A GRADUATE COURSE

IN

OPTIMIZATION

A GRADUATE COURSE

IN

OPTIMIZATION

CIE6010 Notebook

Prof. Yin Zhang

The Chinese University of Hongkong, Shenzhen

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Contents

Acknowledgments ix

Notations xi

1 Week1 1

1.1 Monday 1

1.1.1 Introduction to Optimizaiton 1

1.2 Wednesday 2

1.2.1 Reviewing for Linear Algebra 2

1.2.2 Reviewing for Calculus 2

1.2.3 Introduction to Optimization 3

2 Week2 7

2.1 Monday 7

2.1.1 Reviewing and Announments 7

2.1.2 Quadratic Function Case Study 8

2.2 Wednesday 11

2.2.1 Convex Analysis 11

3 Week3 17

3.1 Wednesday 17

3.1.1 Convex Analysis 17

3.1.2 Iterative Method 18

3.2 Thursday 22

3.2.1 Announcement 22

3.2.2 Sparse Large Scale Optimization 22

4 Week4 . . . 27

4.1 Wednesday 27

4.1.1 Comments for MATLAB Project . . . 27

4.1.2 Local Convergence Rate . . . 28

4.1.3 Newton's Method . . . 29

4.1.4 Tutorial: Introduction to Convexity . . . 30

5 Week5 . . . 33

5.1 Monday 33

5.1.1 Review . . . 33

5.1.2 Existence of solution to Quadratic Programming . . . 36

5.2 Wednesday 39

5.2.1 Comments about Newton's Method . . . 39

5.2.2 Constant Step-Size Analysis . . . 40

6 Week6 . . . 45

6.1 Monday 45

6.1.1 Announcement . . . 45

6.1.2 Introduction to Quasi-Newton Method . . . 45

6.1.3 Constrainted Optimization Problem . . . 46

6.1.4 Announcement on Assignment . . . 47

6.1.5 Introduction to Stochastic optimization . . . 49

6.2 Tutorial: Monday 49

6.2.1 LP Problem . . . 49

6.2.2 Gauss-Newton Method . . . 50

6.2.3 Introduction to KKT and CQ . . . 51

6.3 Wednesday 52

6.3.1 Review . . . 52

6.3.2 Dual-Primal of LP . . . 53

7 Week7 57

7.1 Monday 57

7.1.1 Announcement 57

7.1.2 Recap about linear programming 57

7.1.3 Optimization over convex set 60

7.2 Wednesday 62

7.2.1 Motivation 62

7.2.2 Convex Projections 63

7.2.3 Feasible diection method 65

8 Week8 69

8.1 Monday 69

8.1.1 Constraint optimization 70

8.1.2 Inequality Constraint Problem 71

8.2 Monday Tutorial: Review for CIE6010 71

9 Week9 79

9.1 Monday 79

9.1.1 Reviewing for KKT 79

9.2 Monday Tutorial: Reviewing for Mid-term 82

10 Week10 83

10.1 Monday 83

10.1.1 Penalty Algorithms 86

10.2 Wednesday 88

Acknowledgments

This book is from the CIE6010 in fall semester, 2018.

CUHK(SZ)

Notations and Conventions

X	Set
$\inf X \subseteq \mathbb{R}$	Infimum over the set X
$\mathbb{R}^{m\times n}$	set of all $m \times n$ real-valued matrices
$\mathbb{C}^{m\times n}$	set of all $m \times n$ complex-valued matrices
x_i	ith entry of column vector $\boldsymbol{x}$
a_{ij}	(i,j)th entry of matrix $\boldsymbol{A}$
$\boldsymbol{a}_i$	ith column of matrix $\boldsymbol{A}$
$\boldsymbol{a}_i^{\mathrm{T}}$	ith row of matrix $\boldsymbol{A}$
$\mathbb{S}^n$	set of all $n \times n$ real symmetric matrices, i.e., $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $a_{ij} = a_{ji}$ for all i,j
$\mathbb{H}^n$	set of all $n \times n$ complex Hermitian matrices, i.e., $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ and $\bar{a}_{ij} = a_{ji}$ for all i,j
$\boldsymbol{A}^{\mathrm{T}}$	transpose of $\boldsymbol{A}$, i.e, $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{T}}$ means $b_{ji} = a_{ij}$ for all i,j
$\boldsymbol{A}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{A}$, i.e, $\boldsymbol{B} = \boldsymbol{A}^{\mathrm{H}}$ means $b_{ji} = \bar{a}_{ij}$ for all i,j
$\mathrm{trace}(\boldsymbol{A})$	sum of diagonal entries of square matrix $\boldsymbol{A}$
$\mathbf{1}$	A vector with all 1 entries
$\mathbf{0}$	either a vector of all zeros, or a matrix of all zeros
$\boldsymbol{e}_i$	a unit vector with the nonzero element at the ith entry
$\mathcal{C}(\boldsymbol{A})$	the column space of $\boldsymbol{A}$
$\mathcal{R}(\boldsymbol{A})$	the row space of $\boldsymbol{A}$
$\mathcal{N}(\boldsymbol{A})$	the null space of $\boldsymbol{A}$
$\mathrm{Proj}_{\mathcal{M}}(\boldsymbol{A})$	the projection of $\boldsymbol{A}$ onto the set $\mathcal{M}$

Chapter 1

Week1

1.1. Monday

1.1.1. Introduction to Optimizaiton

The usual optimization formulation is given by:

$$\begin{aligned} &\min f(\boldsymbol{x}), \quad \text{where } f:\mathbb{R}^n \mapsto \mathbb{R} \\ &\text{such that} \quad \boldsymbol{x} \in X \subseteq \mathbb{R}^n \end{aligned}$$

One example of the set X is given by:

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} C_i(\boldsymbol{x}) = \boldsymbol{0}, i = 1,2,\dots,m \le n \\ h_i(\boldsymbol{x}) \ge \boldsymbol{0}, i = 1,2,\dots,p \end{array} \right\}$$

Linear programming can be easily solved, but Integer linear programming is much harder. The equivalent LP formulation is given by:

$$\begin{aligned} \min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{c} \le \boldsymbol{B}\boldsymbol{x} \le \boldsymbol{c}' \end{aligned}$$

1.2. Wednesday

1.2.1. Reviewing for Linear Algebra

Questions:

- What is the necessary and sufficient condition for the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ to have a solution $\boldsymbol{x}$?

  Answer: $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A})$.

- For $\boldsymbol{A} \in \mathbb{S}^n$, what is the necessary and sufficient condition for $\boldsymbol{A} \succeq 0$?

  Answer: $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} \geq 0$ for $\forall x \in \mathbb{R}^n$; or $\lambda_i(\boldsymbol{A}) \geq 0$ for all i.

1.2.2. Reviewing for Calculus

For function $f : \mathbb{R}^n \mapsto \mathbb{R}$:

- We use notation $f \in \mathcal{C}^n$ to denote f is **continuously differentiable to nth order**. This course will basically deal with such functions.

- We use notation $\nabla f(x)$ to denote the **Gradient** of f at x; and $\nabla^2 f(x)$ denotes the second order derivative of f at x. Note that $\nabla^2 f(x) \in \mathbb{S}^n$ for $f \in \mathcal{C}^1$.

- We use notation $\mathbb{S}^n$ to denote the set of all symmetric $n \times n$ matrices, i.e.,

  $$\mathbb{S}^n = \{\boldsymbol{X} \in \mathbb{R}^{n \times n} \mid \boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X}\}$$

  Moreover, $\mathbb{S}^n_+$ denotes the set of all symmetric $n \times n$ matrices with all eigenvalues non-negative:

  $$\mathbb{S}^n_+ = \{\boldsymbol{X} \in \mathbb{R}^{n \times n} \mid \boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X} \succeq 0\}$$

1.2.3. Introduction to Optimization

The usual optimization formulation is given by:

$$\begin{aligned} &\min f(\boldsymbol{x}), && \text{where } f:\mathbb{R}^n \mapsto \mathbb{R} \\ &\text{such that} && \boldsymbol{x} \in X \subseteq \mathbb{R}^n \end{aligned}$$

- The simplest case for the constraint is $X = \mathbb{R}^n$, which leads to **unconstrainted** optimization problem.
- Or $X = P$ is a **polyhedron**, i.e., the boundaries for the region are all lines.

Definition 1.1 [Constraint Regions] In space $\mathbb{R}^n$,

- the hyper-plane is defined as:
$$\left\{\boldsymbol{x} \middle| \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{\beta}\right\}$$
with constants $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{\beta} \in \mathbb{R}$
- the half-space is defined as
$$\left\{\boldsymbol{x} \middle| \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} \le \boldsymbol{\beta}\right\}$$
- the polyhedron is defined as the **intersection** of a **finite** number of hyperplanes or half-spaces

Next, we give the definition for the basic optimization problem:

Definition 1.2 [Linear Programming] The Linear Programming is given by:

$$\begin{aligned} &\min && \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \\ &\text{such that} && \boldsymbol{x} \in P(\text{polyhedron}) \end{aligned}$$

Or it can be reformulated as:

$$
\begin{aligned}
\min \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \\
\text{such that} \quad & \boldsymbol{A}_I\boldsymbol{x} \le \boldsymbol{b}_I \\
& \boldsymbol{A}_E\boldsymbol{x} = \boldsymbol{b}_E \in \mathbb{R}^m, \quad m < n.
\end{aligned}
$$

Definition 1.3 [Optimality] $\boldsymbol{x}^*$ is said to be :

- the **local minimum** of $f(\boldsymbol{x})$ if there exists small ϵ such that

$$f(\boldsymbol{x}^*) \le f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}^*, \epsilon) \bigcap X := \{\boldsymbol{x} \mid \|\boldsymbol{x} - \boldsymbol{x}^*\| \le \epsilon\} \bigcap X$$

- the **global minimum** if

$$f(\boldsymbol{x}^*) \le f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in X$$

R Unless specified, when we want to minimize a non-convex function, it usually means we only find its **local minimum**. This is because usually the local minimum is good enough.

The optimization task is essentially find $\boldsymbol{x}^*$ such that

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}) \in \mathbb{R}^n.$$

philosophy (optimization sufficient and necessity). philosophy of relaxation (convex nulls)

The Optimality conditions are the **most important** theoretical tools for optimization.

Theorem 1.1 — **Optimality condition.** The optimality condition contains

1. Necessary Condition (exclude non-optimal points):

$$n = 1 \text{ special case: } \begin{cases} \text{1st order: } f'(x) = 0 \\ \text{2nd order: } f''(x) \geq 0 \end{cases} \implies \begin{cases} \text{1st order: } \nabla f(x) = 0 \\ \text{2nd order: } \nabla f^2(x) \succeq 0 \end{cases}$$

2. Sufficient Condition (may identify optimal solutions)

$$n = 1 \text{ special case: } \begin{cases} \text{1st order: } f'(x) = 0 \\ \text{2nd order: } f''(x) > 0 \end{cases} \implies \begin{cases} \text{1st order: } \nabla f(x) = 0 \\ \text{2nd order: } \nabla f^2(x) \succ 0 \end{cases}$$

Proof. The $n = 1$ special case can imply the general case for optimality condition. For multivariate f, we set $\boldsymbol{x} = \boldsymbol{x}^* + td$ with t to be the stepsize and d to be the direction. For fixed t and d, we define $h(t) = f(\boldsymbol{x}) = f(\boldsymbol{x}^* + td)$. It follows that

$$h'(t) = \nabla^{\mathrm{T}} f(\boldsymbol{x}^* + td)d$$

We find $h'(0) = \nabla^{\mathrm{T}} f(\boldsymbol{x}^*)d$ for $\forall$ d, which implies $\nabla f(\boldsymbol{x}^*) = 0$. ■

Note that there is a gap between necessary and sufficient conditions, which puts us in an embarrassing position. However, the convex condition can save us:

Theorem 1.2 If f is convex in $\mathcal{C}^1$, then $\nabla f(\boldsymbol{x}) = 0$ is the **necessary** and **sufficient** condition.

Chapter 2

Week2

2.1. Monday

2.1.1. Reviewing and Announments

Tutorial: Thursday 7:00pm -9:00pm, ChengDao 208

Homework is due every Monday.

The first homework has been uploaded.

To proof the optimality condition in $\mathbb{R}^n$, we set $h(t) = f(x^* + td)$ for fixed x^* and d. It follows that

$$h'(t) = \nabla^{\mathrm{T}} f(x^* + td)d$$

and

$$h''(t) = d^{\mathrm{T}} \nabla^2 f(x^* + td)d$$

By the optimality condition for $\mathbb{R}$, we derive the necessary condition:

$$\begin{cases} h'(0) = \nabla^{\mathrm{T}} f(x^*)d = 0 \text{ for } \forall d \implies \nabla f(x^*) = 0; \\ h''(0) = d^{\mathrm{T}} \nabla^2 f(x^*)d = 0 \text{ for } \forall d \implies \nabla^2 f(x^*) \succeq 0 \end{cases}$$

together with the sufficient condition:

$$\begin{cases} \nabla f(x^*) = 0; \\ \nabla^2 f(x^*) \succ 0 \end{cases}$$

2.1.2. Quadratic Function Case Study

Given a quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}$$

w.l.o.g., assume the matrix $\boldsymbol{Q}$ is symmetric (recall the quadratic section studied in linear algebra).

Definition 2.1 [Stationarity] A point $\boldsymbol{x}^*$ is said to be the stationary point of $f(\boldsymbol{x})$ if $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$. ■

To minimize such a function without constraint, we apply the optimality condition:

1. The first order optimality condition is given by:

$$\nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b} = \mathbf{0}$$

   The stationary point of the quadratic function $f(\boldsymbol{x})$ exists iff $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{Q})$.

2. The second order necessary condition should be:

$$\nabla^2 f(\boldsymbol{x}) = \boldsymbol{Q} \succeq 0$$

For this special case, if $\boldsymbol{Q} \succeq 0$, then $f(\boldsymbol{x})$ is convex, the solutions to $\nabla f(\boldsymbol{x}) = \mathbf{0}$ are local minimum points. Furthermore, they are global minimum points (prove by Taylor Expansion). However, for general functions, we cannot obtain such good results.

Least Squares Problem. Such a problem has been well-studied in statistics given by:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) := \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

The first order derivative of the minimizer should satisfy:

$$\nabla f(\boldsymbol{x}) = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$$

Note that $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \in \mathcal{C}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$, thus the least squares problem always has a solution. However, such a solution is not unique unless $\boldsymbol{A}$ is full rank.

A Non-trival Quadratic Function. To minimize the function

$$f(x,y) = \frac{1}{2}(\alpha x^2 + \beta y^2) - x$$

We take the first order derivative to be zero:

$$\nabla f(x,y) = \begin{bmatrix} \alpha x - 1 \\ \beta y \end{bmatrix} = \mathbf{0}$$

The second order derivative is given by:

$$\nabla^2 f(x,y) = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$$

The optimal solutions depend on the value of α and β: (although we haven't introduce the definition for convex formally)

- If $\alpha, \beta > 0$, then this problem is **strongly convex**. By the necessary and sufficient optimality condition for convex problem, we find that $(\frac{1}{\alpha}, 0)$ is the unique local minimum (It is also the global minimum by plotting the figure).

- If $\alpha = 0$, this problem has no solution. The objective value $f(x,y) \to -\infty$ as $x \to \infty$.

- If $\beta = 0, \alpha > 0$, this problem is convex. By the necessary and sufficient optimality condition for convex problem, $\{(\frac{1}{\alpha}, \xi) \mid \xi \in \mathbb{R}\}$ is the set of local minimum. (By plotting the graph, we find that such set is the set of global minimum points)

- For $\alpha > 0, \beta < 0$ case, this problem is non-convex. Actually, $f(x,y) \to -\infty$ as $y \to \infty$. Hence, this problem has no global minimum point.

A Non-trival Function Study. To minimize the function

$$\begin{aligned}
&\min && f(\boldsymbol{y}) = e^{y_1} + \cdots + e^{y_n} \\
&\text{such that} && y_1 + \cdots + y_n = S
\end{aligned}$$

We can transform such a constrainted optimization problem into unconstrainted. Let $y_n = S - y_1 - \cdots - y_{n-1}$ and substitute it into the objective function, it suffices to solve

$$\min e^{y_1} + \cdots + e^{y_{n-1}} + e^{S-y_1-\cdots-y_{n-1}}$$

The stationary point should satisfy:

$$e^{y_i} = e^{S-y_1-\cdots-y_{n-1}}, \qquad i = 1,2,\dots,n-1$$

Or equivalently, $y_1 = y_2 = \cdots = y_{n-1} = y_n$. Hence we derive the unique stationary point:

$$y_1^* = y_2^* = \cdots = y_n^* = \frac{S}{n}$$

The value on the stationary point is $f(y^*) = ne^{S/n}$. By checking the second order sufficient optimality condition,

$$\frac{f}{\partial y_i \partial y_j} = \begin{cases} e^{y_i} + e^{S-y_1-\cdots-y_{n-1}} & i = j \\ e^{S-y_1-\cdots-y_{n-1}} & i \neq j \end{cases} \implies \nabla^2 f = e^{S-y_1-\cdots-y_{i-1}-y_i-y_{n-1}} \boldsymbol{E} + \operatorname{diag}(e^{y_1}, \dots, e^{y_{n-1}})$$

where $\boldsymbol{E}$ is a matrix with entries all ones. Thus $\nabla^2 f \succ 0$ for any stationary point. By the second order sufficient optimality condition, this stationary point is local minimum. Actually, for this special problem, this unique local minimum point is the global minimum.

R In this problem, we find that this stationary point is the unique local minimum point, but the unique local minimum point is not necessarily the global minimum point, unless the function is **coercive** or the feasible region is compact. Here is the counter-example: $f(x) = x^2 - x^4$. We will discuss the definition for coercive in the future.

2.2. Wednesday

2.2.1. Convex Analysis

This lectue will study the convex analysis.

Definition 2.2 [Convex] The subset $\mathcal{C} \subseteq \mathbb{R}^n$ is convex if

$$x, y \in \mathcal{C} \implies \{\lambda x + (1-\lambda)y \mid \alpha \in [0,1]\} \subset \mathcal{C},$$

i.e., the line segment between arbitrarily two elements lines in $\mathcal{C}$ ■

R Intersections of convex sets are convex. Empty set is assumed to be convex.

Definition 2.3 [Convex] The function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if dom f is convex and

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

for $\forall x, y \in \text{dom } f$ and $\forall \lambda \in [0,1]$, i.e., the function evaluated in the line segment is lower than secant line between x and y (f lies below secant line). ■

R

- f is convex iff $-f$ is concave. (The concave definition simply changes the inequality direction in Def.(2.3))
- Affines are both convex and concave.
- The convexity depends on the domain of the function.

For a second order differentiable function, we have a much easier way to determine its convexity.

Theorem 2.1 If $f \in \mathcal{C}^1$, then the followings are equivalent:

1. f is convex
2. $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla^{\mathrm{T}} f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})$ for $\forall x, y \in \operatorname{dom} f$, i.e., f lines above the tangent line.

Proof. 1. From the definition for convexity,

$$f(y) - f(x) \geq \frac{f(\lambda x + (1-\lambda)y) - f(x)}{1-\lambda}$$

Letting $\lambda \to 1$, the RHS becomes a direction derivative:

$$f(y) - f(x) \geq \nabla^{\mathrm{T}} f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})$$

2. To show the converse, we let $z = \lambda x + (1-\lambda)y$. By applying the inequality in (2.1) twice, we have

$$f(\boldsymbol{x}) \geq f(\boldsymbol{z}) + \nabla^{\mathrm{T}} f(\boldsymbol{z})(\boldsymbol{x}-\boldsymbol{z}) \tag{2.1}$$

$$f(\boldsymbol{y}) \geq f(\boldsymbol{z}) + \nabla^{\mathrm{T}} f(\boldsymbol{z})(\boldsymbol{y}-\boldsymbol{z}) \tag{2.2}$$

Letting Eq.(2.1) times λ add Eq.(2.2) times $(1-\lambda)$, we derive that f is convex.

■

Theorem 2.2 If $f \in \mathcal{C}^2$, then the followings are equivalent:

1. f is convex
2. $\nabla^2 f(\boldsymbol{x}) \succeq 0$ for $\forall x \in \operatorname{dom} f$.

Proof. We rewrite $f(\boldsymbol{y})$ by applying Taylor expansion:

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla^{\mathrm{T}} f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}} \nabla^2 f(\boldsymbol{x} + t(\boldsymbol{y}-\boldsymbol{x}))(\boldsymbol{y}-\boldsymbol{x}), \tag{2.3}$$

for some $t \in [0,1]$.

1. If f is convex, from Theorem(2.1) and Eq.(2.3), we derive

$$(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\nabla^2 f(\boldsymbol{x}+t(\boldsymbol{y}-\boldsymbol{x}))(\boldsymbol{y}-\boldsymbol{x}) \geq 0 \Longrightarrow \frac{(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}}{\|\boldsymbol{y}-\boldsymbol{x}\|}\nabla^2 f(\boldsymbol{x}+t(\boldsymbol{y}-\boldsymbol{x}))\frac{(\boldsymbol{y}-\boldsymbol{x})}{\|\boldsymbol{y}-\boldsymbol{x}\|} \geq 0$$

Set $d := \frac{(\boldsymbol{y}-\boldsymbol{x})}{\|\boldsymbol{y}-\boldsymbol{x}\|}$ and let $\boldsymbol{y} \to \boldsymbol{x}$, we derive

$$\boldsymbol{d}^{\mathrm{T}}\nabla^2 f(\boldsymbol{x})\boldsymbol{d} \geq 0,$$

which implies $\nabla^2 f(\boldsymbol{x}) \succeq 0$ since $\boldsymbol{d}$ could have an arbitrary direction.

2. To show the converse, due to the semidefiniteness of $\nabla^2 f(\boldsymbol{x})$, we obtain a new inequality from Eq.(2.3):

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla^{\mathrm{T}} f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})$$

From Theorem(2.1) we imply f is convex.

■

Definition 2.4 [Epigraph] The Epigraph of f is given by:

$$\mathsf{Epi}(f) := \{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid x \in \mathsf{dom}\ f, t \geq f(x)\} \subseteq \mathbb{R}^{n+1}$$

Theorem 2.3 f is convex iff $\mathrm{Epi}(f)$ is convex.

Proof. 1. Suppose f is convex. For any $(x,t),(y,s) \in \mathrm{Epi}(f)$, it suffices to show

$$(\lambda x + (1-\lambda)y, \lambda t + (1-\lambda)s) \in \mathrm{Epi}(f) \Longleftrightarrow \lambda t + (1-\lambda)s \geq f(\lambda x + (1-\lambda)y).$$

The convexity of f implies that

$$
\begin{aligned}
f(\lambda x+(1-\lambda)y) &\le \lambda f(x)+(1-\lambda)f(y) \\
&\le \lambda t+(1-\lambda)s.
\end{aligned}
$$

2. The reverse direction is obvious by applying definitions.

■

Definition 2.5 [Strict Convex] The function $f:\mathbb{R}^n\mapsto\mathbb{R}$ is strict convex if dom f is convex and

$$f(\lambda x+(1-\lambda)y) < \lambda f(x)+(1-\lambda)f(y)$$

for $\forall x\neq y, x,y\in$ dom f and $\forall\lambda\in(0,1)$

R Strict convex implies the uniquesness of minimum

However, for function $f(x)=\frac{1}{x}$, the curvature becomes more and more flat. We want to exclude such kind of functions.

Definition 2.6 The function $f:\mathbb{R}^n\mapsto\mathbb{R}$ is said to be strongly convex if dom f is convex and $\exists\alpha>0$ such that $f(\boldsymbol{x})-\alpha\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ is convex; or equivalently,

$$f(\boldsymbol{y})\ge f(\boldsymbol{x})+\nabla^{\mathrm{T}}f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x})+\frac{\alpha}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$

R The strong convexity places a quadratic lower bound in the curvature of the function, i.e., the function must rise up at least as fast as a quadratic function. How fast it rises depends on the parameter α.

The convexity properties are extremely useful in forcing optimization algorithms to rapidly converge to optima. However, most functions are not convex. The most important result that requires convexity is given below:

Theorem 2.4 If f is convex in $\mathcal{C}^1$, then $\nabla f(\boldsymbol{x}) = 0$ is the **necessary** and **sufficient** condition for **global** minimum.

Note that convex function does not have a local minimum that is not global minimum.

Proof. If $f \in \mathcal{C}^1$ is convex, recall the Theorem(2.1) that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla^{\mathrm{T}} f(\boldsymbol{x})(\boldsymbol{y} - \boldsymbol{x}) \tag{2.4}$$

1. If $\nabla f(\boldsymbol{x}) = 0$, then Eq.(2.3) implies $f(\boldsymbol{y}) \geq f(\boldsymbol{x})$ for $\forall y$.
2. If $\boldsymbol{x}$ is the global minimum, recall the optimality conditon, $\nabla f(\boldsymbol{x}) = \mathbf{0}$.

$\blacksquare$

In practice, we cannot solve all convex optimization problems. So we need to carefully study the structure of every problem we have faced.

Chapter 3

Week3

3.1. Wednesday

Assignment 2 posted.

CIE6010: Exercise 1.2.9 and 1.3.9; together with MATLAB project.

3.1.1. Convex Analysis

Last time we have shown that for a unconstrained problem, $\nabla f(\boldsymbol{x}) = 0$ is the necessary and sufficient condition for global minimum ensurance. However, the case for constrained problem will be different.

Proposition 3.1 For the **constrained** problem

$$\begin{aligned} \min \quad & f(x) \\ & \boldsymbol{x} \in X \subseteq \mathbb{R} \\ & f \text{ is convex in } \mathcal{C}^1 \end{aligned}$$

$\boldsymbol{x}$ is a global minimum iff

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \geq 0$$

for $\forall \boldsymbol{y} \in X$.

Proof. Since f is convex, the inequality below holds:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle, \forall \boldsymbol{x}, \boldsymbol{y} \in X$$

Note that $\boldsymbol{x}$ is a global minimum iff $f(\boldsymbol{y}) \geq f(\boldsymbol{x}), \forall \boldsymbol{y} \in X$. Combining the inequality above, the proof is complete. ■

R

- Such a condition is not so useful unless $\boldsymbol{y}$ lies in the whole space, at that time we have no choice but $\nabla f(\boldsymbol{x}) = \mathbf{0}$. (otherwise we can construct a $\boldsymbol{y}$ to let the inner product negative.)
- An equivalent version of the condition is that every **feasible** direction is **ascending**.

Definition 3.1 [Descending Direction] The vector $\boldsymbol{d} \in \mathbb{R}^n$ is said to be a **descending direction** of f at $\boldsymbol{x}$ if

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{d} \rangle < 0.$$

This definition is the motivation of descent method.

3.1.2. Iterative Method

Definition 3.2 [Descent Method] At any non-stationary $\boldsymbol{x}$, i.e., $\nabla f(x) \neq \mathbf{0}$, we find the descenting direction d, i.e., $\langle \nabla f(\boldsymbol{x}), \boldsymbol{d} \rangle < 0$. We update our old $\boldsymbol{x}$ as:

$$\boldsymbol{x}^{r+1} \leftarrow \boldsymbol{x}^r + \alpha^r \boldsymbol{d}^r, \quad \alpha > 0.$$

The key is how to choose $\boldsymbol{d}$ and α. We have a general formula for $\boldsymbol{d}$:

$$\boldsymbol{d} = -\boldsymbol{D} \cdot \nabla f(\boldsymbol{x}),$$

where $\boldsymbol{D} \in \mathbb{S}^n$ and $\boldsymbol{D} \succ 0$. (Verify by yourself that $\boldsymbol{d}$ satisfies the descending direction definition)

1. $\boldsymbol{D} = \boldsymbol{I}$ implies gradient method (Steepest Descent).
2. $\boldsymbol{D} = \left(\nabla^2 f(\boldsymbol{x})\right)^{-1}$ implies the Newton's method.

Nonlinear LS. The optimization problem is

$$\begin{aligned} \min \qquad & f(\boldsymbol{x}) = \tfrac{1}{2}\textstyle\sum_{i=1}^m g_i^2(\boldsymbol{x}) := \tfrac{1}{2}\|g(\boldsymbol{x})\|_2^2 \\ & g(\boldsymbol{x}) = \begin{pmatrix} g_1(\boldsymbol{x}) & g_2(\boldsymbol{x}) & \cdots & g_m(\boldsymbol{x}) \end{pmatrix}^{\mathrm{T}} \end{aligned}$$

The gredient function is

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= \sum_{i=1}^m g_i(\boldsymbol{x}) \nabla g_i(\boldsymbol{x}) \\ &= \underbrace{\begin{bmatrix} \nabla g_1(\boldsymbol{x}) & \cdots & \nabla g_m(\boldsymbol{x}) \end{bmatrix}}_{\nabla g(\boldsymbol{x}) \in \mathbb{R}^{n\times m}} \begin{bmatrix} g_1(\boldsymbol{x}) \\ \vdots \\ g_m(\boldsymbol{x}) \end{bmatrix} \\ &= \nabla g(\boldsymbol{x}) \cdot g(\boldsymbol{x}) \\ &= \langle \boldsymbol{J}(\boldsymbol{x}), g(\boldsymbol{x}) \rangle, \end{aligned}$$

where $J(\boldsymbol{x}) \in \mathbb{R}^{m\times n}$ is said to be the Jacobian matrix of $g(\boldsymbol{x})$.

The second order derivative function is given as: (complete the calculation process by yourself)

$$\nabla^2 f(\boldsymbol{x}) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{J}(\boldsymbol{x}) + \sum_{i=1}^m g_i(\boldsymbol{x}) \nabla^2 g_i(\boldsymbol{x}),$$

the second term in RHS is complicated and hard to compute. To solve this LS problem, the Gauss-Newton method directly ignore it, which leads to the descent direction

$$\boldsymbol{d} = -(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}} g(\boldsymbol{x})$$

Choice of Step Length α. We apply the Limited Minimization Rule to find α, i.e., for fixed $s > 0$, choose α^r such that

$$\min_{\alpha^r \in (0,s]} f(\boldsymbol{x}^r + \alpha^r \boldsymbol{d}^r).$$

Usually this rule is too computationally expensive. The alternative ways are:

- Choose α just as a constant
- Choose $\alpha^r \to 0$ as $r \to \infty$ but also satisfies the infinite travel condition

$$\sum_{r=0}^{\infty} \alpha^r = \infty$$

Adding Lipschitz condition will make the choice of step-length easier:

Definition 3.3 [Lipschitz Continuous] ∇f is **Lipschitz continuous** with Lipschitz constant L if

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$$

for all $\boldsymbol{x}, \boldsymbol{y}$.

R

- It is useful to note that convexity places a lower bound on the growth of the function at every point; whereas Lipschitzness places a upper bound on the growth of the function that is linear in the perturbation i.e., $\|\boldsymbol{x} - \boldsymbol{y}\|_2$. Also note that Lipschitz functions need not be differentiable. However, differentiable functions with bounded gradients are always Lipschitz.
- The Lipshitz conditon induces that for iterative method we have

$$f(\boldsymbol{x}^r) - f(\boldsymbol{x}^{r+1}) \geq \frac{L}{2}\|\nabla f(\boldsymbol{x}^r)\|^2.$$

  From this inequality, we imply that the result of itervative convergence is $\nabla f(\boldsymbol{x}^r) \to 0$, but the minimum point is still un-guaranteed. In Deep Learing people often train the data using this way, which is not so rigorous.

Convergence Rate Analysis. We apply the Lipschitzness to analysis the rate of convergence first. Setting $h(t) = f(\boldsymbol{x} + t\alpha \boldsymbol{d})$, we find that

$$
\begin{aligned}
f(\boldsymbol{x} + \alpha\boldsymbol{d}) - f(\boldsymbol{x}) &= h(1) - h(0) = \int_0^1 h'(t)\,\mathrm{d}t \\
&= \int_0^1 \langle \nabla f(\boldsymbol{x} + t\cdot\alpha\boldsymbol{d}), \alpha\boldsymbol{d}\rangle\,\mathrm{d}t \\
&= \int_0^1 \left[\langle \nabla f(\boldsymbol{x} + t\cdot\alpha\boldsymbol{d}), \alpha\boldsymbol{d}\rangle - \langle \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle + \langle \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle\right]\mathrm{d}t \\
&= \langle \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle + \int_0^1 \langle \nabla f(\boldsymbol{x} + t\cdot\alpha\boldsymbol{d}) - \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle\,\mathrm{d}t \\
&\le \langle \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle + \int_0^1 \|\nabla f(\boldsymbol{x} + t\cdot\alpha\boldsymbol{d}) - \nabla f(\boldsymbol{x})\|\cdot\|\alpha\boldsymbol{d}\|\,\mathrm{d}t \\
&\le \langle \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle + L\int_0^1 t\alpha^2\|\boldsymbol{d}\|^2\,\mathrm{d}t \\
&= \underbrace{\langle \nabla f(\boldsymbol{x}), \alpha\boldsymbol{d}\rangle}_{\text{negative}} + \frac{L\alpha^2\|d\|^2}{2}
\end{aligned}
$$

Choice of Step Length. To get the optimal step length α, differentiating the RHS w.r.t. α leads to

$$
\langle \nabla f(\boldsymbol{x}), \boldsymbol{d}\rangle + L\alpha\|\boldsymbol{d}\|^2 = 0 \implies \alpha = -\frac{\langle \nabla f(\boldsymbol{x}), \boldsymbol{d}\rangle}{L\|\boldsymbol{d}\|^2} > 0,
$$

which seems a reasonable choice. If $\boldsymbol{d}$ is the steepest descent direction, the step-length becomes a constant:

$$
\alpha = \frac{1}{L}.
$$

3.2. Thursday

3.2.1. Announcement

The assignment 2 requires to do a MATLAB project. The grade usually depends on your understanding of the reading materials and the time spent on experimentation.

3.2.2. Sparse Large Scale Optimization

Given an underlying signal $\boldsymbol{x} \in \mathbb{R}^n$ satisfying the undermined system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, we aim to recover the desired $\hat{\boldsymbol{x}}$. It suffices to solve the optimization problem

$$\begin{aligned} \min \quad & \|\boldsymbol{D}\boldsymbol{x}\|_1 \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{A} \in \mathbb{R}^{m\times n}, m < n \end{aligned}$$

with $\boldsymbol{D}$ to be the difference matrix and $\min \|\boldsymbol{D}\boldsymbol{x}\|_1$ is sparsity promoting. Here we list two basic but effective ways to solve such a problem.

Linear Programming Approach. One way is to reformulate the problem into LP.

1. Define new variables $t_i = |(\boldsymbol{D}\boldsymbol{x})_i|$, we can reformulate the origin problem as:

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^n t_i \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & -t_i \le \textstyle\sum_{k=1}^n d_{ik}x_k \le t_i \\ & \boldsymbol{A} \in \mathbb{R}^{m\times n}, m < n \end{aligned}$$

2. Alternatively, recall what we have learnt in MAT3007. Define slack variables

$(\boldsymbol{Dx})_i = u_i - v_i$, where $u_i := (\boldsymbol{Dx})_i^+, v_i = (\boldsymbol{Dx})_i^-$. It suffices to solve

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^n (u_i + v_i) \\ \text{s.t.} \quad & \boldsymbol{Ax} = \boldsymbol{b} \\ & -t_i \le \textstyle\sum_{k=1}^n d_{ik} x_k \le t_i \\ & \boldsymbol{A} \in \mathbb{R}^{m \times n}, m < n \\ & u_i, v_i \ge 0 \end{aligned}$$

However, linear programming is not the optimal way to solve large-scale problem.

Gredient-Based Approach. We can also transform it into the unconstraint minimization problem, i.e., we add the penalty for the constraint $\boldsymbol{Ax} - \boldsymbol{b} = \boldsymbol{0}$:

$$\min \|\boldsymbol{Dx}\|_1 + \frac{\mu}{2}\|\boldsymbol{Ax} - \boldsymbol{b}\|^2$$

You may see that this reformulation is not exactly equivalent to the origin problem. However, it is not meaningful to stress $\boldsymbol{Ax}$ should exactly equal to $\boldsymbol{b}$, as there exsists some noise perturbing the equality in nature.

Another problem is that this objective function is not differentiable once there is at least zero entry from $\boldsymbol{Dx}$. Thus we do the approximation

$$|t| \approx \sqrt{t^2 + \sigma}, \text{ for small } \sigma > 0.$$

Hence, it suffices to solve

$$\min f(x) := \Theta_\sigma(\boldsymbol{Dx}) + \frac{\mu}{2}\|\boldsymbol{Ax} - \boldsymbol{b}\|^2 \tag{3.1}$$

where

$$\Theta_\sigma(\boldsymbol{y}) = \sum_{i=1}^n \sqrt{y_i^2 + \sigma}$$

Descent Direction. Since problem(3.1) is convex, taking the derivative leads to minimum point. Hence we use the gredient descent method, i.e., $\boldsymbol{d} = -\nabla f(\boldsymbol{x})$.

Although this direction is not optimal (trying another direction may be faster after several iterations), let's assume we are short-sighted such that we just want to take the steepest direction.

Hence the iterative algorithm to solve this problem can be summarized into one formula: Take a initial guess $\boldsymbol{x}^0$, then for $r = 0, 1, 2 \ldots$

$$\boldsymbol{x}^{r+1} = \boldsymbol{x}^r - \alpha^r \nabla f(\boldsymbol{x}^r)$$

Stopping Criteria. The stopping criteria has two conditions, either one is satisfied is ok. Always keep mind of scaling for stopping criteria, i.e., how large of an objective should depend on the scale of the problem.

- First is $\|\nabla f(\boldsymbol{x}^k)\| \le 10^{-2} \|\nabla f(\boldsymbol{x}^0)\|$, i.e., the iterative method converge to the near stationary point
- Another is $|f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1})| \le 10^{-8} |f(\boldsymbol{x}^k)|$, i.e., the function does not change too much.

The next questions turn out that how to choose initial guess? How to choose step-length? Is steepest descent usuallt effective?

1. For large-scale optimization, the steepest descent is usually one of the **best** way among iterative methods.
2. To choose the initial guess, sometimes we choose the LS solution, i.e., enter the matlab command $A'A/A'b$.
3. Last lecture we tell that we can choose step-length to be the Lipstichz constant, but the disadvantage is that the constant L for large scale optimization is too small. We have a better alternative.

Armijo Condition and BB Step. The motivation is that we aim to let

$$f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) \le f(\boldsymbol{x}^k) + C_1 \alpha \langle \nabla f(\boldsymbol{x}), \boldsymbol{d}^k \rangle, (0 < C_1 < 1) \tag{3.2}$$

i.e., our updated function value should be at least less than the old function minus the descent decrease gain, i.e., it should sufficiently decrease faster than a constant times the steepest gredient descent.

This method has a geometrically meaning:

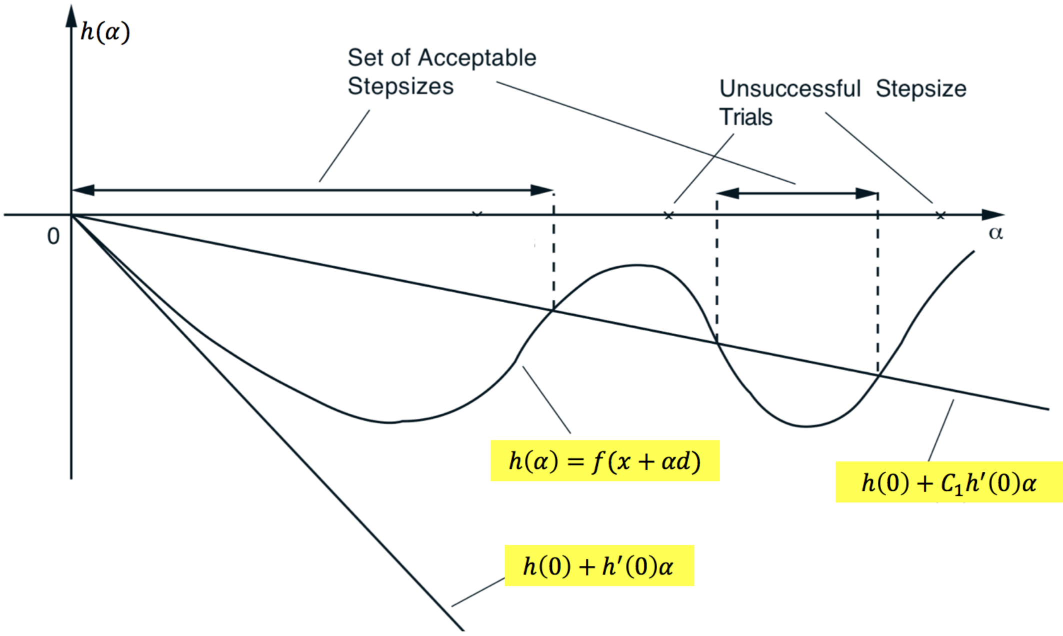

Figure 3.1: Geometric Interpretaiton of Armijo Condition

We set $h(\alpha) := f(\boldsymbol{x} + \alpha \boldsymbol{d})$, then $h'(0) = \langle \nabla f(\boldsymbol{x} + \alpha \boldsymbol{d}), \boldsymbol{d} \rangle$, thus the tangent line at $h(0)$ is given by:

$$h(0) + \alpha h'(0) := f(\boldsymbol{x}) + \alpha \langle \nabla f(\boldsymbol{x} + \alpha \boldsymbol{d}), \boldsymbol{d} \rangle$$

Geometrically we can see that no α can be chosen such that the updated function value $h(\alpha)$ is less than this tangent line. Hence we make the tangent line more flat, i.e., we want to find α such that the updated function value $h(\alpha)$ is below the line $h(0) + C_1 h'(0)\alpha$, $0 < C_1 < 1$:

$$h(\alpha) \le h(0) + C_1 h'(0)\alpha$$

How to choose such α? Take a initial long step-length $\bar{\alpha}$ first, if condition(3.2) is not satisfied, try step length $\beta\bar{\alpha}, \beta^2\bar{\alpha}, \ldots$ respectively. (Take a big step, if not satisfied, shorten the step.)

R However, it is not sugested to do that. Although it is mathematically true, during the computer run, the step-length will decrease exponentially.

How to choose C_1? Empirically, $C_1 = 10^{-3}$ or 10^{-4}, i.e., it is very flat.

How to choose initial $\bar{\alpha}$? It depends on the scale of functionm which requires for your reading of materials.

How to choose the value of β? Do the experiment. (0.5, 0.8 for example).

Chapter 4

Week4

4.1. Wednesday

4.1.1. Comments for MATLAB Project

You need to take care of several things during your assignment:

Do Not Repeat Computation. For example, enter

$$|f(x^{k+1}) - f(x^k)| \le 10^{-\varepsilon}|f(x^k)|,$$

is very bad, since you evaluated this function three times.

Arrange Computation Properly. For example, compute

$$\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}$$

is very expensive, but $\boldsymbol{x}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A})$ is not. Be aware of the size of matrices or vectors.

Appreciate sparsity. For example, when faced with high-dimensional matrices, compute $\boldsymbol{A}\boldsymbol{D}\boldsymbol{A}^{\mathrm{T}}$ is bad for using $\mathrm{D} = \mathrm{diag(d)}$, while using $\mathrm{D} = \mathrm{sparse}(1:\mathrm{n}, 1:\mathrm{n}, \mathrm{d})$ is better.

Grading Criteria. Your code should be at least faster than the testing script, while the error should be smaller than the testing script.

4.1.2. Local Convergence Rate

The study of the rate of convergence is often the dominant criteria for selecting appropriate algorithms. In this lecture we only focus on the local behaviour of the method in a neighborhood of an optimal solution.

Definition 4.1 [Q_1 Factor] Restrict the attention to a convergent sequence $\{\boldsymbol{x}^k\}$ with limit $\boldsymbol{x}^*$. Define an error function $e_k = \|\boldsymbol{x}^k - \boldsymbol{x}^*\| \to 0$. The Q_1 factor of $\{\boldsymbol{x}^k\}$ is given as:

$$Q_1 = \limsup_{k\to\infty} \frac{e_{k+1}}{e_k}$$

Here we want to study the performance of e_k. In our case, we compare $\{e_k\}$ with the geometric prograssion

$$\beta^k, \quad k = 0, 1, \dots$$

- If there exists $\beta \in (0,1)$ such that

  $$Q_1 \le \beta,$$

  then we can show $e_k \le q\beta^k$ for some $q > 0$. In this case $\{e_k\}$ is said to be **Q-linear convergent**.

- If $Q_1 = 0$, then we say $\{e_k\}$ is **Q-super-linear convergent**

- If $Q_1 = 1$, then we say $\{e_k\}$ is **Q-sub-linear convergent**

R

1. Q-linear convergence is not always so good, e.g., $\beta = .999$ may require 20000 iterations to meet satisfaction, while $\beta = .1$ may only require 20.
2. Linear is always better than sublinear; super linear is always btetter than linear.

We have a faster type of convergence:

Definition 4.2 [Q_2 Factor]

$$Q_2 = \limsup_{k\to\infty} \frac{e^{k+1}}{(e^k)^2}$$

If $Q_2 = M < +\infty$,i.e., $e^{k+1} = O((e^k)^2)$, then $\{x^k\} \to x^*$ Q-quadratically.

Newton's method generally gives us the quadratic convergence.

4.1.3. Newton's Method

The newton's method requires to solve a non-linear system of equations $\nabla f(x) = 0$.

We don't know how to solve non-linear system in general. Fortunately, Newton gives us the remediation: in order to search for $\boldsymbol{d}$ to make $\nabla f(\boldsymbol{x}+\boldsymbol{d}) = 0$, do the linearization first.

$$\nabla f(\boldsymbol{x}+\boldsymbol{d}) \approx \nabla f(\boldsymbol{x}) + \langle \nabla^2 f(\boldsymbol{x}), \boldsymbol{d}\rangle = 0 \tag{4.1}$$

To get the optimal solution, it suffices to solve (4.1) for $\boldsymbol{d}$:

$$\boldsymbol{d} = -(\nabla^2 f(\boldsymbol{x}))^{-1}\nabla f(\boldsymbol{x}),$$

and hence update the solution to be $\boldsymbol{x} \leftarrow \boldsymbol{x} + \alpha\boldsymbol{d}$.

Interpretation. In order to minimize a strictly convex function $f(x)$, we find

$$f(\boldsymbol{x}+\boldsymbol{d}) \approx f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{d}\rangle + \frac{1}{2}\boldsymbol{d}^{\mathrm{T}}\nabla^2 f(\boldsymbol{x})\boldsymbol{d} := q(\boldsymbol{d})$$

It suffices to minimize $q(\boldsymbol{d})$:

$$\nabla q(\boldsymbol{d}) = 0 \Longleftrightarrow \nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x})\boldsymbol{d} = 0$$

How to guarntee $\boldsymbol{d}$ is the descent direction? Not necessarily we are able to do that. It becomes an art.

Convergence of Rate.

Proposition 4.1 Newton's method gurantees the Q-quadratic convergence.

Proof. Given a nonlinear system $F(x) = 0$, suppose the sequence $\{\boldsymbol{x}^k\}$ is generated by Newton with limit $\boldsymbol{x}^*$ and $F(\boldsymbol{x}^*) = 0$. By Newton's iteration,

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - [F'(\boldsymbol{x}^k)]^{-1}F(\boldsymbol{x}^k), \tag{4.2}$$

which follows that

$$\begin{aligned} \boldsymbol{x}^{k+1} - \boldsymbol{x}^* &= \boldsymbol{x}^k - \boldsymbol{x}^* - [F'(\boldsymbol{x}^k)]^{-1}\left(F(\boldsymbol{x}^k) - F(\boldsymbol{x}^*)\right) \\ &= [F'(\boldsymbol{x}^k)]^{-1}\left(F(\boldsymbol{x}^*) - F(\boldsymbol{x}^k) - F'(\boldsymbol{x}^k)(\boldsymbol{x}^* - \boldsymbol{x}^k)\right) \end{aligned} \tag{4.3}$$

Note that $F(\boldsymbol{x}^*) = F(\boldsymbol{x}^k) + F'(\boldsymbol{x}^k)(\boldsymbol{x}^* - \boldsymbol{x}^k) + O(\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2)$, which implies

$$\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\| \le \|[F'(x^k)]^{-1}\|O(\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2) = O(\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2).$$

■

R During the proof we assume two things:

1. Step-size is 1!
2. The limit exists.

- Hence, in practice, to implement Newton's method, **1** is the first choice; but gredient descent method is not.
- Newton's method is good for nice problems, i.e., the funciton is convex, and the inverse of gredient is easy to solve.
- In machine learning most time we implement the gredient descent method, since Newton's method is expensive for computing the inverse.

4.1.4. Tutorial: Introduction to Convexity

First we discuss some exercises:

1. Given a sequence of convex functions f_i, the maximum over all functions

$$\max\{f_1(x), f_2(x), \dots, f_n(x)\} := f(x)$$

is also convex. (Proof using Epi-Graph)

2. Given a sequence of convex functions f_i, the combination $\sum_i \lambda_i f_i$ is also convex
3. Composition function may not be convex, .e.g., if $h = x^2$ and $g = -x$, we find $g \circ h$ is concave.
4. However, if g is convex and h is affine, then $g \circ h$ is convex; if g, h is convex, and g is non-decreasing, then $g \circ h$ is convex.

Convexity Examples. Given the problem

$$\begin{aligned} \min \quad & \|\boldsymbol{x}\|_0 \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \end{aligned}$$

which is difficult to solve. Alternatively, we relax it and solve

$$\begin{aligned} \min \quad & \|\boldsymbol{x}\|_1 \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

Also, solving the problem

$$\begin{aligned} \min \quad & f(X) \\ \text{s.t.} \quad & \text{rank}(X) = k \end{aligned}$$

is hard, we relax it and solve

$$\begin{aligned} \min \quad & f(X) \\ \text{s.t.} \quad & \|X\|_* \leq k \end{aligned}$$

Announcement. Learn and implement these things by yourself:

- Luo's note: Lecture #4, P7,Nesterov's optimal 1st-order (also called acceleration) method

- Textbook P67-P72.

Chapter 5

Week5

5.1. Monday

5.1.1. Review

Optimality Condition. Given a general problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X \subseteq \mathbb{R}^n \\ & f \text{ is } \mathcal{C}^1 \text{ or } \mathcal{C}^2 \end{aligned}$$

One of the most important thing is the optimality condition.

- Unconstrainted: $X = \mathbb{R}^n$. (First order and second order)
- Constrainted:
  - 1st order necessary condition: Let $\boldsymbol{x}^*$ be a local minimum, then

$$\langle \nabla f(\boldsymbol{x}^*), (\boldsymbol{x} - \boldsymbol{x}^*) \rangle \geq 0, \forall x \in X$$

  - For convex function f, the above is also the sufficient condition, since

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) + \langle \nabla f(\boldsymbol{x}^*), (\boldsymbol{x} - \boldsymbol{x}^*) \rangle, \forall x, x^* \in X.$$

The optimality condition for constrainted problem is difficult to check. We want a more efficient way, which will be discussed later.

Iterative descent methods. For unconstraint problem, we consider the iterative descent methods:

$$x \leftarrow x - \alpha \cdot D \cdot \nabla f(x) \quad (D \succ 0)$$

- If $D = I$, it is the first order (gradient) method.
- If $D = \left(\nabla^2 f(x)\right)^{-1}$, it is the second order (Newton's) method

R Sometimes it is difficult to compute $\nabla^2 f(x)$. We can apply finite difference method to accurately approximate the Hessian matrix.

- If $D = (J^{\mathrm{T}} J)^{-1}$ with Jacobian matrix for nonlinear least squares problem, it is the Gauss-Newton method.
- Sometimes we apply rough method to approximate the Hessian matrix inaccurately, which is called the **Quasi-Newton** method. The most famous one is BFGS (L-BFGS).

There are more generalized iterative descent methods, such as the accelerated descent method tried in Assignment 3.

Reading materials. CG-conjugate gradient methods; and Nestorov's method (**optimal** accelerated method in **worse** case).

There is a method which is much faster than Nestorov's method in most cases:

$$\begin{aligned}
D^r &= \frac{1}{L} I + \frac{S^r (S^r)^{\mathrm{T}}}{(y^r)^{\mathrm{T}} (y^r)} \succ 0 \\
\alpha &= 1 \\
S^r &= x^{r+1} - x^r \\
y^r &= \nabla f(x^{r+1}) - \nabla f(x^r)
\end{aligned}$$

Step-size.

- Back-tracking with Amijo condition:

$$\text{Amijo condition} \quad f(x + \alpha d) \le f(x) + C_1 \alpha \langle \nabla f(x), d \rangle, 0 < C_1 < 1$$

- Wolfe condition for line search:

$$\text{Wolfe condition}\quad\begin{cases} f(\boldsymbol{x}+\alpha\boldsymbol{d}) \le f(\boldsymbol{x}) + C_1\alpha\langle\nabla f(\boldsymbol{x}),\boldsymbol{d}\rangle, & 0<C_1<1 \\ \langle\nabla f(\boldsymbol{x}+\alpha\boldsymbol{d}),\boldsymbol{d}\rangle \ge C_2\langle\nabla f(\boldsymbol{x}),\boldsymbol{d}\rangle, & 0<C_2<1 \end{cases}$$

  Define $h(\alpha)=h(\boldsymbol{x}+\alpha\boldsymbol{d})$, then the Wolfe condition is essentially

$$\begin{cases} h(\alpha)\le h(0)+C_1h'(0) \\ h'(\alpha)\ge C_2h'(0) \end{cases}$$

- Constant step-size: $\alpha^r \equiv \frac{1}{L}$ with L be the Lipschitz constant of $\nabla f(\boldsymbol{x})$.

- $\alpha^r \to 0$ with $\sum\alpha^r = +\infty$.

R Amijo condition guarantees that $f(\boldsymbol{x}^r)-f(\boldsymbol{x}^{r+1}) \ge -C_1\alpha^r\langle\nabla f(\boldsymbol{x}^r),\boldsymbol{d}^r\rangle$. Assume $f(\boldsymbol{x})>-\infty$, then

$$\alpha^r\langle\nabla f(\boldsymbol{x}^r),\boldsymbol{d}^r\rangle \to 0$$

We want $\nabla f(\boldsymbol{x}^r)\to 0$, which means your direction $\boldsymbol{d}^r$ should not be perpendicular to $\nabla f(\boldsymbol{x}^r)$ after some iterations. If choosing $\boldsymbol{d}^r=-\nabla f(\boldsymbol{x}^r)$, then $\alpha^r\|\nabla f(\boldsymbol{x}^r)\|^2\to 0$, which implies $\nabla f(\boldsymbol{x}^r)\to 0$.

Under reasonable conditions, applying first order condition we expect $\nabla f(\boldsymbol{x}^r)\to 0$. Is $\boldsymbol{x}^r$ always convergent? not necessarily.

Local convergence rate. The first order method has linear or sub-linear convergence rate; while the second order method has quadratic convergence rate.

Finite difference Method. Given $F(\boldsymbol{x}):\mathbb{R}^n\mapsto\mathbb{R}^n$, its Jacobian is given by:

$$F'(\boldsymbol{x}) = \begin{bmatrix} \nabla^{\mathrm{T}}F_1(\boldsymbol{x}) \\ \vdots \\ \nabla^{\mathrm{T}}F_n(\boldsymbol{x}) \end{bmatrix}$$

Its jth column is given by:

$$
\begin{aligned}
F'(\boldsymbol{x})\boldsymbol{e}_j := & \lim_{h\to 0} \frac{F(\boldsymbol{x}+h\boldsymbol{e}_j)-F(\boldsymbol{x})}{h} \\
\approx & \frac{F(\boldsymbol{x}+h\boldsymbol{e}_j)-F(\boldsymbol{x})}{h} \quad \text{for small } h
\end{aligned}
$$

where for $\varepsilon = 10^{-8}$,

$$h = \varepsilon \max\{1, |x_j|\} \text{sign}(x_j),$$

more-multiplying the term $\text{sign}(x_j)$ means we avoid subtract between $\boldsymbol{x}$ and $h\boldsymbol{e}_j$.

5.1.2. Existence of solution to Quadratic Programming

Theorem 5.1 Let $\{S^k\}$ be a sequence of non-empty closed nested sets. Suppose that all **asymptotic** sequences corresponding to asymptotic directions of $\{S^k\}$ are **retractive**, then $\bigcap_{k=0}^{\infty} S^k$ is **non-emtpy**.

Definition 5.1 Let $\{S^k\}$ be a sequence of non-empty closed nested sets. We say that a vector $\boldsymbol{d} \neq \boldsymbol{0}$ is an **asymptotic direction** of $\{S^k\}$ if there exists a sequence $\{\boldsymbol{x}^k\}$ such that

$$\boldsymbol{x}^k \in S^k, \quad \boldsymbol{x}^k \neq \boldsymbol{0}, \quad k = 0,1,2,\dots$$

and

$$\|\boldsymbol{x}^k\| \to \infty, \quad \frac{\boldsymbol{x}^k}{\|\boldsymbol{x}^k\|} \to \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$$

- $(\{\boldsymbol{x}^k\}, \boldsymbol{d})$ is said to be the asymptptoc pair of $\{S^k\}$
- $\{\boldsymbol{x}^k\}$ is said to be **retractive** if there exists a bounded sequence of positive numbers $\{\alpha^k\}$ and $\bar{k}$ such that

$$\boldsymbol{x}^k - \alpha^k \boldsymbol{d} \in S^k, \qquad \forall k \geq \bar{k}$$

Theorem 5.2 Let $\boldsymbol{Q}$ be a positive semi-definite symmetric $n \times n$ matrix, let $\boldsymbol{c}$ and $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r$ be vectors in $\mathbb{R}^n$, and let $b_1, \ldots, b_r$ be scalars. Assume that the optimal value of the problem

$$\begin{aligned} \min \quad & \boldsymbol{x}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} \\ \text{such that} \quad & \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} + b_j \leq 0, \qquad j = 1, 2, \ldots, r, \end{aligned} \tag{5.1}$$

is finite. Then the problem has **at least one optimal** solution.

Proof. Suppose f^* is the optimal solution. The feasible region is denoted by:

$$F = \{\boldsymbol{x} \mid a_j^{\mathrm{T}} \boldsymbol{x} + b_j \leq 0, j = 1, \ldots, r\}.$$

Set a decreasing sequence $\{\gamma^k\}$ with limit f^*, and set

$$S^k := \{\boldsymbol{x} \in F \mid \boldsymbol{x}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} \leq \gamma^k\}$$

Thus the set of optimal solutions is $\bigcap_{k=0}^{\infty} S^k$. It suffices to show that all asymptotic seqeuences corresponding to asymptotic directions are **retractive**.

Asymptotic Directions are essentially Boundary Directions. For fiexed asymptotic pair $(\{\boldsymbol{x}^k\}, \boldsymbol{d})$, we claim that

$$\boldsymbol{Q}\boldsymbol{d} = 0, \langle \boldsymbol{c}, \boldsymbol{d} \rangle \leq 0 \tag{5.2}$$

$$\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle \leq 0, j = 1, 2, \ldots, r \tag{5.3}$$

For first equaility, define $\boldsymbol{d}^k = \frac{\boldsymbol{x}^k}{\|\boldsymbol{x}^k\|}$. Since $\boldsymbol{x}^k \in S^k$, we have

$$(\boldsymbol{d}^k)^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{d}^k + \frac{\langle \boldsymbol{c}, \boldsymbol{d}^k \rangle}{\|\boldsymbol{x}^k\|} \leq \frac{\gamma^k}{\|\boldsymbol{x}^k\|^2} \tag{5.4}$$

Taking $k \to \infty$, we imply $(\boldsymbol{d})^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{d} \leq 0$, and therefore $\boldsymbol{Q}\boldsymbol{d} = 0$ as $\boldsymbol{Q} \succeq 0$.

Due to (5.4) and the semi-definiteness of $\boldsymbol{Q}$, we have

$$\langle \boldsymbol{c}, \boldsymbol{d}^k \rangle \leq \|\boldsymbol{x}^k\| (\boldsymbol{d}^k)^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{d}^k + \langle \boldsymbol{c}, \boldsymbol{d}^k \rangle \leq \frac{\gamma^k}{\|\boldsymbol{x}^k\|}$$

Taking $k \to \infty$, we imply $\langle \boldsymbol{c}, \boldsymbol{d}^k \rangle \le 0$. Similarly, $\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle \le 0$ since $\langle \boldsymbol{a}_j, \boldsymbol{d}^k \rangle \le -\frac{b_j}{\|\boldsymbol{x}^k\|}$.

Finiteness of optimal value. Next we show $\langle \boldsymbol{c}, \boldsymbol{d} \rangle = 0$. For a feasible vector $\bar{\boldsymbol{x}}$, consider $\tilde{\boldsymbol{x}} := \bar{\boldsymbol{x}} + m\boldsymbol{d}$ for any positive m, which is also feasible as $\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle \le 0$. Then checking the function evaluated at $\tilde{\boldsymbol{x}}$:

$$f^* \le (\tilde{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{Q} (\tilde{\boldsymbol{x}}) + \langle \boldsymbol{c}, \tilde{\boldsymbol{x}} \rangle = (\bar{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{Q} \bar{\boldsymbol{x}} + \langle \boldsymbol{c}, \bar{\boldsymbol{x}} \rangle + m \langle \boldsymbol{c}, \boldsymbol{d} \rangle$$

As f^* is finite, $\langle \boldsymbol{c}, \boldsymbol{d} \rangle \ge 0$, i.e., $\langle \boldsymbol{c}, \boldsymbol{d} \rangle = 0$.

As a result, for fixed $\boldsymbol{x}^k$, the function evaluated at $\boldsymbol{x}^k - \alpha \boldsymbol{d}$ satisfies

$$(\boldsymbol{x}^k - \alpha \boldsymbol{d})^{\mathrm{T}} \boldsymbol{Q} (\boldsymbol{x}^k - \alpha \boldsymbol{d}) + \langle \boldsymbol{c}, \boldsymbol{x}^k - \alpha \boldsymbol{d} \rangle = (\boldsymbol{x}^k)^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x}^k + \langle \boldsymbol{c}, \boldsymbol{x}^k \rangle \le \gamma^k, \forall \alpha, k,$$

Feasibleness of $\boldsymbol{x}^k - \alpha \boldsymbol{d}$. It suffices to choose $\alpha > 0$ to let $\boldsymbol{x}^k - \alpha \boldsymbol{d} \in F$ for sufficiently large k, i.e.,

$$\langle \boldsymbol{a}_j, \boldsymbol{x}^k - \alpha \boldsymbol{d} \rangle + b_j \le 0, \quad j = 1, \ldots, r$$

- This is true for $\forall \alpha > 0$ if $\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle = 0$
- Otherwise, suppose $\langle \boldsymbol{a}_j, \boldsymbol{d} \rangle < -\varepsilon$. Thus $\langle \boldsymbol{a}_j, \boldsymbol{d}^k \rangle < -\varepsilon$ for sufficiently large k, i.e., $\langle \boldsymbol{a}_j, \boldsymbol{x}^k \rangle \le -\varepsilon \|\boldsymbol{x}^k\|$. Combining the unboundness of $\{\boldsymbol{x}^k\}$, we imply

$$\langle \boldsymbol{a}_j, \boldsymbol{x}^k - \alpha \boldsymbol{d} \rangle + b_j \le -\varepsilon \|\boldsymbol{x}^k\| - \alpha \langle \boldsymbol{a}_j, \boldsymbol{d} \rangle + b_j < 0$$

■

5.2. Wednesday

Announcement. You need to study the textbook by yourself. Those materials will be tested in the mid-term.

5.2.1. Comments about Newton's Method

Newton's Method may not necessarily have quadratic rate of convergence. One counter-example is the optimization on the function x^2, where we have the iteration

$$x^{k+1} = x^k - \frac{1}{2x^k}(x^k)^2 = \frac{x^k}{2},$$

which has linear rate of convergence.

Theorem 5.3 — Sufficient Condition for quadratic convergence of Newton's method. To minimize the function $f(x)$ with gradient function $F(x) = \nabla f(x)$, the Newton's method iteration gives

$$x^{k+1} = x^k - \left[F'(x^k)\right]^{-1} F(x^k),$$

the quadratic (local) rate of convergence is guaranteed if the following conditions hold:

1. there exists x^* such that $F(x^*) = 0$
2. $[F'(x^*)]^{-1}$ exists
3. F is **Lipschitz continuous** near x^*.

For example, to minimize the function $x^2 - x$, the iteration gives the quadratic convergence:

$$x^{k+1} - 1 = \frac{1}{2}(x^k - 1)(1 - \frac{1}{x^k}) = O((x^k - 1)^2)x^{k+1} = x^k - (2x^k)$$

5.2.2. Constant Step-Size Analysis

For the unconstraint minimization

$$\min_{x\in\mathbb{R}^n} f(x),$$

with convex function $f \in \mathcal{C}^1$, our iteration for constant step-size is given by

$$x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k).$$

Now we are interested in the convergence rate of this choice of step-size.

Theorem 5.4 — Convergence Rate for invariant step-size. Given a convex function $f \in \mathcal{C}^2$ with Lipschitz gradient, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\|^2,$$

the iteration $x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k)$ with local minimum point x^* gives sub-linear convergence rate, i.e.,

$$f(x^k) - f(x^*) \le \frac{L\|x^0 - x^*\|}{k+1}$$

First prove a simple proposition:

Proposition 5.1 A convex function $f \in \mathcal{C}^2$ with Lipschitz gradient constant L has a bounded Hessian matrix:

$$\nabla^2 f(x) \preceq L\boldsymbol{I}$$

Proof for proposition(5.1). Otherwise $\exists x_0, v$ such that

$$\|\nabla^2 f(x_0)v\| > L\|v\|.$$

Thus we apply taylor expansion near x_0 for $x = x_0 + v$:

$$\nabla f(x) = \nabla f(x_0) + \nabla^2 f(x_0)(x-x_0) + o(1)(x-x_0)$$

It follows that

$$\|\nabla f(x) - \nabla f(x_0)\| = \|\nabla^2 f(x_0)(x - x_0) + o(1)(x - x_0)\| \le L\|x - x_0\|$$

Thus for sufficiently small v, we have $\|\nabla f(x) - \nabla f(x_0)\| \le L\|x - x_0\|$, which is a contradiction. ■

Step 1: Apply Lipschitz condition. We do the taylor expansion of x^k for the point x^{k+1}:

$$\begin{aligned} f(x^{k+1}) &= f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{1}{2}(x^{k+1} - x^k)^{\mathrm{T}} \nabla^2 f(x^k + \tau(x^{k+1-x^k}))(x^{k+1} - x^k) \\ &\le f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2 \\ &= f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|^2 \end{aligned}$$

implying that

$$\|\nabla f(x^k)\|^2 \le 2L[f(x^k) - f(x^{k+1})]$$

and therefore

$$\sum_{k=0}^{r} \|\nabla f(x^k)\|^2 \le \sum_{k=0}^{\infty} \|\nabla f(x^k)\|^2 \le 2L[f(x^0) - f(x^*)] \le L^2\|x^0 - x^*\|^2 \tag{5.5}$$

Step 2: Applying Convexity of f. By the convexity of f and the bound on its gradient, we can estimate the total error $\sum_{k=0}^{r}(f(x^k) - f(x^*))$:

$$\begin{aligned} \|x^{k+1} - x^*\|^2 &= \|x^k - \frac{1}{L}\nabla f(x^k) - x^*\|^2 \\ &= \|x^k - x^*\|^2 - \frac{2}{L}\langle \nabla f(x^k), x^k - x^* \rangle + \frac{1}{L^2}\|\nabla f(x^k)\|^2 \\ &\le \|x^k - x^*\|^2 - \frac{2}{L}(f(x^k) - f(x^*)) + \frac{1}{L^2}\|\nabla f(x^k)\|^2 \end{aligned}$$

which implies

$$\begin{aligned}\sum_{k=0}^{r} f(x^k) - f(x^*) &\le \frac{L}{2}\sum_{k=0}^{r}\left[\|x^k - x^*\| - \|x^{k+1} - x^*\|^2\right] + \frac{1}{2L}\sum_{k=0}^{r}\|\nabla f(x^k)\|^2 \\ &\le \frac{L}{2}\left[\|x^0 - x^*\|^2 - \|x^{r+1} - x^*\|^2\right] + \frac{L}{2}\|x^0 - x^*\|^2 \\ &\le L\|x^0 - x^*\|^2\end{aligned}$$

Step 3: applying monotonicity of $f(x^k) - f(x^*)$. By the monotonicity of $f(x^k) - f(x^*)$,

$$\begin{aligned}f(x^r) - f(x^*) &\le \frac{1}{r+1}\sum_{k=0}^{r} f(x^k) - f(x^*) \\ &\le \frac{L\|x^0 - x^*\|^2}{r+1}\end{aligned}$$

R The convergence rate for this method is **super-linear**. This upper bound is order-tight, i.e., we can find one example satisfying the equality.

However, the constant step-size method can be faster if the given function f is **strongly convex**.

Proposition 5.2 The iteration $x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k)$ for a strongly conex function $f \in \mathcal{C}^2$ with Lipschitz gradient gives linear convergence rate, i.e.,

$$f(x^k) - f(x^*) = O(\rho^k),$$

with $\rho = 1 - \frac{\sigma}{L}$.

Step 1: Lipschitz gradient gives upper bound on $f(x^{k+1}) - f(x^k)$. Applying Taylor Expansion,

$$\begin{aligned}f(x^{k+1}) - f(x^k) &= \langle \nabla f(x^k), x^{k+1} - x^k\rangle + \frac{1}{2}(x^{k+1} - x^k)^{\mathrm{T}}\nabla^2 f(x^k + \tau(x^{k+1} - x^k))(x^{k+1} - x^k) \\ &\le \langle \nabla f(x^k), x^{k+1} - x^k\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2\end{aligned}$$

Substituting $x^{k+1} - x^k = -\frac{1}{L}\nabla f(x^k)$, we derive:

$$f(x^{k+1}) - f(x^k) \le -\frac{1}{2L}\|\nabla f(x^k)\|^2$$

Step 2: Stronly convex gives upper bound on $f(x^*) - f(x)$. For $\forall x, y$, we have

$$\begin{aligned} f(y) &= f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}(y-x)^{\mathrm{T}}\nabla^2 f(z)(y-x) \\ &\ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\sigma}{2}\|y-x\|^2 \end{aligned}$$

where $\min_i \lambda_i(\nabla^2 f(x)) \ge \sigma > 0, \forall$. Minimizing both sides in terms of y gives

$$f(x^*) \ge f(x) - \frac{1}{2\sigma}\|\nabla f(x)\|^2$$

Step 3: Re-arrange bounds on $f(x^{k+1}) - f(x^*)$. Applying those bounds, we derive

$$\begin{aligned} f(x^{k+1}) - f(x^*) &\le f(x^k) - f(x^*) - \frac{1}{2L}\|\nabla f(x^k)\|^2 \\ &\le f(x^k) - f(x^*) - \frac{1}{2L}\|\nabla f(x^k)\|^2 \\ &\le f(x^k) - f(x^*) - \frac{\sigma}{L}[f(x^k) - f(x^*)] \\ &= (1 - \frac{\sigma}{L})[f(x^k) - f(x^*)] \end{aligned}$$

Applying this trick recursively,

$$f(x^k) - f(x^*) \le (1 - \frac{\sigma}{L})^k[f(x^0) - f(x^*)]$$

Thus we have shown the linear convergence rate $f(x^k) - f(x^*) = O(\rho^k)$ with $\rho = 1 - \frac{\sigma}{L}$.

Reading Assignment: .

1. Nesterov's Introductory Courses on Convex Programming
2. Prof. Luo's note #4.

Chapter 6

Week6

6.1. Monday

6.1.1. Announcement

The new homework is assgined, which is due on Friday. The programming task is relatively easy this week, but requires research on some paper.

6.1.2. Introduction to Quasi-Newton Method

$$x^{k+1} \leftarrow x^k - \alpha_k D_k \nabla f(x^k)$$

Here D^k is the approximation of the inverse of the Hessian matrix. Our initial guess is usually $D_0 = \tau I$, and then

$$D^{k+1} \leftarrow \text{Update}(D^k, s^k, y^k), \tag{6.1}$$

with $s^k = x^{k+1} - x^k$, $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$.

QN-Equation. Note that for any x, x' in the domain,

$$\nabla f(x') - \nabla f(x) = \left[\int_0^1 \nabla^2 f(x + \tau(x' - x))\,\mathrm{d}\tau\right](x' - x) \approx \nabla^2 f(x)(x' - x)$$

Hence, we assume that the update rule (6.2) has to satisfy the Quasi-Newton equation:

$$D^{k+1} s^k = y^k$$

Least-Change. Moreover, we want to use previous information, i.e.,

$$\begin{aligned} \min \quad & \|D^k - D^k\|_w \\ \text{such that} \quad & Ds^k = y^k \\ & D \succ 0 \end{aligned}$$

Prof.YZ did his phd research on this topic.

The convergence rate of QN-equation could be super-linear.

6.1.3. Constrainted Optimization Problem

Given the un-reduency problem

$$\begin{aligned} \min \quad & f(x) \\ \text{such that} \quad & Ax = b, \quad A \in \mathbb{R}^{m\times n}, m < n \end{aligned} \tag{6.2}$$

we can convert this problem into unconstrainted. Let $x = A^{\mathrm{T}}u + B^{\mathrm{T}}v$ with:

$$AB^{\mathrm{T}} = 0; \qquad \begin{bmatrix} A \\ B \end{bmatrix}_{m+(n-m)} \quad \text{is non-singular}$$

It follows that

$$AA^{\mathrm{T}}u = b \implies u^* = (AA^{\mathrm{T}})^{-1}b,$$

and therefore every $x = A^{\mathrm{T}}u^* + B^{\mathrm{T}}v$ satisfies the constraint. It suffices to solve

$$\min_v f(A^{\mathrm{T}}u^* + B^{\mathrm{T}}v) := h(v)$$

Thus the problem becomes unconstraint.

Optimality Condition for (6.2). The necessary optimality condition is

$$\nabla h(v) = B\nabla f(x) = 0, \quad \text{for } Ax = b.$$

which basically says that $\nabla f(x) \in \mathcal{R}(A^{\mathrm{T}})$ for x such that $Ax = b$, i.e.,

$$\begin{cases} \nabla f(x) = A^{\mathrm{T}}\lambda \\ \quad Ax = b \end{cases} \qquad \text{1st order necessary condition for (6.2)}$$

Therefore, optimization problem (6.2) is essentially a linear equality constrainted optimization.

Optimality condition for general function. For a more general problem

$$\begin{aligned} &\min && f(x) \\ &\text{such that} && C(x) = 0 \end{aligned} \tag{6.3}$$

The necessary optimality condition is

$$\begin{cases} \nabla f(x) = \nabla C(x)\lambda \\ \quad C(x) = 0 \end{cases}$$

Beside from the necessary optimality condition, the problem should meet the Constraint Qualification (CQ), i.e., gradient should be linear independent. We will discuss this issue in the future.

6.1.4. Announcement on Assignment

1. For phd student, please do this problem: Exercise 2.2.1
2. In the programming project, we are required to sovle the problem

$$\min_{X \in \mathbb{R}^{n \times k}} \|A - XX^{\mathrm{T}}\|_F^2$$

   The Gauss-Newton method to this problem is even faster than SVD or something else.
3. The second programming task is about Linear Programming, please go to tutorial

to have idea about that:

$$\begin{aligned} \min \quad & c^{\mathrm{T}} x \\ \text{such that} \quad & Ax = b, A \in \mathbb{R}^{m\times n}, m < n \\ & x \geq 0 \end{aligned} \tag{6.4}$$

We can approximate this problem by

$$\begin{aligned} \min \quad & c^{\mathrm{T}} x \underbrace{-\mu \sum \ln x_i}_{\text{barrier}} := f(x) \\ \text{such that} \quad & Ax = b, A \in \mathbb{R}^{m\times n}, m < n \end{aligned}$$

with $\mu > 0$. As $\mu \to 0$, this problem is the approximation of (6.4). As long as it satisfies the necessary optimality condition, we get the minimum point:

$$\begin{cases} c - \mu \begin{pmatrix} 1/x_1 \\ \vdots \\ 1/x_n \end{pmatrix} = A^{\mathrm{T}} y \\ \qquad Ax = b \end{cases}$$

Set $z = \mu./x$, it suffices to solve

$$\begin{cases} c - z = A^{\mathrm{T}} y \\ Ax = b \\ x_i z_i = u, \quad i = 1, 2, \dots, n \end{cases} \iff F_\mu(x, y, z) = \begin{bmatrix} A^{\mathrm{T}} y + z - c \\ Ax - b \\ x \circ z - u \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Optimal Algorithm for optimization. Given a collection of problem, the optimality of algorithm is defined as the performance of such for the worse case.

The gradient descent method gives

$$\begin{aligned} x^{k+1} &= x^0 - \alpha_0 g_0 - \alpha_1 g_1 - \cdots - \alpha_k g_k \\ &= x^0 - \operatorname{span}\{g_0, g_1, \dots, g_n\} \end{aligned}$$

The optimal accelerated method's performance order is $O(\frac{1}{k^2})$.

6.1.5. Introduction to Stochastic optimization

Consider the Stochastic optimization problem

$$\min_x \mathbb{E}_\xi f(x,\xi)$$

When the distribution of ξ is unknown, we take samples and then solve

$$\min_x \frac{1}{m}\sum_{i=1}^m f(x,\xi_i) \Longleftrightarrow \min \sum_{i=1}^m f_i(x,\xi)$$

If $f_i(x) := \|g_i(x) - y_\xi\|^2$, the origin problem becomes the sum of least squares.

We apply gradient descent method to solve this problem:

$$\nabla \frac{1}{2} f(x) = \sum_{i=1}^m \nabla g_i(x)\,[g_i(x) - y_i]$$

We will explain this application in detail in the future.

6.2. Tutorial: Monday

This tutorial aims to cover topics in Assignment 4 and previous assignments.

6.2.1. LP Problem

$$\begin{array}{ll} \min & c^{\mathrm{T}}x \\ \text{such that} & Ax = b \\ & x \geq 0 \end{array}$$

It suffices to solve the nonlinear system:

$$F(x,y,z) = \begin{pmatrix} A^{\mathrm{T}}y + z - c \\ Ax - b \\ x \circ z - \mu \end{pmatrix} = 0, \quad x,z \geq 0$$

Or equivalently,

$$\begin{cases} F'(x,y,z)\begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} c - A^{\mathrm{T}}y - z \\ b - Ax \\ \mu - x \circ z \end{pmatrix} \\ F' = \begin{pmatrix} A & 0 & 0 \\ 0 & A^{\mathrm{T}} & I \\ z & 0 & x \end{pmatrix} \\ z := \operatorname{diag}(z_1, \ldots, z_n) \\ x := \operatorname{diag}(x_1, \ldots, x_n) \end{cases}$$

We solve this system by Gaussian Elimination:

$$\begin{pmatrix} A^{\mathrm{T}} & I & 0 \\ 0 & 0 & A \\ 0 & X & Z \end{pmatrix} \begin{pmatrix} dy \\ dz \\ dz \end{pmatrix} = \begin{pmatrix} r_d \\ r_p \\ r_c \end{pmatrix} \Longrightarrow \left[\begin{array}{ccc|c} A\frac{X}{Z}A^{\mathrm{T}} & 0 & 0 & A\frac{Xr_d - r_c}{Z} + r_p \\ 0 & 0 & A & r_p \\ 0 & X & Z & r_c \end{array}\right]$$

and therefore

$$\begin{pmatrix} A\frac{X}{Z}A^{\mathrm{T}} & 0 & 0 \\ 0 & 0 & A \\ 0 & X & Z \end{pmatrix} \begin{pmatrix} dy \\ dz \\ dx \end{pmatrix} = \begin{pmatrix} A\frac{Xr_d - r_c}{Z} + r_p \\ r_p \\ r_c \end{pmatrix}$$

Hint: Use $spdiags(X./Z)$ to accelerate computation.

6.2.2. Gauss-Newton Method

In second task, we are required to minimize $\|XX^{\mathrm{T}} - A\|^2$, during calculation, we need to arrange wisely to determine which X, Y, A, Z to compute first.

6.2.3. Introduction to KKT and CQ

Consider the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{such that} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

The KKT is essentially first-order necessary conditions for it.

$$\begin{aligned} \text{Dual Feasibility} \quad & \nabla_x(f(x) + \mu g(x) + \gamma h(x)) = 0, \quad \mu \geq 0 \\ \text{Primal Feasibility} \quad & g(x) \leq 0 \\ & h(x) = 0 \\ & z \cdot g(x) = 0, \quad z \geq 0 \end{aligned}$$

However, the KKT condition does not hold for optimal solution necessarily. Furthermore, it must meet the Constraint Qualification.

Example 6.1

$$\begin{aligned} \min \quad & x \\ \text{such that} \quad & x^2 \leq 0 \end{aligned}$$

If we use KKT condition, then $\nabla_x(x + \lambda x^2) = 1 + 2\lambda x = 1$ at $x = 0$. This contradction is because we haven't tested the Constraint Qualification.

There are two kinds of CQ, either one is satisfied is OK.

1. Slaters Condition: for convex problem, if exists x such that $g(x) < 0$ and $h(x) = 0$.
2. Linear Independence Constraint Qualification:

$$\nabla g_x, \qquad \nabla h$$

   are linear independent.

6.3. Wednesday

6.3.1. Review

1. To minimize the term $\|A - XX^{\mathrm{T}}\|_F^2$, just implement the algorithm in the paper.
2. The next question is to solve the linear system by Newton's method

$$F(x+dx,y+dy,z+dz) = \begin{bmatrix} A^{\mathrm{T}}y+z-c \\ Ax-b \\ x\circ z-\mu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \implies F_\mu(x,y,z) + F'_\mu(x,y,z)(dx,dy,dz)^{\mathrm{T}} = 0$$

Stochastic Gradient Method. The objective function is

$$f(x) = \frac{1}{2}(x-c_1)^2 + \frac{1}{2}(x-c_2)^2,$$

where $x^* = \frac{c_1+c_2}{2}$.

$$\begin{cases} x^r(2) = x^r(1) - \alpha(x^r(1)-c_1) = (1-\alpha)x^r(1) + \alpha c_1 \\ x^{r+1}(1) = x^r(2) - \alpha(x^r(2)-c_2) = (1-\alpha)x^r(2) + \alpha c_2 \end{cases}$$

After computation, we derive $x^{r+1}(1) = (1-\alpha)^2 x^r(1) + (1-\alpha)\alpha c_1 + \alpha c_2$.

How to guarantee convergence? We have generated two sequences

$$\{x^{r-1}(1).x^r(1),x^{r+1}(1),\dots\} \to x_\alpha(1), \qquad \{x^{r-1}(2).x^r(2),x^{r+1}(2),\dots\} \to x_\alpha(2),$$

We can solve for $x_\alpha(1)$ and $x_\alpha(2)$:

$$x_\alpha(1) = (1-\alpha)^2 x_\alpha(1) + (1-\alpha)\alpha c_1 + \alpha c_2 \implies x_\alpha(1) = \frac{(1-\alpha)c_1 + c_2}{2-\alpha}$$
$$x_\alpha(2) = \frac{(1-\alpha)c_2 + c_1}{2-\alpha}$$

For fixed α we can see that those two limits are not the same. From the formula above we can see that as $\alpha \to 0$, $x_\alpha(1) \to \frac{c_1+c_2}{2}$ and $x_\alpha(2) \to \frac{c_1+c_2}{2}$.

R Note that α^r cannot converge too slow or too fast, the right setting is $\alpha^r = \frac{1}{r}$.

6.3.2. Dual-Primal of LP

Given the primal linear programming problem

$$\begin{aligned} \min \quad & c^{\mathrm{T}} x \\ \text{such that} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Recap. Recall how to solve the general optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{such that} \quad & Ax = b \end{aligned}$$

The 1st order necessary condition is

$$\begin{cases} \nabla f(x) = A^{\mathrm{T}} \lambda \\ \quad Ax = b \end{cases}$$

Or rewrite it into a system $F(x, \lambda) = 0$, which can be solved by Newton's method.

Barrier Term. But the LP problem has the inequality $x \geq 0$, we add a log barrier $B_{\mu}(x) = -\mu \sum \log x_i$ with $\mu > 0$ to sovle this problem:

$$\begin{aligned} \min \quad & c^{\mathrm{T}} + B_{\mu}(x) \\ \text{such that} \quad & Ax = b \end{aligned}$$

Solving this problem is equivalent to solving the system $F_{\mu}(x, y, z) = 0$ in assignment 4. As $\mu \to 0$, it suffices to solve the original problem, i.e., $(x(\mu), y(\mu), z(\mu)) \to (x^*, y^*, z^*)$.

R There is a faster method. Every iteration just approximate μ, and after new iteration, μ is updated.

1. Change step-size to make convergence rate faster
2. Appreciate the use of dense matrix A.

Dual Problem of LP.

1. We form the **Lagrange function** of origin LP:

$$L(x,y) = c^{\mathrm{T}}x - y^{\mathrm{T}}(Ax - b)$$

(We take care of the constraint $x \geq 0$ in next step).

2. The **dual function** is a function of y:

$$\begin{aligned} Q(y) &= \inf_{x\geq 0} L(x,y) \\ &= \inf_{x\geq 0} (c - A^{\mathrm{T}}y)^{\mathrm{T}}x + b^{\mathrm{T}}y \\ &= \begin{cases} b^{\mathrm{T}}y, & c - A^{\mathrm{T}}y \geq 0 \\ -\infty & \text{otherwise} \end{cases} \end{aligned}$$

3. The **dual problem** is given by:

$$\begin{aligned} &\max_{y\in\mathbb{R}^m} && Q(y) = \max_y b^{\mathrm{T}}y \\ &\text{such that} && A^{\mathrm{T}}y \leq c \end{aligned}$$

The constraint is because that violeting the constraint makes the objective goes to minus infinite.

Then we introduce a new variable $z \geq 0$:

$$\begin{aligned} &\max && b^{\mathrm{T}}y \\ &\text{such that} && A^{\mathrm{T}}y + z = c \qquad \text{Dual} \\ & && z \geq 0 \end{aligned}$$

- The primal and dual problem of LP is that they share the same (A,b,c)

- Let x be P-feasible, (y,z) be D-feasible, then $c^{\mathrm{T}}x - b^{\mathrm{T}}y \geq 0$ (**weak duality**):

$$\begin{aligned} c^{\mathrm{T}}x - b^{\mathrm{T}}y &= x^{\mathrm{T}}c - x^{\mathrm{T}}A^{\mathrm{T}}y \\ &= x^{\mathrm{T}}(c - A^{\mathrm{T}}y) = x^{\mathrm{T}}z \geq 0 \end{aligned}$$

- For linear programming, the optimal solution satisfies $c^{\mathrm{T}}x^* = b^{\mathrm{T}}y^*$, i.e., $x \circ z = 0$.

(**strong duality**)

- The optimality condition for the primial-dual problem is:

$$\begin{aligned} A^{\mathrm{T}}y + z - c = 0, &\quad z \geq 0 \\ Ax - b = 0, &\quad x \geq 0 \\ x \circ z = 0 & \end{aligned}$$

  which means that the optimal solution of LP must satisfy both constraints from primal and dual. (The assignment 4 makes $x \circ z - \mu = 0$ and let $\mu \to 0$.)

Chapter 7

Week7

7.1. Monday

7.1.1. Announcement

- The new homework is assigned. You are required to write a code for linear programming. The detail will be announced in tutorial.

- The mid-term will be some written problem and computation porblem. We will cover contents in Chapter 1,2,3,4 in the textbook.

7.1.2. Recap about linear programming

The standard form of linear programming is

$$\begin{array}{ll} \min & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{such that} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{array}$$

Define the Lagrange function

$$\begin{aligned} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) &= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) - \boldsymbol{z}^{\mathrm{T}}\boldsymbol{x} \\ &= (\boldsymbol{c} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{z})^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} \end{aligned}$$

From this Lagrange function we define a dual function

$$Q(\boldsymbol{y},\boldsymbol{z}) = \inf_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \begin{cases} \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}, & \text{if } \boldsymbol{c} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{z} = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

For any feasible $\boldsymbol{x}$ and $\boldsymbol{z} \geq 0$, we always have

$$Q(\boldsymbol{y},\boldsymbol{x}) \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{z}^{\mathrm{T}}\boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

Moreover,

$$\max_{\boldsymbol{A}^{\mathrm{T}}y+\boldsymbol{z}=\boldsymbol{c},\boldsymbol{z}\geq 0} \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y} = \sup_{\boldsymbol{z}\geq 0} Q(\boldsymbol{y},\boldsymbol{z}) \leq \min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\boldsymbol{x}\geq 0} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

- The weak duality says that the optimal dual objective is always no more than the optimal primal objective.

- The strong duality says that the optimal dual objective is equal to the optimal primal objective.

Theorem 7.1 The LP optimality condition is given by:

1. Primal-Feasibility:
$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq 0$$
2. Dual-Feasibility:
$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c}, \boldsymbol{z} \geq 0$$
3. Complementarity/Strong-Duality:
$$\boldsymbol{x} \circ \boldsymbol{z} = 0$$

R Hence, the LP is essentially solving the linear systems. The Assignment 5 aims to solve the linear programming problem via primal-dual algorithm. Furthermore, it is called the primal-dual interior-point method (pdipm). The idea of interior-point method is to set the barrier:

$$\boldsymbol{x} \circ \boldsymbol{z} = u * \mathbf{1},$$

and with $u \to 0$, $\boldsymbol{x} \circ \boldsymbol{z} = 0$.

Something about A5. During the implementation, you need to solve the system

$$\underbrace{(A(Z^{-1}X)A^{\mathrm{T}})}_{\boldsymbol{B}} dy = rh_s$$

Given that the A is sparse, we call the command chol to decompose $\boldsymbol{M} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$ with $\boldsymbol{R}$ to be upper triangular. Then we obtain the solution

$$dy = R/(R'/rhs)$$

However, it is still time-consuming. We can accelerate it by re-arrange the order of the linear system by the command symamd. The detailed handle discussing the advantage for re-ordering is attached in assignment.

Incremental Problem. When solving the least squares problem

$$\min \frac{1}{2}\sum_{i=1}^{m}(\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{b}_i)^2 = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2,$$

the optimal analytic solution is given by $\boldsymbol{x}^* = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$. Let's try the numerical method:

$$\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k - \alpha^k \boldsymbol{g}^1 - \alpha^k g^2 - \cdots - \alpha^k \boldsymbol{g}^m,$$

where $\alpha^k = \frac{\theta}{k}$. Prof.YZ's α is greater than 10. Although we have the analytic solution, it may not necessarily better than the numerical one since it will perturbed by the noise.

7.1.3. Optimization over convex set

For the optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{such that} \quad & x \in X \end{aligned} \tag{7.1}$$

with X to be convex, the necessary conditon for optimality is

$$\nabla^{\mathrm{T}} f(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*) \geq 0, \quad \forall x \in X$$

If f is convex, then it is indeed the necessary and sufficient condition.

Example. For $X = \{x \mid x \geq 0\}$, we find

$$(\nabla f(\boldsymbol{x}^*))_i = \begin{cases} \dfrac{\partial f(\boldsymbol{x}^*)}{\partial x_i} \geq 0, & x_i^* = 0 \\ \dfrac{\partial f(\boldsymbol{x}^*)}{\partial x_i} = 0, & x_i^* > 0 \end{cases}$$

Project. Given $z \in \mathbb{R}^n$, the least square problem

$$\begin{aligned} \min \quad & \|\boldsymbol{x} - \boldsymbol{z}\|_2 \\ & \boldsymbol{x} \in X \end{aligned}$$

is essentially the projection problem:

$$[\boldsymbol{z}]^+ = \boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in X} \|\boldsymbol{x} - \boldsymbol{z}\|$$

Proposition 7.1 — Non-expansive.

$$\|[\boldsymbol{z}_1]^+ - [\boldsymbol{z}_2]^+\| \leq \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|$$

with X to be a convex set.

Projection-based optimality condition. Applying projection property, we derive alternative optimality condition for (7.1):

Proposition 7.2 $\boldsymbol{x}^*$ is a stationary point iff

$$\boldsymbol{x}^* = [\boldsymbol{x}^* - \alpha \nabla f(\boldsymbol{x}^*)]+,$$

for $\forall \alpha > 0$

7.2. Wednesday

7.2.1. Motivation

Given the optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{such that} \quad & x \in X \text{ is convex} \end{aligned} \tag{7.2}$$

Recall that The necessary optimality condition for (7.2) is

$$\langle \nabla f(x^*), (x - x^*) \rangle \geq 0 \quad \forall x \in X$$

R

- If we know x^* is exactly interior to the convex set, we have $\nabla f(x^*) = 0$;
- If x^* is on the boundary of the convex set, then the gradient should be perpendicular to the convex set, i.e., $\nabla f(x^*)$ should be orthogonal to the tangent line at $x = x^*$.
- In particular, if $x^* \in \partial X$ with X to be an affine, then

$$\nabla f(x^*) \perp X.$$

Motivation of Feasible direction method. Recall that the idea of gradient descent is to find a direction $\boldsymbol{d}^k$ at the point $\boldsymbol{x}^k$ such that $\langle \nabla f(x^k), \boldsymbol{d}^k \rangle < 0$ and therefore $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{d}^k$; However, sometimes in constraint optimization problem (7.2) the update position $\boldsymbol{x}^{k+1}$ may not be necessarily feasible. To handle such a barrier, we can try two algorithms: conditional gradient and projected gradient descent method. Before it let's study some preliminaries about projection.

7.2.2. Convex Projections

The **projection** step aims to solve the optimization problem defined as follows:

Definition 7.1 [Projection] for given z, define the projection as

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|x-z\|_2^2 \\ \text{such that} \quad & x \in X \text{ is convex and closed,} \end{aligned} \tag{7.3}$$

where $[z]^+ := \arg\min_x \frac{1}{2}\|x-z\|_2^2$ is called the **projection operator**. Sometimes we also write $\arg\min_x \frac{1}{2}\|x-z\|_2^2$ as $\mathrm{Proj}_X(z)$. ▪

Proposition 7.3 — **Zero order projection property.** Given any set $X \subseteq \mathbb{R}^n$ (may not necessarily convex) and $z \in \mathbb{R}^n$, for any $x \in X$, we have $\|[z]^+ - z\|_2 \le \|x - z\|_2$

The proof is due to the optimality of $[z]^+$.

Proposition 7.4 — **First order projection property.** Given **convex** set X, the necessary and sufficient condition for the local minimum x^* for problem (7.3) is:

$$\langle x - [z]^+, z - [z]^+ \rangle \le 0, \quad \forall x \in X \tag{7.4}$$

Proof. Define $f(x) := \frac{1}{2}\|x-z\|_2^2$, thus $\nabla f(x) = x - z$, with the necessary condition of local minimum x^* as:

$$\langle \nabla f(x^*), (x - x^*) \rangle \ge 0, \quad \forall x \in X \implies \langle x - [z]^+, z - [z]^+ \rangle \le 0.$$

The sufficency of this condition is due to the convexity of problem (7.3). ■

Proposition 7.5 — **Non-expansive.**

$$\|[\mathbf{z}_1]^+ - [\mathbf{z}_2]^+\| \le \|\mathbf{z}_1 - \mathbf{z}_2\|$$

with X to be a convex set.

Proof. Recall the first order property on z_1, z_2, we obtain

$$\begin{cases} \langle z_1 - [z_1]^+, x - [z_1]^+ \rangle \le 0, \forall x \in X \\ \langle z_2 - [z_2]^+, x - [z_2]^+ \rangle \le 0, \forall x \in X \end{cases} \implies \begin{cases} \langle z_1 - [z_1]^+, [z_2]^+ - [z_1]^+ \rangle \le 0, \forall x \in X \\ \langle z_2 - [z_2]^+, [z_1]^+ - [z_2]^+ \rangle \le 0, \forall x \in X \end{cases}$$

Adding the inequalities above, we derive

$$\langle z_1 - z_2 + [z_2]^+ - [z_1]^+, [z_2]^+ - [z_1]^+ \rangle \le 0 \implies \langle [z_2]^+ - [z_1]^+, [z_2]^+ - [z_1]^+ \rangle \le \langle z_2 - z_1, [z_2]^+ - [z_1]^+ \rangle$$

Applying Cauchy Scharwz inequality, we obtain:

$$\|[z_2]^+ - [z_1]^+\|_2^2 \le \langle z_2 - z_1, [z_2]^+ - [z_1]^+ \rangle \le \|z_2 - z_1\| \|[z_2]^+ - [z_1]^+\|$$

Or equivalently, $\|[z_2]^+ - [z_1]^+\| \le \|z_2 - z_1\|\|$. ■

R The non-expansive property guarantee that $[z]^+$ is unique for any z (over convex set), since otherwise

$$0 \le \|[z]_1^+ - [z]_2^+\| \le \|z - z\| = 0.$$

■ **Example 7.1** 1. For $X = \{x | x \ge 0\} \subseteq \mathbb{R}^2$, we have

$$[z]_i^+ = \max(0, z_i), \qquad i = 1, 2$$

2. For $X = \{\boldsymbol{X} \in \mathbb{S}^n | \boldsymbol{X} \succeq 0\}$, every element admits eigen-decomposition:

$$\boldsymbol{X} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^{\mathrm{T}}.$$

If we want to minimize $\|\boldsymbol{X} - \boldsymbol{Z}\|_F^2$ for given $\boldsymbol{Z}$, the projection

$$[\boldsymbol{Z}]^+ = \boldsymbol{Q}\max(\boldsymbol{0}, \Lambda)\boldsymbol{Q}^{\mathrm{T}}$$

3. For the **ellipsoid set** $X = \{\boldsymbol{x} | \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} \le 1\}$ with $\boldsymbol{Q} \succ 0$, we aim to minimize $\|\boldsymbol{x} - \boldsymbol{z}\|_2^2$ for given $\boldsymbol{z}$:

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} = (\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x})^{\mathrm{T}}\Lambda(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}) := \boldsymbol{y}^{\mathrm{T}}\Lambda\boldsymbol{y}.$$

Thus $\|\boldsymbol{x} - \boldsymbol{z}\|_2 = \|\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{z})\|_2 = \|\boldsymbol{y} - \bar{\boldsymbol{z}}\|_2$. The problem just becomes

$$\begin{aligned} \min \quad & \|\boldsymbol{y} - \bar{\boldsymbol{z}}\| \\ \text{such that} \quad & \boldsymbol{y} \in Y = \{\boldsymbol{y} | \boldsymbol{y}^{\mathrm{T}}\Lambda\boldsymbol{y} \le 1\} \end{aligned}$$

Assume $\bar{\boldsymbol{z}} \notin Y$, then it suffices to solve

$$\begin{aligned} \min \quad & \|\boldsymbol{y} - \bar{\boldsymbol{z}}\|_2^2 \\ \text{such that} \quad & \boldsymbol{y}^{\mathrm{T}}\Lambda\boldsymbol{y} = 1 \end{aligned}$$

Define $L(y, \lambda) = \|\boldsymbol{y} - \bar{\boldsymbol{z}}\|_2^2 + \lambda(\boldsymbol{y}^{\mathrm{T}}\Lambda\boldsymbol{y} - 1)$ $(\lambda \ge 0)$, the necessary condition is

$$\nabla L(y, z) = 0 \implies \begin{cases} 2(\boldsymbol{y} - \bar{\boldsymbol{z}}) + 2\lambda\Lambda\boldsymbol{y} = 0 \\ \boldsymbol{y}^{\mathrm{T}}\Lambda\boldsymbol{y} - 1 = 0 \end{cases}$$

The first equation gives $(\boldsymbol{I} + \lambda\Lambda)\boldsymbol{y} = \boldsymbol{z} \implies \boldsymbol{y} = (\boldsymbol{I} + \lambda\Lambda)^{-1}\bar{\boldsymbol{z}}$, and therefore

$$\bar{\boldsymbol{z}}^{\mathrm{T}}(\boldsymbol{I} + \lambda\Lambda)^{-1}\Lambda(\boldsymbol{I} + \lambda\Lambda)^{-1}\bar{\boldsymbol{z}} = 1$$

Define $\Lambda = \operatorname{diag}(\mu_1, \dots, \mu_n) \succ 0$, we can solve for λ, and then derive $\boldsymbol{y}$:

$$\sum_{i=1}^{n} \frac{\bar{z}_i{}^2\mu_i}{(1 + \lambda\mu_i)^2} = 1$$

■

7.2.3. Feasible diection method

First let's discuss the motivation of feasible direction method:

Proposition 7.6 Consider the problem (7.2), $x^* \in X$ is stationary point if and only if

$$x^* = [x^* - \alpha \nabla f(x^*)]^+. \quad \forall \alpha > 0$$

Proof.

- For the forward direction, the stationarity of x^* is equivalent to

$$\langle \nabla f(x^*), (x - x^*) \rangle \geq 0, \quad \forall x \in X \tag{7.5}$$

Let $z := x^* - \alpha \nabla f(x^*)$, and consider

$$\begin{aligned}
\min_x \|x - z\|^2 &= \|x - x^* + \alpha \nabla f(x^*)\|^2 && (7.6a)\\
&= \|x - x^*\|^2 + 2\alpha \langle \nabla f(x^*), (x - x^*) \rangle + \alpha^2 \|\nabla f(x^*)\|^2 && (7.6b)\\
&\geq \alpha^2 \|\nabla f(x^*)\|^2, && (7.6c)
\end{aligned}$$

which implies the minimum point for (7.6) is x^*, i.e., $x^* = [z]^+$.

- For the reverse direction, assume x^* is not the stationary point, i.e., $\exists x^0 \in X$ such that $\langle \nabla f(x^*), (x^0 - x^*) \rangle < 0$. We set $d^0 = x^0 - x^*$. For fixed $\alpha > 0$, we construct $x^1 \in X$ such that

$$d^1 := x^1 - x^* := \frac{-\alpha \langle \nabla f(x^*), d^0 \rangle}{\|d^0\|_2^2} d^0$$

Substituting x^1 into the problem(7.6), we have

$$\begin{aligned}
\|x^1 - z\|^2 &= \|d^1\|^2 + 2\alpha \langle \nabla f(x^*), d^1 \rangle + \alpha^2 \|\nabla f(x^*)\|^2\\
&= \left[\frac{\alpha^2 \langle \nabla f(x^*), d^0 \rangle^2}{\|d^0\|^4} \right] \|d^0\|^2 + 2\alpha \cdot \frac{-\alpha \langle \nabla f(x^*), d^0 \rangle}{\|d^0\|_2^2} \langle \nabla f(x^*), d^0 \rangle + \alpha^2 \|\nabla f(x^*)\|^2\\
&= -\frac{\alpha^2 \langle \nabla f(x^*), d^0 \rangle^2}{\|d^0\|^2} + \alpha^2 \|\nabla f(x^*)\|^2\\
&< \alpha^2 \|\nabla f(x^*)\|^2 = \|x^* - z\|^2,
\end{aligned}$$

i.e., x^* cannot be the minimizer of problem(7.6), which contradicts to the fact that $x^* = [z]^+$.

■

The idea for feasible direction method is that we generate a sequence of $\{x^k\} \subseteq X$ such that $f(x^{r+1}) \leq f(x^r)$, or equivalently, find $\bar{x^r} \in X$ such that the $\bar{x^r} - x^r$ is the descent direction:

$$\langle \nabla f(x^r), (\bar{x^r} - x^r) \rangle \leq 0 \tag{7.7a}$$

and therefore

$$x^{r+1} = x^r + \alpha_r(\bar{x^r} - x^r), \quad \alpha_r \in (0,1] \tag{7.7b}$$

The feasibility of x^{r+1} is guaranteed since x^{r+1} is a convex combination of feasible points x^r and $\bar{x^r}$. Here the problem remains to find $\bar{x}^r$

Projection Gradient Method. One way of finding $\bar{x}^r$ is to compute $x^r - s_r \nabla f(x^r)$ and project it back into X, as $\bar{x^r}$:

$$\bar{x^r} = [x^r - s_r \nabla f(x^r)]^+, \quad s_r > 0 \ \cdot$$

Conditional Gradient. Another way is to linearize the objective function and solve for $\bar{x}^r$:

$$\bar{x}^r \approx \arg\min_{x \in X} f(x) \implies \bar{x}^r = \arg\min_{x \in X} f(x^r) + \langle \nabla f(x^r), (x - x^r) \rangle.$$

Chapter 8

Week8

8.1. Monday

Comments for P-C method in A5. The Newton's method gives the iterative formula

$$x^{k+1} = x^k - [F'(x^k)]^{-1}F(x^k)$$

Moreover, the two-step Newton's method gives:

$$x^{k+2} = x^k - [F'(x^k)]^{-1}F(x^k) - [F'(x^{k+1})]^{-1}F(x^{k+1}), \tag{8.1}$$

with convergence order $O(\|x^k - x^*\|^4)$.

However, computing Hessian matrix inverse twice in one iteration is expansive. The predictor-corrector method just change $F'(x^{k+1})$ as $F'(x^k)$ in (8.2):

$$\begin{aligned} x^{k+2} &= x^k - [F'(x^k)]^{-1}F(x^k) - [F'(x^k)]^{-1}F(x^{k+1}) \\ &= x^k - [F'(x^k)]^{-1}[F(x^k) + F(x^{k+1})], \end{aligned}$$

or equivalently,

$$x^{k+1} = x^k - [F'(x^k)]^{-1}[F(x^k) + F(x^{k+1/2})],$$

which is also called Composite Newton, with convergence rate $O(\|x^k - x^*\|^3)$.

Comments for Incremental Gradient Method. The reason why the disminishing step size performs poorer than the constant step size remains to be found.

8.1.1. Constraint optimization

Theorem 8.1 — Lagrange necesary optimality condition. Given the problem

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{such that} \quad & h_i(\boldsymbol{x}) = 0, \quad i = 1, \dots, m \end{aligned} \tag{8.2}$$

with $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $h_i : \mathbb{R}^n \mapsto \mathbb{R}$ are continuously differentiable functions.Let $\boldsymbol{x}^*$ be a regular point, i.e., $\nabla h_i(\boldsymbol{x}^*)$ are linearly independent, λ^* is the corresponding multiplier, then

- The first order necessary optimality condition is

$$\nabla_x L(\boldsymbol{x}^*, \lambda^*) = 0, \Longleftrightarrow \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(\boldsymbol{x}^*) = 0,$$

  where λ^* is uniquely determined.

- The second order necessary optimality condition is (pre-assume $f, g \in \mathcal{C}^2$):

$$\boldsymbol{y}^{\mathrm{T}} \left[\nabla^2_{xx} L(\boldsymbol{x}^*, \lambda^*)\right] \boldsymbol{y} \ge 0, \quad \forall \boldsymbol{y} \text{ s.t. } \langle \nabla h(\boldsymbol{x}^*), \boldsymbol{y} \rangle = 0,$$

  or equivalently,

$$\boldsymbol{y}^{\mathrm{T}} \left[\nabla^2 f(\boldsymbol{x}^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 h_i(\boldsymbol{x}^*)\right] \boldsymbol{y} \ge 0, \quad \forall \boldsymbol{y} \text{ s.t. } \langle \nabla h(\boldsymbol{x}^*), \boldsymbol{y} \rangle = 0,$$

  i.e., the Hessian matrix $L_{xx}(\boldsymbol{x}^*, \lambda^*)$ is PSD over the null space of Jacobian matrix of $h(\boldsymbol{x}^*)$.

Theorem 8.2 — Lagrange sufficient optimality condition. Given the problem (8.2), the second order sufficent optimality conditon is

$$\begin{cases} \nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*,\lambda^*) = 0, & \nabla_{\lambda} L(\boldsymbol{x}^*,\lambda^*) = 0 \\ \boldsymbol{y}^{\mathrm{T}} \nabla^2_{\boldsymbol{x}\boldsymbol{x}} L(\boldsymbol{x}^*,\lambda^*)\boldsymbol{y} > 0, & \forall \boldsymbol{y} \neq \boldsymbol{0} \text{ with } \langle \nabla h(\boldsymbol{x}^*), \boldsymbol{y} \rangle = \boldsymbol{0} \end{cases}$$

8.1.2. Inequality Constraint Problem

Given the problem

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{such that} \quad & h(\boldsymbol{x}) = 0 \\ & g(\boldsymbol{x}) \leq 0, \end{aligned} \tag{8.3}$$

with $f : \mathbb{R}^n \mapsto \mathbb{R}, h : \mathbb{R}^n \mapsto \mathbb{R}^m, m < n; g : \mathbb{R}^n \mapsto \mathbb{R}^p$, we set a general set

$$X = \{x \mid h(\boldsymbol{x}) = 0, g(\boldsymbol{x}) \leq 0\}$$

We set the Lagrange function

$$L(\boldsymbol{x},\lambda,\mu) = f(\boldsymbol{x}) + \lambda^{\mathrm{T}} h(\boldsymbol{x}) + \mu^{\mathrm{T}} g(\boldsymbol{x})$$

Define the dual function

$$Q(\lambda,\mu) = \inf_X L(\boldsymbol{x},\lambda,\mu) \leq f(\boldsymbol{x})$$

8.2. Monday Tutorial: Review for CIE6010

Matrix Calculus.

- $$\frac{\partial \boldsymbol{a}^{\mathrm{T}} \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{a}$$

  $$\frac{\partial \boldsymbol{a}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{a}$$

- $$\frac{\partial \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{\partial \boldsymbol{x}} = 2\boldsymbol{x}$$

- $$\frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{A}^{\mathrm{T}}$$

- $$\frac{\partial f^{\mathrm{T}}g}{\partial \boldsymbol{x}} = \frac{\partial f}{\boldsymbol{x}}g + \frac{\partial g}{\boldsymbol{x}}f$$

- $$\frac{\partial x^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}$$

- $$\frac{\partial \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2}{\partial \boldsymbol{x}} = 2\frac{\partial \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}) = -2\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})$$

-

Proposition 8.1 — Chain Rule. Let $h(x) = g(f(x))$ with $g : \mathbb{R}^m \mapsto \mathbb{R}^n$ and $f : \mathbb{R}^k \mapsto \mathbb{R}^m$, then we have

$$\nabla h(x) = \nabla f(x) \nabla g(f(x)) \tag{8.4}$$

$$\nabla_{\mathbf{X}} \|\boldsymbol{A}\boldsymbol{X} - \boldsymbol{B}\|_F^2 =$$

Proof. Firstly we compute $\nabla_{\mathbf{X}} \|\boldsymbol{X}\|_F^2$:

$$\|\boldsymbol{X}\|_F^2 = \sum_{i,j} x_{ij}^2 \implies \nabla_{\mathbf{X}} \|\boldsymbol{X}\|_F^2 = \left[\frac{\partial \sum_{i,j} x_{ij}^2}{\partial \mathbf{X}}\right]$$

Note that

$$\left[\frac{\partial \sum_{i,j} x_{ij}^2}{\partial \mathbf{X}}\right]_{ij} = \left[\frac{\partial \sum_{i,j} x_{ij}^2}{\partial x_{ij}}\right] = 2x_{ij},$$

which follows that $\nabla_{\boldsymbol{X}}\|\boldsymbol{X}\|_F^2 = 2\boldsymbol{X}$, and therefore

$$\nabla_{\boldsymbol{X}}\|\underbrace{\boldsymbol{A}\boldsymbol{X}-\boldsymbol{B}}_{\boldsymbol{Y}}\|_F^2 = \nabla_{\boldsymbol{X}}(\boldsymbol{A}\boldsymbol{X}-\boldsymbol{B})\nabla_{\boldsymbol{Y}}\|\boldsymbol{Y}\|_F^2 = 2\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}$$

■

Necessary and Sufficient Optimality condition.

Proposition 8.2 — Unconstraint optimality condition. Suppose $f: X \mapsto \mathbb{R}$ is a second order differentiable function defined on domain X

- Necessary condition for local minimum x^*:

$$\begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) \succeq 0 \end{cases}$$

- Sufficient condition for local minimum x^*:

$$\begin{cases} \nabla f(x^*) = 0 \\ \nabla^2 f(x^*) \succ 0 \end{cases}$$

Corollary 8.1 For x^* satisfying the sufficient condition, there exists $\gamma, \varepsilon > 0$ such that

$$f(x) \geq f(x^*) + \frac{\gamma}{2}\|x - x^*\|^2, \qquad \forall x \text{ such that } \|x - x^*\| < \varepsilon$$

Proof. For any direction d, we have

$$\begin{aligned} f(x^* + d) - f(x^*) &= \langle \nabla f(x^*), d \rangle + \frac{1}{2} d^{\mathrm{T}} \nabla^2 f(x^*) d + o(\|d\|_2^2) \\ &\geq \frac{1}{2}\lambda_{\min}\|d\|^2 + o(\|d\|_2^2) \\ &= \left[\frac{1}{2}\lambda_{\min} + o(1)\right]\|d\|_2^2 \end{aligned}$$

■

Proposition 8.3 — **Constraint but convex optimality condition.** Consider the convex optimization problem

$$\begin{aligned} \min \quad & f_0(x) \\ & x \in X \text{ for convex set } X, \end{aligned} \tag{8.5}$$

x is optimal if and only if

$$\langle \nabla f_0(x), (y - x) \rangle \geq 0, \quad \forall y \in X \tag{8.6}$$

Proof. For the reverse direction, consider the Taylor expansion and the inequality below:

$$f_0(y) \geq f_0(x) + \langle \nabla f_0(x), (y - x) \rangle, \quad \forall y \in X$$

For the forward direction, assume x is optimal but (8.6) does not hold, i.e., for some $y \in X$, we have

$$\langle \nabla f_0(x), (y - x) \rangle < 0.$$

Consider the point $z(t) = ty + (1 - t)x$ with the derivative

$$\frac{\mathrm{d}}{\mathrm{d}t} f_0(z(t))|_{t=0} = \langle \nabla f_0(x), (y - x) \rangle < 0,$$

i.e., for some $t > 0$, we have $f_0(z(t)) < f_0(x)$, which is a contradiction. ■

Proposition 8.4 — **KKT conditions.** Consider the standard optimization problem

$$\begin{aligned} \min \quad & f_0(x) \\ \text{such that} \quad & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{aligned} \tag{8.7}$$

with the dual problem

$$\begin{aligned} \max \quad & g(\lambda, \gamma) \\ & \lambda_i \geq 0, \quad i = 1, \ldots, m \end{aligned} \tag{8.8}$$

1. The necessary condition for primal and dual optimal points x^*, (λ, γ^*) is

- Primal Feasibility: $f_i(x) \leq 0, \quad i = 1, \ldots, m; \, h_i(x) = 0, \quad i = 1, \ldots, p$
- Dual Feasibility: $\lambda \geq 0$
- Complementary Slacknes: $\lambda_i f_i(x^*) = 0, \quad i = 1, \ldots, m$
- Stationarity of Lagrange: $\nabla_x L(x^*, \lambda^*, \gamma^*) = 0$

2. When the primal problem is convex, the KKT conditions above are both necessary and sufficient optimality conditions.

Convex functions and convex sets. How to verify the convexity?

1. By definition:

Proposition 8.5 The functions f_1, f_2 are convex implies that $f := \max\{f_1, f_2\}$ is convex.

Proof. For any $\lambda \in (0,1)$ and $x, y \in \operatorname{dom} f$, we have

$$
\begin{aligned}
f(\lambda x + (1-\lambda)y) &= \max\{f_1(\lambda x + (1-\lambda)y), f_2(\lambda x + (1-\lambda)y)\} \\
&\leq \max\{\lambda f_1(x) + (1-\lambda) f_1(y), \lambda f_2(x) + (1-\lambda) f_2(y)\} \\
&\leq \lambda \max\{f_1(x), f_2(x)\} + (1-\lambda) \max\{f_1(y), f_2(y)\} \\
&= \lambda f(x) + (1-\lambda) f(y)
\end{aligned}
$$

■

2. By properties:

Proposition 8.6 — **First Order Conditions.** For differentiable function f, f is convex if and only if for all $x, y \in \operatorname{dom} f$,

$$f(y) \geq f(x) + \langle \nabla f(x), (y - x) \rangle$$

Proposition 8.7 — **Second Order Conditions.** For twice differentiable function f, f is convex if and only if for all $x \in \operatorname{dom} f$,

$$\nabla^2 f(x) \succeq 0$$

Algorithms. For algorithms, make clear of those things below:

- How it works?
- What kind of problems does it intend for?
- What is the convergence rate?

Primal and Dual.

■ **Example 8.1** Derive the dual for the optimization problem

$$\min \quad \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_\infty. \tag{8.9}$$

1. First make some transformations. We set $t := \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_\infty$, and thus suffices to solve

$$\begin{aligned} \min \quad & t \\ & \boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\le t \\ & -\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}\le t \end{aligned} \tag{8.10}$$

2. Thus the Lagrange function is defined as:

$$L(\boldsymbol{x},t,u,v) = t + \begin{bmatrix}\boldsymbol{u}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}}\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{A} & -\boldsymbol{1} \\ -\boldsymbol{A} & -\boldsymbol{1}\end{bmatrix}\begin{bmatrix}\boldsymbol{x} \\ t\end{bmatrix} - \begin{bmatrix}\boldsymbol{b} \\ -\boldsymbol{b}\end{bmatrix}\right)$$

and $\inf_{\boldsymbol{x},t} L(\boldsymbol{x},t,u,v) = -\begin{bmatrix}\boldsymbol{u}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}}\end{bmatrix}\begin{pmatrix}\boldsymbol{b} \\ -\boldsymbol{b}\end{pmatrix}$, which implies the dual problem

$$\begin{aligned} \max \quad & -\begin{bmatrix}\boldsymbol{u}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}}\end{bmatrix}\begin{pmatrix}\boldsymbol{b} \\ -\boldsymbol{b}\end{pmatrix} \\ & \boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}=0 \\ & -\boldsymbol{u}^{\mathrm{T}}\boldsymbol{1}-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{1}=-1 \end{aligned}$$

■

Theorem 8.3 — **Weak Duality.** For any optimization problem with dual optimal value q^* and primal optimal value f^*, we have

$$q^* \leq f^*$$

Theorem 8.4 — **Strong Duality.** For convex optimization problem, we have

1. Both primal and dual has optimal feasible solution
2. One is infeasible implies the other is unbounded
3. If both has optimal feasible solution, then $q^* = f^*$.

Convergence Analysis. For (not necessarily quadratic) optimization problem $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$, consider the steepest descent

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha^k \nabla f(\boldsymbol{x}^k) = (\boldsymbol{I} - \alpha^k \boldsymbol{Q})\boldsymbol{x}^k,$$

which implies

$$\begin{aligned}
\|\boldsymbol{x}^{k+1}\|^2 &= (\boldsymbol{x}^k)^{\mathrm{T}}(\boldsymbol{I} - \alpha^k \boldsymbol{Q})^2 \boldsymbol{x}^r \leq \lambda_1[(\boldsymbol{I} - \alpha^k \boldsymbol{Q})^2]\|\boldsymbol{x}^r\|^2 \\
&= \|\boldsymbol{x}^k\|^2 \cdot \max\{(1-\alpha^k m)^2, (1-\alpha^k M)^2\},
\end{aligned}$$

which implies

$$\frac{\|\boldsymbol{x}^{k+1}\|}{\|\boldsymbol{x}^k\|} \leq \max\{|1-\alpha^k m|, |1-\alpha^k M|\}$$

Choosing $\alpha^k = \frac{2}{M+m}$ s.t. $\max\{|1-\alpha^k m|, |1-\alpha^k M|\}$ is maximized, we have

$$\frac{\|\boldsymbol{x}^{k+1}\|}{\|\boldsymbol{x}^k\|} \leq \frac{M-m}{M+m}.$$

Chapter 9

Week9

9.1. Monday

Annonuncement. The exam is on Wednesday, from 10:30 to 12:00 in F302, Shaw College. The exam will be quite different frome previous years, since some of you search all the tests the lecturer given online. KKT are the most important one in the test.

9.1.1. Reviewing for KKT

$$\begin{aligned}\min \quad & f(x) \\ & h_i(x) = 0, \quad i = 1,2,\dots,m < n \\ & g_i(x) \le 0, \quad i = 1,2,\dots,r\end{aligned} \tag{9.1}$$

The Lagrange function is given by:

$$L(x,\lambda,\mu) = f(x) + \lambda^{\mathrm{T}} h(x) + \mu^{\mathrm{T}} g(x)$$

The active set is the set of indices such that the inequalities are satisfied:

$$A(x) = \{i \mid g_i(x) = 0, 1 \le i \le r\}$$

The pre-assumption is the constraint qualification (QC), or called **regularity**. At the local minimum point x^*, we have

$$\{\nabla h_i(x^*), \forall i, \nabla g_i(x^*), \forall i \in A(x^*)\} \text{ are linearly independent}$$

Under differentiability and a QC, we have KKT (necessary) conditions:

Proposition 9.1 — First Order KKT Conditions. For local minimum x^* of (9.1) that is differentiable, we have

1. $h(x) = 0, g(x) \le 0$
2. $\nabla_x L(x, \lambda, \mu) = 0, \mu \ge 0$
3. $\mu \circ g(x) = 0$

Proposition 9.2 — Second Order KKT Conditions. For local minimum x^* of (9.1) that is twice differentiable, we have

1. $\boldsymbol{y}^{\mathrm{T}} \nabla^2_{xx} L(x, \lambda, \mu) \boldsymbol{y} \ge 0$, for $\forall \boldsymbol{y} \in V(x^*)$ with

$$V(x^*) = \{\boldsymbol{y} \mid \nabla^{\mathrm{T}} h_i(x) \boldsymbol{y} = 0, \forall i; \nabla^{\mathrm{T}} g_i(x) \boldsymbol{y} = 0, \forall i \in A(x^*)\}$$

During the exam, the counter-example for KKT conditon is given (the pre-assumption of QC is not met).

■ **Example 9.1**

$$\begin{aligned} \min \quad & x_1 \\ & x_2 - x_1^2 \le 0 \\ & -x_1 \le 0 \end{aligned}$$

The Lagrange funcion is

$$L(x, \mu) = x_1 + \mu_1(x_2 - x_1^2) + \mu_2(-x_1)$$

$$\nabla_x L(x, \mu) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \mu_2 \begin{pmatrix} -2x_1 \\ 1 \end{pmatrix} + \mu_2 \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \mu_1, \mu_2 \ge 0$$

which follows that

$$\mu_1 = 0, \mu_2 = 1, \mu_2(-x_1) = 0 \implies x_1 = 0.$$

■

■ **Example 9.2**

$$\begin{aligned} \min \quad & x_2 \\ & x_2 - x_1^2 \leq 0 \\ & -x_2 \leq 0 \end{aligned}$$

The Lagrange function is

$$L(x,\mu) = x_2 + \mu_1(x_2 - x_1^2) + \mu_2(-x_2)$$

$$\nabla_x L = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \mu_1 \begin{pmatrix} -2x_1 \\ 1 \end{pmatrix} + \mu_2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

In this case, the regularity condition is not satisfied. Let's try to solve for this question first:

$$1 + \mu_1 - \mu_2 = 0 \implies \mu_2 \geq 1, x_2 = 0$$

$$\nabla^2_{xx} L = \mu_1 \begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix} \begin{cases} = 0, & x_2 < x_1^2 \\ \text{negative semi-definite}, & x_2 = x_1^2 \end{cases}$$

However, the condition for NSD case, $y^{\mathrm{T}} \nabla^2_{xx} L y \geq 0$ for $y \in V(x^*)$ must be true. ■

Constraint Qualifications.

1. Regularity (LICQ)
2. Slater Condition: If equality constraint $h(x)$ is affine, g_i's are convex, there exists x such that $h(x) = 0, g(x) < 0$.
3. Linear Constraints: All h and g are linear
4. MFCQ: Proposition 4.3.8. in the book

R If f is convex, then all KKT conditions are sufficient.

9.2. Monday Tutorial: Reviewing for Mid-term

How to answer sufficient or necessary conditions during the exam? . First derive some inituitive and weak conditons; and then strengthen it by some properties.

Something highlight during the exam. How to compute the dual for specific problems? How to derive the KKT condition? How to compute the gradient? How to dervie iterative formula for some specific problems? What's the convergence rate of iterative algorithms based on which pre-assumptions? e.g., Newton's method has quadratic convergence once the lagrange function is Lipschitz continuous and Hessian matrix is non-singular. What's the proof for the convergence rate?

Chapter 10

Week10

10.1. Monday

Announcement. No assignment in this week, so you may take a break. However, in next week new assignments and projects will be updated, which requires you to apply penalty algorithms.

Theorem 10.1 — Farka's Lemma. Let $\boldsymbol{a}_1,\dots,\boldsymbol{a}_r \in \mathbb{R}^n$, and

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{a}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_r^{\mathrm{T}} \end{pmatrix},$$

then for any $\boldsymbol{c} \in \mathbb{R}^n$,

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} \le 0, \quad \forall \boldsymbol{y} \text{ such that } \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \le 0, \tag{10.1a}$$

if and only if

$$\boldsymbol{c} = \boldsymbol{A}\boldsymbol{u}, \forall \boldsymbol{u} \ge 0, \boldsymbol{u} \in \mathbb{R}^r \tag{10.1b}$$

(R) The interpretation is that the vector $\boldsymbol{c}$ has more than 90 degrees angle with all vectors $\boldsymbol{a}_i$ in the polar cone, if and only if $\boldsymbol{c}$ is in the polar cone.

Proof. To show the converse, we have

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y},$$

with $\boldsymbol{u} \geq 0, \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \leq 0$, and thereofre $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} \leq 0$. ■

Proof. ■

Convex Program.

$$\begin{aligned} \min \quad & f(x) \\ \text{such that} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & g(\boldsymbol{x}) \leq 0 \end{aligned}$$

with f, g to be convex. The Lagrangian function is given by:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{\mu}^{\mathrm{T}} g(\boldsymbol{x}),$$

with $\boldsymbol{\mu} \geq 0$. This function is convex in $\boldsymbol{x}$. Therefore the dual function is given by:

$$Q(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{x}, \boldsymbol{\mu} \geq 0} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

Proposition 10.1 — Weak Deality.

$$Q(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq f(\boldsymbol{x})$$

for dual feasible $\boldsymbol{\lambda}, \boldsymbol{\mu}$ and primal feasible $\boldsymbol{x}$.

We are curious on the tightest lower bound on LHS, thus maximizing the dual function to obtain the dual program:

$$\begin{aligned} \max \quad & Q(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{such that} \quad & \boldsymbol{\mu} \geq 0 \end{aligned}$$

Proposition 10.2 — Strong Duality. For convex programming, we have

$$d^* = p^*,$$

with d^*, p^* to be the optimal value from dual and primal problems, respectively.

QC: one of the followings is satisfied

- $g_i(\boldsymbol{x})$ are linear

- $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, g(\boldsymbol{x}) \le 0$

- Regularity

Under QC, the primal and dual could attain optimality togethoer iff

- $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, g(\boldsymbol{x}) \le 0$

- $\boldsymbol{\mu} \ge 0$

- $\boldsymbol{\mu} \circ g(\boldsymbol{x}) = \mathbf{0}$

We have derive the dual formula for linear programming, but how about the quadratic programming?

■ **Example 10.1**

$$\begin{aligned} p^* = \min \quad & \tfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}, \quad \boldsymbol{Q} \succ 0 \\ \text{such that} \quad & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \end{aligned} \tag{10.2}$$

The Lagrangian function $L(\boldsymbol{x},\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$, and therefore

$$Q(\boldsymbol{\mu}) = \min_{\boldsymbol{x},\boldsymbol{\mu}\ge 0} L(\boldsymbol{x},\boldsymbol{\mu}) \tag{10.3}$$

The optimality condition implies that

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{\mu}) = \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu} = 0 \implies \boldsymbol{x} = -\boldsymbol{Q}^{-1}(\boldsymbol{c} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu})$$

Thus substituting optimal $\boldsymbol{x}$ into (10.3), we derive

$$Q(\boldsymbol{\mu}) = -\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{t}^{\mathrm{T}}\boldsymbol{\mu} + \text{constant}$$

Thus we derive the dual program:

$$\begin{aligned} d^* = \min \quad & \tfrac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{\mu} + \boldsymbol{t}^{\mathrm{T}}\boldsymbol{\mu}, \\ \text{such that} \quad & \boldsymbol{\mu} \ge 0 \end{aligned} \tag{10.4}$$

where $\boldsymbol{P} := \boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^{\mathrm{T}}$.

10.1.1. Penalty Algorithms

Logarithm Penalty. Consider the inequality constraint problem

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ & g_i(\boldsymbol{x}) \le 0 \end{aligned}$$

The Barrier problem is given by:

$$\min f(x) - \mu \sum_i \log(-g_i(\boldsymbol{x})), \quad \mu > 0$$

As $\mu \to 0$, $x(\mu)$ converges to the optimal solution. We pick big μ at first and obtain a good initial guess, and then we continue to decrease μ.

Quadratic Penalty. For the constraint problem

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ & h(\boldsymbol{x}) = 0 \\ & \boldsymbol{x} \in X \end{aligned}$$

The quadratic penalty algorithm aims to solve

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) + \lambda^{\mathrm{T}} h(\boldsymbol{x}) + \tfrac{c}{2}\|h(\boldsymbol{x})\|_2^2 \\ & \boldsymbol{x} \in X, \end{aligned}$$

where λ is **bounded**. Conversely, as $c \to \infty$, $\boldsymbol{x}(c)$ converges to the optimal solution. We pick small c at first and obtain a good initial guess, and then we continus to increase c.

Example 10.2

$$\begin{aligned} \min \quad & \tfrac{1}{2}(x_1^2 + x_2^2) \\ & x_1 = 1 \end{aligned}$$

The quadratic penalty function is

$$L_c(x) = \frac{1}{2}(x_1^2 + x_2^2) + \lambda(x_1 - 1) + \frac{c}{2}(x_1 - 1)^2$$

and therefore

$$\nabla_{\boldsymbol{x}} L_c(\boldsymbol{x}) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 0 \end{pmatrix} + c \begin{pmatrix} x_1 - 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

which follows that

$$x_1(\lambda, c) = \tfrac{c-\lambda}{c+1}, \quad x_2(\lambda, c) = 0$$

We can apply two algorithm to converge to optimal solution:

1. Quadratic Pendlty Method: As $c \to \infty$ with λ bounded, we derive $x_1(\lambda, c) \to 1$.
2. Lagrangian Multiplier Method: We set $\nabla L(\boldsymbol{x}, \lambda) = 0$ to obtain an appropriate $\lambda^* = -1$. As $\lambda \to \lambda^*$, we obtain $x_1(\lambda, c) \to 1$ for $c > 1$ (the key for this kind of algorithm is to choose big c).

Such an algorithm can also be applied for the non-convex problem, e.g.,

$$\begin{aligned} \min \quad & \tfrac{1}{2}(-x_1^2 + x_2^2) \\ & x_1 = 1 \end{aligned}$$

10.2. Wednesday

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ & h(\boldsymbol{x}) = 0 \\ & \boldsymbol{x} \in X \subseteq \mathbb{R}^n \end{aligned}$$

The augmented Lagrangian function is given by:

$$L_{\boldsymbol{c}}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathrm{T}} h(\boldsymbol{x}) + \frac{\boldsymbol{c}}{2}\|h(\boldsymbol{x})\|^2$$

Quadratic Penalty (Courant, 1943).

$$\begin{aligned} \boldsymbol{x}^r &= \arg\min_{\boldsymbol{x} \in X} L_{\boldsymbol{c}^r}(\boldsymbol{x}, \boldsymbol{\lambda}^r, \boldsymbol{c}^r) \\ & \text{increase } \boldsymbol{c}^{r+1} > \boldsymbol{c}^r \to \infty, \qquad \|\boldsymbol{\lambda}^r\| < +\infty \end{aligned}$$

Augmented Lagrangian Multiplier Method.

$$\begin{aligned} \boldsymbol{x}^r &= \arg\min_{\boldsymbol{x} \in X} L_{\boldsymbol{c}}(\boldsymbol{x}, \boldsymbol{\lambda}^r) \\ \boldsymbol{\lambda}^{r+1} &= \boldsymbol{\lambda}^r + \boldsymbol{c}h(\boldsymbol{x}) \end{aligned}$$

where $\boldsymbol{c}$ is sufficiently large in general, but not goes to infinite.

■ **Example 10.3** Given problems

$$\begin{aligned} \min \quad & \tfrac{1}{2}(-x_1^2 + x_2^2) \\ & x_1 = 1 \end{aligned}$$

with optimal solution $\boldsymbol{x}^* = (1,0); \lambda^* = 1$. The Lagrangian function is given by:

$$L_c(\boldsymbol{x}, \lambda) = \frac{1}{2}(-x_1^2 + x_2^2) + \lambda(x_1 - 1) + \frac{c}{2}(x_1 - 1)^2,$$

with

$$\nabla_{\boldsymbol{x}} L_c = \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix} + [\lambda + c(x_1 - 1)] \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Question: Why $\nabla_{\boldsymbol{x}} L_c = \mathbf{0}$ to be the necessary condition?

$$\boldsymbol{x} = \begin{pmatrix} \frac{c-\lambda}{c-1} \\ 0 \end{pmatrix}$$

Also,

$$\nabla^2_{\boldsymbol{x}\boldsymbol{x}} L_c = \begin{pmatrix} c-1 & 0 \\ 0 & c+1 \end{pmatrix}$$

Comments on the change of c and λ.

Optimality condtion for orign problem.

$$\nabla_{\boldsymbol{x}} L_{\boldsymbol{c}}(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) + \nabla h(\boldsymbol{x})\lambda + c\nabla h(\boldsymbol{x}) h(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) + \nabla h(\boldsymbol{x})(\boldsymbol{\lambda} + ch(\boldsymbol{x}))$$

If $c \to \infty$, then $h(\boldsymbol{x}) \to \mathbf{0}$