10. Interior-point methods for linear optimization
Midterm

• **Date/Time:** Thursday October 13, 11:00am — 12:20pm (80 min)

• **Where:** in class

• **Material allowed:** single sheet of paper, double-sided, hand-written or typed.

• **Topics:** linear optimization
Since introducing a new constraint to a primal problem is equivalent to introducing a new variable to the dual, is there a difference between choosing to solve one problem rather than another?

- How efficient is differentiable optimization? In neural network, people use ReLU because it is simple in evaluating both the forward pass values and backward differentiations. However, it seems like we need to solve a LP in every node in forward pass and to solve a linear system in backward pass.

- In lecture 9, we have seen that if we perturb the constraints a bit by some vector $u$, the optimal value behaves as a piecewise linear function in $u$. In case $c^T x$ and $AX$ are replaced by any suitable convex functions in $x$, is such a sensitivity analysis still possible?
Recap
Adding new variables

minimize \[ c^T x \]
subject to \[ Ax = b \]
\[ x \geq 0 \]

Solution \( x^*, y^* \)
Adding new variables

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b, \quad x \geq 0 \\
\text{Solution} & \quad x^*, y^*
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad c^T x + c_{n+1} x_{n+1} \\
\text{subject to} & \quad Ax + A_{n+1} x_{n+1} = b, \quad x, x_{n+1} \geq 0
\end{align*}
\]

Adding new variables

minimize \( c^T x \)
subject to \( Ax = b \)
\( x \geq 0 \)

Solution \( x^*, y^* \)

minimize \( c^T x + c_{n+1} x_{n+1} \)
subject to \( Ax + A_{n+1} x_{n+1} = b \)
\( x, x_{n+1} \geq 0 \)

Solution \( (x^*, 0), y^* \) optimal for the new problem?
Adding new variables
Optimality conditions

minimize \[ c^T x + c_{n+1} x_{n+1} \]
subject to \[ Ax + A_{n+1} x_{n+1} = b \]
\[ x, x_{n+1} \geq 0 \]

\[ \longrightarrow \] Solution \((x^*, 0)\) is still **primal feasible**
Adding new variables
Optimality conditions

minimize \( c^T x + c_{n+1} x_{n+1} \)
subject to \( Ax + A_{n+1} x_{n+1} = b \) \( x, x_{n+1} \geq 0 \)

Solution \((x^*, 0)\) is still \textbf{primal feasible}

Is \( y^* \) still \textbf{dual feasible}?

\( A_{n+1}^T y^* + c_{n+1} \geq 0 \)
Adding new variables

Optimality conditions

minimize $c^T x + c_{n+1} x_{n+1}$
subject to $A x + A_{n+1} x_{n+1} = b$
$x, x_{n+1} \geq 0$

Solution $(x^*, 0)$ is still **primal feasible**

Is $y^*$ still **dual feasible**?

$$A_{n+1}^T y^* + c_{n+1} \geq 0$$

**Yes**
$(x^*, 0)$ still **optimal** for new problem

**Otherwise**
Primal simplex
Adding new constraints

minimize $c^T x$
subject to $Ax = b$
$x \geq 0$

Solution $x^*, y^*$
Adding new constraints

minimize \( c^T x \)  \quad \text{subject to} \quad \begin{align*} Ax &= b \\ x &\geq 0 \end{align*} \quad \text{Solution} \ x^*, y^*

\rightarrow

minimize \( c^T x \)  \quad \text{subject to} \quad \begin{align*} Ax &= b \\ a_{m+1}^T x &= b_{m+1} \\ x &\geq 0 \end{align*}
Adding new constraints

minimize \( c^T x \)

subject to \( Ax = b \)
\( x \geq 0 \)

Solution \( x^*, y^* \)

minimize \( c^T x \)

subject to \( Ax = b \)
\( a_{m+1}^T x = b_{m+1} \)
\( x \geq 0 \)

Dual

maximize \( -b^T y - b_{m+1} y_{m+1} \)

subject to \( A^T y + a_{m+1} y_{m+1} + c \geq 0 \)
Adding new constraints

minimize \[ c^T x \]
subject to \[ A x = b \]
\[ x \geq 0 \]

Solution \( x^*, y^* \)

minimize \[ c^T x \]
subject to \[ A x = b \]
\[ a_{m+1}^T x = b_{m+1} \]
\[ x \geq 0 \]

Dual

maximize \[ -b^T y \]
subject to \[ A^T y + a_{m+1} y_{m+1} + c \geq 0 \]

Solution \( x^*, (y^*, 0) \) optimal for the new problem?
Adding new constraints
Optimality conditions

maximize \(-b^T y\)
subject to \(A^T y + a_{m+1}y_{m+1} + c \geq 0\)  \(\rightarrow\) Solution \((y^*, 0)\) is still dual feasible
Adding new constraints
Optimality conditions

maximize $-b^T y$
subject to $A^T y + a_{m+1} y_{m+1} + c \geq 0$

Solution $(y^*, 0)$ is still dual feasible

Is $x^*$ still primal feasible?

$A x = b$
$a_{m+1}^T x = b_{m+1}$
$x \geq 0$
Adding new constraints
Optimality conditions

maximize \(-b^T y\)
subject to \(A^T y + a_{m+1}y_{m+1} + c \geq 0\)  \(\rightarrow\) Solution \((y^*, 0)\) is still dual feasible

Is \(x^*\) still primal feasible?
\[Ax = b\]
\[a^T_{m+1}x = b_{m+1}\]
\[x \geq 0\]

Yes
\(x^*\) still optimal for new problem

Otherwise
Dual simplex
Today’s lecture

[Chapter 14, NO][Chapters 17/18, LP]

- History
- Newton’s method
- Central path
- Primal-dual path following method
- Logarithmic barrier functions
- Convergence
History
Ellipsoid method
Khachian (1979)

Answer to major question
Is worst-case LP complexity polynomial? Yes!

Shazam! A Shortcut for Computers

—Jonathan Friendly

Published: November 11, 1979
Copyright © The New York Times
Ellipsoid method
Khachian (1979)

Answer to major question
Is worst-case LP complexity polynomial? Yes!

Drawbacks
Very inefficient. Much slower than simplex!
Ellipsoid method
Khachian (1979)

Answer to major question
Is worst-case LP complexity polynomial? Yes!

Drawbacks
Very inefficient. Much slower than simplex!

Benefits
Motivated new research directions
Interior-point methods

1950s-1960s: nonlinear convex optimization

• Sequential unconstrained optimization (Fiacco & McCormick), Logarithmic barrier method (Frish), affine scaling method (Dikin), etc.

• No worst-case complexity theory but often good practical performance
Interior-point methods

1950s-1960s: nonlinear convex optimization
- Sequential unconstrained optimization (Fiacco & McCormick), Logarithmic barrier method (Frish), affine scaling method (Dikin), etc.
- No worst-case complexity theory but often good practical performance

1980s-1990s: interior point methods
- Karmarkar’s algorithm (1984)
- Competitive with simplex, often faster for larger problems
Newton’s method
Newton’s method for nonlinear equations

**Goal:** solve

\[ h(x) = 0 \]

**Derivative**

\[
Dh = \begin{bmatrix}
\frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n}
\end{bmatrix}
\]
Newton’s method for nonlinear equations

Goal: solve
\[ h(x) = 0 \]

First-order approximation
\[ h(x) \approx h(\bar{x}) + Dh(\bar{x})(x - \bar{x}) \]

Derivative
\[ Dh = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix} \]

Iteratively set to zero
\[ h(x^k) + Dh(x^k)(x^{k+1} - x^k) = 0 \]
\[ \Delta x \]
Newton’s method for nonlinear equations

Goal: solve 
\[ h(x) = 0 \]

Derivative 
\[ Dh = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix} \]

First-order approximation 
\[ h(x) \approx h(\bar{x}) + Dh(\bar{x})(x - \bar{x}) \]

Iteratively set to zero 
\[ h(x^k) + Dh(x^k)(x^{k+1} - x^k) = 0 \]

\[ \Delta x \]

Iterations
- Solve \[ Dh(x^k) \Delta x = -h(x^k) \]
- \[ x^{k+1} \leftarrow x^k + \Delta x \]
Newton method

Convergence

**Iterations**
- Solve $Dh(x^k) \Delta x = -h(x^k)$
- $x^{k+1} \leftarrow x^k + \Delta x$

**Remarks**
- Iterations can be expensive (linear system solution)
- **Fast (quadratic) convergence** close to the solution $x^*$
Optimality conditions

minimize \[ c^T x \]
subject to \[ Ax \leq b \]
Optimality conditions

minimize \( c^T x \)    \[ \rightarrow \]    minimize \( c^T x \)
subject to \( Ax \leq b \)    subject to \( Ax + s = b \)
                \( s \geq 0 \)    subject to \( A^T y + c = 0 \)
                \( y \geq 0 \)

Primal

Dual
Optimality conditions

**Primal**

minimize $c^T x$

subject to $Ax \leq b$

minimize $c^T x$

subject to $Ax + s = b$

subject to $s \geq 0$

**Dual**

maximize $-b^T y$

subject to $A^T y + c = 0$

subject to $y \geq 0$

**Optimality conditions**

$Ax + s - b = 0$ \hspace{1cm} **Primal Feas**

$A^T y + c = 0$ \hspace{1cm} **Dual Feas**

$s_i y_i = 0$ \hspace{1cm} **Compl Slack**

$s, y \geq 0$
Main idea

Optimality conditions

\[ h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY_1 \end{bmatrix} = 0 \]

\[ S = \text{diag}(s) \]

\[ Y = \text{diag}(y) \]

\[ s, y \geq 0 \]

- Apply variants of Newton’s method to solve \( h(x, s, y) = 0 \)
- Enforce \( s, y > 0 \) (strictly) at every iteration
- **Motivation** avoid getting stuck in “corners”
Newton’s method for optimality conditions

Root-finding equation

\[ h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY1 \end{bmatrix} = 0 \]

Linear system

\[
\begin{bmatrix}
0 & A & I \\
A^T & 0 & 0 \\
S & 0 & Y
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta x \\
\Delta s
\end{bmatrix}
\begin{bmatrix}
-h \\
-r_p \\
-r_d \\
-SY1
\end{bmatrix}
\]

Residuals

\[ r_p = Ax + s - b \]
\[ r_d = A^T y + c \]
Newton’s method for optimality conditions

Root-finding equation

\[ h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY1 \end{bmatrix} = 0 \]

Linear system

\[
\begin{bmatrix}
0 & A & I \\
A^T & 0 & 0 \\
S & 0 & Y
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta x \\
\Delta s
\end{bmatrix}
= \begin{bmatrix}
-h \\
-r_p \\
-r_d
\end{bmatrix}
\]

Residuals

\[
\begin{align*}
r_p &= Ax + s - b \\
r_d &= A^T y + c
\end{align*}
\]

Line search to enforce \( s, y > 0 \)

\( (x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y) \)
Newton’s method for optimality conditions

Root-finding equation

\[ h(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY1 \end{bmatrix} = 0 \]

Linear system

\[
\begin{bmatrix}
Dh \\
\begin{bmatrix}
0 & A & I \\
A^T & 0 & 0 \\
S & 0 & Y
\end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta x \\
\Delta s
\end{bmatrix}
\begin{bmatrix}
-h \\
r_p \\
r_d \end{bmatrix}
= \begin{bmatrix}
-r_p \\
r_d \\
-SY1
\end{bmatrix}
\]

Residuals

\[ r_p = Ax + s - b \]
\[ r_d = A^T y + c \]

Issue

Newton’s step does not allow significant progress towards both

\[ h(x, s, y) = 0 \text{ and } s, y \geq 0. \]

Line search to enforce \( s, y > 0 \)

\[ (x, s, y) \leftarrow (x, s, y) + \alpha (\Delta x, \Delta s, \Delta y) \]
Central path
Smoothed optimality conditions

Optimality conditions

\[ Ax + s - b = 0 \]
\[ A^T y + c = 0 \]
\[ s_i y_i = \tau \]
\[ s, y \geq 0 \]

Same optimality conditions for a “smoothed” version of our problem
Smoothed optimality conditions

Optimality conditions

\[ \begin{align*}
Ax + s - b &= 0 \\
A^T y + c &= 0 \\
\forall i, y_i &= \frac{\tau}{s_i} \\
s, y &\geq 0
\end{align*} \]

Same optimality conditions for a "smoothed" version of our problem

Duality gap

\[ s^T y = (b - Ax)^T y = b^T y - x^T A^T y = b^T y + c^T x \]

Same \( \tau \) for every pair
Newton’s method for smoothed optimality conditions

Smoothed optimality conditions

\[ h_\tau(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY 1 - \tau 1 \end{bmatrix} = 0 \]

\[ s, y \geq 0 \]
Newton’s method for smoothed optimality conditions

Smoothed optimality conditions

\[ h_{\tau}(x, s, y) = \begin{bmatrix} Ax + s - b \\ A^T y + c \\ SY 1 - \tau 1 \end{bmatrix} = 0 \]

\[ s, y \geq 0 \]

Linear system

\[
\begin{bmatrix}
0 & A & I \\
A^T & 0 & 0 \\
S & 0 & Y \\
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta x \\
\Delta s \\
\end{bmatrix}
= 
\begin{bmatrix}
-r_p \\
-r_d \\
-SY + \tau 1 \\
\end{bmatrix}
\]

**Line search** to enforce \( s, y > 0 \)

\( (x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y) \)
The path parameter

Duality measure

\[ \mu = \frac{s^T y}{m} \]  
(average value of the pairs \( s_i y_i \))
The path parameter

Duality measure

\[ \mu = \frac{s^T y}{m} \]  
(average value of the pairs \( s_i y_i \))

Linear system

\[
\begin{bmatrix}
0 & A & I \\
A^T & 0 & 0 \\
S & 0 & Y
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta x \\
\Delta s
\end{bmatrix}
= 
\begin{bmatrix}
-r_p \\
-r_d \\
-SY1 + \sigma \mu 1
\end{bmatrix}
\]
The path parameter

Duality measure

\[ \mu = \frac{s^T y}{m} \] (average value of the pairs \( s_i y_i \))

Centering parameter

\[ \sigma \in [0, 1] \]

Linear system

\[
\begin{bmatrix}
0 & A & I \\
A^T & 0 & 0 \\
S & 0 & Y
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta x \\
\Delta s
\end{bmatrix}
= \begin{bmatrix}
-r_p \\
-r_d \\
-SY1 + \sigma \mu 1
\end{bmatrix}
\]
The path parameter

Duality measure
$$\mu = \frac{s^T y}{m} \quad \text{(average value of the pairs } s_i y_i \text{)}$$

Linear system
$$\begin{bmatrix} 0 & A & I \\ AT & 0 & 0 \\ S & 0 & Y \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_p \\ -r_d \\ -SY 1 + \sigma \mu 1 \end{bmatrix}$$

Centering parameter
$$\begin{align*}
\sigma = 0 & \implies \text{Newton step} \\
\sigma \in [0, 1] & \quad \sigma = 1 & \implies \text{Centering step towards } (y^*(\mu), x^*(\mu), s^*(\mu))
\end{align*}$$
The path parameter

Duality measure

\[ \mu = \frac{s^T y}{m} \] (average value of the pairs \( s_i y_i \))

Centering parameter

\( \sigma \in [0, 1] \)

\( \sigma = 0 \quad \Rightarrow \quad \text{Newton step} \)

\( \sigma = 1 \quad \Rightarrow \quad \text{Centering step towards } (y^*(\mu), x^*(\mu), s^*(\mu)) \)

Line search to enforce \( s, y > 0 \)

\[ (y, x, s) \leftarrow (y, x, s) + \alpha(\Delta y, \Delta x, \Delta s) \]
The central path

\[ s_{2y_2} \]

\[ \sigma = 0 \]

\[ \sigma' = 1 \]

\[ \sigma = 0 \]

CENTRAL PATH

GOAL
Primal-dual path-following method
Path-following algorithm idea

Newton step \( \sigma = 0 \)

Centering step \( \sigma = 1 \)

Combined step

\( x^* \)
Path-following algorithm idea

Newton step
$\sigma = 0$

Centering step
$\sigma = 1$

Combined step
$x^*$

Centering step
It brings towards the central path and is usually biased towards $s, y > 0$.

No progress on duality measure $\mu$
Path-following algorithm idea

Newton step
\[ \sigma = 0 \]

Centering step
\[ \sigma = 1 \]

Combined step

Centering step
It brings towards the **central path** and is usually biased towards \( s, y > 0 \).
**No progress** on duality measure \( \mu \)

Newton step
It brings towards the **zero duality measure** \( \mu \). Quickly violates \( s, y > 0 \).
Path-following algorithm idea

Centering step
It brings towards the central path and is usually biased towards $s, y > 0$. **No progress** on duality measure $\mu$.

Newton step
It brings towards the zero duality measure $\mu$. Quickly violates $s, y > 0$.

Combined step
Best of both worlds with longer steps
Primal-dual path-following algorithm

Initialization
1. Given \((x_0, s_0, y_0)\) such that \(s_0, y_0 > 0\)

Iterations
1. Choose \(\sigma \in [0, 1]\)
2. Solve
   \[
   \begin{bmatrix}
   0 & A & I \\
   A^T & 0 & 0 \\
   S & 0 & Y \\
   \end{bmatrix}
   \begin{bmatrix}
   \Delta y \\
   \Delta x \\
   \Delta s \\
   \end{bmatrix}
   =
   \begin{bmatrix}
   -r_p \\
   -r_d \\
   -SY1 + \sigma\mu1 \\
   \end{bmatrix}
   \]
   where \(\mu = s^Ty/m\)
3. Find maximum \(\alpha\) such that \(y + \alpha\Delta y > 0\) and \(s + \alpha\Delta s > 0\)
4. Update \((x, s, y) \leftarrow (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y)\)
Working towards optimality conditions

Optimality conditions satisfied only at convergence

Primal residual
\[ r_p = Ax + s - b \to 0 \]

Dual residual
\[ r_d = A^T y + c \to 0 \]

Complementary slackness
\[ s^T y \to 0 \]
Working towards optimality conditions

Optimality conditions satisfied only at convergence

Primal residual
\[ r_p = Ax + s - b \to 0 \]

Dual residual
\[ r_d = A^T y + c \to 0 \]

Complementary slackness
\[ s^T y \to 0 \]

Stopping criteria
\[ \|r_p\| \leq \epsilon_{pri} \]
\[ \|r_d\| \leq \epsilon_{dual} \]
\[ s^T y \leq \epsilon_{gap} \]
Logarithmic Barrier Functions
Smoothed optimality conditions

Optimality conditions

\[ Ax + s - b = 0 \]
\[ A^T y + c = 0 \]
\[ s_i y_i = \tau \]
\[ s, y \geq 0 \]

Same \( \tau \) for every pair

Same optimality conditions for a “smoothed” version of our problem
Smoothed optimality conditions

Optimality conditions

\[ Ax + s - b = 0 \]
\[ A^T y + c = 0 \]
\[ s_i y_i = \tau \quad \text{Same } \tau \text{ for every pair} \]
\[ s, y \geq 0 \]

Same optimality conditions for a “smoothed” version of our problem

Do solutions actually exist?
What do they represent?
Logarithmic barrier

\[ \phi(s) = -\tau \sum_{i=1}^{m} \log(s_i) \quad \text{on domain} \quad s_i > 0 \]

As \( \tau \to 0 \) it approximates

\[ I_{s_i \geq 0} = \begin{cases} 0 & \text{if } s_i \geq 0 \\ \infty & \text{otherwise} \end{cases} \]
Smoothed problem

minimize \( c^T x \)

subject to \( Ax + s = b \)
\( s \geq 0 \)
Smoothed problem

minimize \( c^T x \)
subject to \( Ax + s = b \) \( s \geq 0 \)

\( c^T x + \phi(x) = c^T x - \tau \sum_{i=1}^{m} \log(s_i) \)
subject to \( Ax + s = b \)
Smoothed problem

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax + s = b \\
& \quad s \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad c^T x + \phi(\mathbf{s}) = c^T x - \tau \sum_{i=1}^{m} \log(s_i) \\
\text{subject to} & \quad Ax + s = b
\end{align*}
\]

Dual cost

\[
g(y) = \minimize_{x,s} \mathcal{L}(x, s, y) = c^T x + \phi(s) + y^T (Ax + s - b)
\]
Smoothed problem

minimize $c^T x$
subject to $Ax + s = b$
$s \geq 0$

$\rightarrow$

minimize $c^T x + \phi(x) = c^T x - \tau \sum_{i=1}^{m} \log(s_i)$
subject to $Ax + s = b$

Dual cost

$g(y) = \minimize_{x,s} \mathcal{L}(x, s, y) = c^T x + \phi(s) + y^T (Ax + s - b)$

$\frac{\partial \mathcal{L}}{\partial x} = A^T y + c = 0$

$\frac{\partial \mathcal{L}}{\partial s_i} = -\tau \frac{1}{s_i} + y_i = 0 \implies s_i y_i = \tau$
Central path

minimize \( c^T x - \tau \sum_{i=1}^{m} \log(s_i) \)
subject to \( Ax + s = b \)

Set of points \((x^*(\tau), s^*(\tau), y^*(\tau))\)
with \( \tau > 0 \) such that

\[
Ax + s - b = 0
\]
\[
A^T y + c = 0
\]
\[
s_i y_i = \tau
\]
\[
s, y \geq 0
\]
Central path

minimize \[ c^T x - \tau \sum_{i=1}^{m} \log(s_i) \]
subject to \[ Ax + s = b \]

Set of points \((x^*(\tau), s^*(\tau), y^*(\tau))\) with \(\tau > 0\) such that
\[
\begin{align*}
Ax + s - b &= 0 \\
A^T y + c &= 0 \\
s_i y_i &= \tau \\
s, y &\geq 0
\end{align*}
\]

Main idea
Follow central path as \(\tau \to 0\)
Convergence
Definitions

Primal-dual strictly feasible set

\[ \mathcal{F}^\circ = \{(x, s, y) \mid Ax + s = b, A^T y + c = 0, s, y > 0\} \]

Central path neighborhood

\[ \mathcal{N}(\gamma) = \{(x, s, y) \in \mathcal{F}^\circ \mid s_i y_i \geq \gamma \mu\} \quad \text{with } \gamma \in (0, 1] \quad \text{(almost all the feasible region)} \]
Theorem

[Page 402-406, NO]

[Theorem 14.3] Smallest decrement

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \quad \text{with constant } \delta > 0 \]
Theorem

[Page 402-406, NO]

[Theorem 14.3] Smallest decrement

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \quad \text{with constant } \delta > 0 \]

Iteration complexity

Given \((x_0, s_0, y_0) \in \mathcal{N}(\gamma)\), there exists \(K = O(n \log(1/\epsilon))\) such that

\[ \mu_k \leq \epsilon \mu_0 \quad \text{for all } k \geq K \]
Theorem
[Page 402-406, NO]

[Theorem 14.3] Smallest decrement

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \] with constant \( \delta > 0 \)

Iteration complexity

Given \((x_0, s_0, y_0) \in \mathcal{N}(\gamma)\), there exists \( K = O(n \log(1/\epsilon)) \) such that

\[ \mu_k \leq \epsilon \mu_0 \] for all \( k \geq K \)

Remark Modified versions achieve \( O(\sqrt{n} \log(1/\epsilon)) \)
Iteration complexity proof

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \]
Iteration complexity proof

[Page 402-406, NO]

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \]

(take logarithm)

\[ \log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k \]
Iteration complexity proof

[Page 402-406, NO]

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \]

(take logarithm)

\[ \log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k \]

(apply iteratively)

\[ \log \mu_k \leq k \log (1 - \delta/n) + \log \mu_0 \]
Iteration complexity proof

[Page 402-406, NO]

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \]

(take logarithm)

\[ \log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k \]

(apply iteratively)

\[ \log \mu_k \leq k \log (1 - \delta/n) + \log \mu_0 \]

Since \( \log(1 + \beta) \leq \beta \), \( \forall \beta > -1 \)

\[ \log(\mu_k/\mu_0) \leq k(-\delta/n) \]
Iteration complexity proof

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \]

(take logarithm)
\[
\log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k
\]

(apply iteratively)
\[
\log \mu_k \leq k \log (1 - \delta/n) + \log \mu_0
\]

Since \( \log(1 + \beta) \leq \beta, \quad \forall \beta > -1 \)
\[
\log(\mu_k/\mu_0) \leq k(-\delta/n) \leq \log(\varepsilon)
\]

If \( k(-\delta/n) \leq \log(\varepsilon) \)
then \( \log(\mu_k/\mu_0) \leq \log(\varepsilon) \). Therefore, \( \mu_k/\mu_0 \leq \varepsilon \)
Iteration complexity proof

\[ \mu_{k+1} \leq (1 - \delta/n) \mu_k \]

(take logarithm)
\[ \log \mu_{k+1} \leq \log (1 - \delta/n) + \log \mu_k \]

(apply iteratively)
\[ \log \mu_k \leq k \log (1 - \delta/n) + \log \mu_0 \]

Since \( \log(1 + \beta) \leq \beta, \ \forall \beta > -1 \)
\[ \log(\mu_k/\mu_0) \leq k(-\delta/n) \]

If \( k(-\delta/n) \leq \log(\epsilon) \)
\[ k \geq \frac{\log(\epsilon)}{-\delta/n} \]

then \( \log(\mu_k/\mu_0) \leq \log(\epsilon) \). Therefore, \( \mu_k/\mu_0 \leq \epsilon \)

Rewriting the inequality: \( k \geq (n/\delta) \log(1/\epsilon) \)
Interior-point methods for linear optimization

Today, we learned to:

• **Apply** Newton’s method to solve optimality conditions

• **Analyze** the central path and the smoothed optimality conditions

• **Develop** a prototype primal-dual path-following algorithm
Next lecture

- Practical interior-point method (Mehrotra predictor-corrector algorithm)
- Linear algebra implementation details
- Linear optimization recap