top s \geq 0$$

- Conic optimization can be solved in polynomial time with IPMs

Second-order conic optimization

- ▶ $K = \mathbb{L} := \{(x, t) \mid x \in \mathbf{R}^{n-1}, t \in \mathbf{R}, \|x\|_2 \leq t, t \geq 0\}$ (Lorenz or second-order cone)

Second-order conic optimization

- ▶ $K = \mathbb{L} := \{(x, t) \mid x \in \mathbf{R}^{n-1}, t \in \mathbf{R}, \|x\|_2 \leq t, t \geq 0\}$ (Lorenz or second-order cone)
- ▶ Logarithmic barrier function for the second-order cone

$$f(x, t) = \begin{cases} -\ln(t^2 - \|x\|_2^2) & \text{if } \|x\| < t \\ +\infty & \text{otherwise} \end{cases}$$

Second-order conic optimization

- ▶ $K = \mathbb{L} := \{(x, t) \mid x \in \mathbf{R}^{n-1}, t \in \mathbf{R}, \|x\|_2 \leq t, t \geq 0\}$ (Lorenz or second-order cone)
- ▶ Logarithmic barrier function for the second-order cone

$$f(x, t) = \begin{cases} -\ln(t^2 - \|x\|_2^2) & \text{if } \|x\| < t \\ +\infty & \text{otherwise} \end{cases}$$

Second-order conic optimization

- ▶ $K = \mathbb{L} := \{(x, t) \mid x \in \mathbf{R}^{n-1}, t \in \mathbf{R}, \|x\|_2 \leq t, t \geq 0\}$ (Lorenz or second-order cone)
- ▶ Logarithmic barrier function for the second-order cone

$$f(x, t) = \begin{cases} -\ln(t^2 - \|x\|_2^2) & \text{if } \|x\| < t \\ +\infty & \text{otherwise} \end{cases}$$

Theorem

The barrier function $f(x, t)$ is self-concordant on \mathbb{L} .

Second-order conic optimization

- ▶ $K = \mathbb{L} := \{(x, t) \mid x \in \mathbf{R}^{n-1}, t \in \mathbf{R}, \|x\|_2 \leq t, t \geq 0\}$ (Lorenz or second-order cone)
- ▶ Logarithmic barrier function for the second-order cone

$$f(x, t) = \begin{cases} -\ln(t^2 - \|x\|_2^2) & \text{if } \|x\| < t \\ +\infty & \text{otherwise} \end{cases}$$

Theorem

The barrier function $f(x, t)$ is self-concordant on \mathbb{L} .

Exercise: Prove in case $n = 2$.

Semidefinite programming

- ▶ Variable now is a symmetric matrix $X \in K = \mathbf{S}^n$

SDP and its dual

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

$$\begin{array}{ll}\text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m y_i A_i + S = C \\ & S \succeq 0\end{array}$$

$A_i, C \in \mathbf{S}^n$ and $b \in \mathbf{R}^m$ given, and $X, S \in \mathbf{S}^n$ and $y \in \mathbf{R}^m$ unknown.

Semidefinite programming

- ▶ Variable now is a symmetric matrix $X \in K = \mathbf{S}^n$
- ▶ Define $X \bullet Y = \text{tr}(X^\top Y)$

SDP and its dual

minimize $C \bullet X$
subject to $A_i \bullet X = b_i, i = 1, \dots, m$
 $X \succeq 0$

maximize $b^\top y$
subject to $\sum_{i=1}^m y_i A_i + S = C$
 $S \succeq 0$

$A_i, C \in \mathbf{S}^n$ and $b \in \mathbf{R}^m$ given, and $X, S \in \mathbf{S}^n$ and $y \in \mathbf{R}^m$ unknown.

Semidefinite programming

- ▶ Variable now is a symmetric matrix $X \in K = \mathbf{S}^n$
- ▶ Define $X \bullet Y = \text{tr}(X^\top Y)$

SDP and its dual

minimize $C \bullet X$
subject to $A_i \bullet X = b_i, i = 1, \dots, m$
 $X \succeq 0$

maximize $b^\top y$
subject to $\sum_{i=1}^m y_i A_i + S = C$
 $S \succeq 0$

$A_i, C \in \mathbf{S}^n$ and $b \in \mathbf{R}^m$ given, and $X, S \in \mathbf{S}^n$ and $y \in \mathbf{R}^m$ unknown.

Semidefinite programming

- ▶ Variable now is a symmetric matrix $X \in K = \mathbf{S}^n$
- ▶ Define $X \bullet Y = \text{tr}(X^\top Y)$

SDP and its dual

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

$$\begin{array}{ll}\text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m y_i A_i + S = C \\ & S \succeq 0\end{array}$$

$A_i, C \in \mathbf{S}^n$ and $b \in \mathbf{R}^m$ given, and $X, S \in \mathbf{S}^n$ and $y \in \mathbf{R}^m$ unknown.

Theorem (Weak duality for SDP)

If X is primal feasible and (y, S) is dual feasible, then

$$C \bullet X - b^\top y = X \bullet S \geq 0$$

Logarithmic barrier for SDP

- Logarithmic barrier function for the semi-definite cone

$$f(X) = \begin{cases} -\ln(\det(X)) & \text{if } X \succ 0 \\ +\infty & \text{otherwise} \end{cases}$$

Logarithmic barrier for SDP

- ▶ Logarithmic barrier function for the semi-definite cone

$$f(X) = \begin{cases} -\ln(\det(X)) & \text{if } X \succ 0 \\ +\infty & \text{otherwise} \end{cases}$$

- ▶ Facts (for small t):

Logarithmic barrier for SDP

- ▶ Logarithmic barrier function for the semi-definite cone

$$f(X) = \begin{cases} -\ln(\det(X)) & \text{if } X \succ 0 \\ +\infty & \text{otherwise} \end{cases}$$

- ▶ Facts (for small t):
 - ▶ $\det(I + tU) = 1 + t \operatorname{tr}(U) + \mathcal{O}(t^2)$

Logarithmic barrier for SDP

- ▶ Logarithmic barrier function for the semi-definite cone

$$f(X) = \begin{cases} -\ln(\det(X)) & \text{if } X \succ 0 \\ +\infty & \text{otherwise} \end{cases}$$

- ▶ Facts (for small t):
 - ▶ $\det(I + tU) = 1 + t \operatorname{tr}(U) + \mathcal{O}(t^2)$
 - ▶ $\ln(1 + t \operatorname{tr}(U)) \approx t \operatorname{tr}(U)$

Logarithmic barrier for SDP

- ▶ Logarithmic barrier function for the semi-definite cone

$$f(X) = \begin{cases} -\ln(\det(X)) & \text{if } X \succ 0 \\ +\infty & \text{otherwise} \end{cases}$$

- ▶ Facts (for small t):
 - ▶ $\det(I + tU) = 1 + t \operatorname{tr}(U) + \mathcal{O}(t^2)$
 - ▶ $\ln(1 + t \operatorname{tr}(U)) \approx t \operatorname{tr}(U)$
- ▶ Let $X \succ 0$ and $H \in \mathbf{S}^n$ be given. Then

$$\begin{aligned} f(X + tH) &= -\ln(\det(X + tH)) = -\ln(\det(X(I + tX^{-1}H))) \\ &= -\ln(\det(X)) - \ln(\det(I + tX^{-1}H)) \\ &= -\ln(\det(X)) - \ln(1 + t \operatorname{tr}(X^{-1}H) + \mathcal{O}(t^2)) \\ &= f(X) - tX^{-1} \bullet H + \mathcal{O}(t^2) \end{aligned}$$

Derivatives of Logarithmic barrier for SDP

- First derivative of $f(X)$

$$f'(X) = \lim_{t \rightarrow 0} \frac{f(X + tH) - f(X)}{t} = -X^{-1}$$

Derivatives of Logarithmic barrier for SDP

- First derivative of $f(X)$

$$f'(X) = \lim_{t \rightarrow 0} \frac{f(X + tH) - f(X)}{t} = -X^{-1}$$

Derivatives of Logarithmic barrier for SDP

- First derivative of $f(X)$

$$f'(X) = \lim_{t \rightarrow 0} \frac{f(X + tH) - f(X)}{t} = -X^{-1}$$

So $Df(X)[H] = -X^{-1} \bullet H$.

Derivatives of Logarithmic barrier for SDP

- First derivative of $f(X)$

$$f'(X) = \lim_{t \rightarrow 0} \frac{f(X + tH) - f(X)}{t} = -X^{-1}$$

So $Df(X)[H] = -X^{-1} \bullet H$.

- Second derivative of $f(X)$

$$\begin{aligned} f'(X + tH) &= -[X(I + tX^{-1}H)]^{-1} = -[I - tX^{-1}H + \mathcal{O}(t^2)]X^{-1} \\ &= f'(X) + tX^{-1}HX^{-1} + \mathcal{O}(t^2) \end{aligned}$$

Derivatives of Logarithmic barrier for SDP

- First derivative of $f(X)$

$$f'(X) = \lim_{t \rightarrow 0} \frac{f(X + tH) - f(X)}{t} = -X^{-1}$$

So $Df(X)[H] = -X^{-1} \bullet H$.

- Second derivative of $f(X)$

$$\begin{aligned} f'(X + tH) &= -[X(I + tX^{-1}H)]^{-1} = -[I - tX^{-1}H + \mathcal{O}(t^2)]X^{-1} \\ &= f'(X) + tX^{-1}HX^{-1} + \mathcal{O}(t^2) \end{aligned}$$

Derivatives of Logarithmic barrier for SDP

- First derivative of $f(X)$

$$f'(X) = \lim_{t \rightarrow 0} \frac{f(X + tH) - f(X)}{t} = -X^{-1}$$

So $Df(X)[H] = -X^{-1} \bullet H$.

- Second derivative of $f(X)$

$$\begin{aligned} f'(X + tH) &= -[X(I + tX^{-1}H)]^{-1} = -[I - tX^{-1}H + \mathcal{O}(t^2)]X^{-1} \\ &= f'(X) + tX^{-1}HX^{-1} + \mathcal{O}(t^2) \end{aligned}$$

so $f''(X)[H] = X^{-1}HX^{-1}$ and $D^2f(X)[H, G] = X^{-1}HX^{-1} \bullet G$.

- $f'''(X)[H, G] = -X^{-1}HX^{-1}GX^{-1} - X^{-1}GX^{-1}HX^{-1}$

Characterization of self-concordance for SDP

Theorem

The function $f(X) = -\ln \det X$ is a convex barrier for \mathbf{S}_+^n .

Characterization of self-concordance for SDP

Theorem

The function $f(X) = -\ln \det X$ is a convex barrier for \mathbf{S}_+^n .

Proof sketch.

Let $\varphi(t) = F(X + tH)$. Then, prove that $\varphi''(t) \geq 0$ for $t > 0$ such that $X + tH \succ 0$. Therefore, when $X \succ 0$ approaches a singular matrix, its determinant approaches zero, and the function $f(X) \rightarrow +\infty$. □

Characterization of self-concordance for SDP

Theorem

The function $f(X) = -\ln \det X$ is a convex barrier for \mathbf{S}_+^n .

Proof sketch.

Let $\varphi(t) = F(X + tH)$. Then, prove that $\varphi''(t) \geq 0$ for $t > 0$ such that $X + tH \succ 0$. Therefore, when $X \succ 0$ approaches a singular matrix, its determinant approaches zero, and the function $f(X) \rightarrow +\infty$. □

Theorem ([Nestervov and Nemirovskii, 1994])

The barrier function $f(X) = -\ln \det X$ is self-concordant on \mathbf{S}_+^n .

Solving SDPs with IPMs

- Replace the primal SDP

$$\begin{aligned} &\text{minimize} && C \bullet X \\ &\text{subject to} && \mathcal{A}X = b, \\ &&& X \succeq 0, \end{aligned}$$

with the primal barrier SDP

$$\begin{aligned} &\text{minimize} && C \bullet X + \mu f(X) \\ &\text{subject to} && \mathcal{A}X = b, \end{aligned}$$

(with a barrier parameter $\mu \geq 0$).

Solving SDPs with IPMs

- Replace the primal SDP

$$\begin{aligned} & \text{minimize} && C \bullet X \\ & \text{subject to} && \mathcal{A}X = b, \\ & && X \succeq 0, \end{aligned}$$

with the primal barrier SDP

$$\begin{aligned} & \text{minimize} && C \bullet X + \mu f(X) \\ & \text{subject to} && \mathcal{A}X = b, \end{aligned}$$

(with a barrier parameter $\mu \geq 0$).

- Formulate the Lagrangian

$$L(X, y, S) = C \bullet X + \mu f(X) - y^T (\mathcal{A}X - b),$$

with $y \in \mathcal{R}^m$, and write the first order conditions (FOC) for a stationary point of L :

$$C + \mu f'(X) - \mathcal{A}^* y = 0$$

Solving SDPs with IPMs (cont'd)

- Use $f(X) = -\ln \det X$ and $f'(X) = -X^{-1}$ to obtain

$$C - \mu X^{-1} - \mathcal{A}^* y = 0$$

Solving SDPs with IPMs (cont'd)

- ▶ Use $f(X) = -\ln \det X$ and $f'(X) = -X^{-1}$ to obtain

$$C - \mu X^{-1} - \mathcal{A}^* y = 0$$

- ▶ Denote $S = \mu X^{-1}$, i.e., $XS = \mu I$. Then, the FOC can be written as

$$\mathcal{A}X = b$$

$$\mathcal{A}^* y + S = C$$

$$XS = \mu I$$

with $X, S \in \mathbf{S}_{++}^n$.

Newton direction

Differentiating this system is hard! The Newton direction solves:

$$\begin{bmatrix} \mathcal{A} & 0 & 0 \\ 0 & \mathcal{A}^* & \mathcal{I} \\ \mu(X^{-1} \odot X^{-1}) & 0 & \mathcal{I} \end{bmatrix} \cdot \begin{bmatrix} \Delta X \\ \Delta y \\ \Delta S \end{bmatrix} = \begin{bmatrix} \xi_b \\ \xi_c \\ \xi_\mu \end{bmatrix}.$$

We introduce a useful notation $P \odot Q$ for $n \times n$ matrices P and Q is the Kronecker product. This defines a linear operator from \mathbf{S}^n to \mathbf{S}^n given by

$$(P \odot Q)U = \frac{1}{2} (PUQ^T + QUP^T).$$

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:
 - ▶ polynomial complexity (predictable behaviour)

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:
 - ▶ polynomial complexity (predictable behaviour)
- ▶ Unified algorithm with fast convergence

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:
 - ▶ polynomial complexity (predictable behaviour)
- ▶ Unified algorithm with fast convergence
 - ▶ from LP via QP to NLP, SOCP and SDP

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:
 - ▶ polynomial complexity (predictable behaviour)
- ▶ Unified algorithm with fast convergence
 - ▶ from LP via QP to NLP, SOCP and SDP
- ▶ efficient for LP, QP, SOCP

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:
 - ▶ polynomial complexity (predictable behaviour)
- ▶ Unified algorithm with fast convergence
 - ▶ from LP via QP to NLP, SOCP and SDP
- ▶ efficient for LP, QP, SOCP
- ▶ problematic for SDP because solving a problem of size n involves linear algebra operations in dimension n^2

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]

Summary

- ▶ IPM for SOCP and SDP with self-concordant barrier:
 - ▶ polynomial complexity (predictable behaviour)
- ▶ Unified algorithm with fast convergence
 - ▶ from LP via QP to NLP, SOCP and SDP
- ▶ efficient for LP, QP, SOCP
- ▶ problematic for SDP because solving a problem of size n involves linear algebra operations in dimension n^2
 - ▶ and this requires n^6 flops!

source: [Gondzio, 2012: Interior Point Methods 25 Years Later]