

# Optimization Algorithm Notes

Depu Meng

April 17, 2019



# Contents

<b>1</b>	<b>Introduction to Optimization Algorithms</b>	<b>1</b>
1.1	Goal of the Course . . . . .	1
1.2	Basic Concepts . . . . .	1
1.2.1	Problem Definition . . . . .	1
1.3	Optimal Conditions . . . . .	2
1.3.1	Unconstrained Optimization . . . . .	2
1.3.2	Constrained Optimization . . . . .	2
1.4	Descent function . . . . .	5
1.5	Convergence of Algorithm . . . . .	5
1.5.1	Search Methods . . . . .	5
<b>2</b>	<b>Unconstrained Optimization</b>	<b>7</b>
2.1	Gradient Based Methods . . . . .	7
2.1.1	Determine Search Direction . . . . .	7
2.1.2	Determine Step Factor – Line Search . . . . .	8
2.1.3	Global Convergence . . . . .	8
2.1.4	Steepest Descent Method . . . . .	9
2.1.5	Newton Method . . . . .	9
2.1.6	Quasi-Newton Method . . . . .	10
2.1.7	Conjugate Gradient Method . . . . .	12
2.2	Trust Region Method . . . . .	15
2.2.1	Trust Region Subproblem . . . . .	15
2.2.2	How to select $e_k$ . . . . .	16
<b>3</b>	<b>Constrained Optimization</b>	<b>17</b>
3.1	Quadratic Programming . . . . .	17
3.1.1	Solution of Quadratic Programming . . . . .	17
3.1.2	Equality Constrained Quadratic Programming . . . . .	17
3.1.3	General Quadratic Programming . . . . .	17
3.2	Equality Constrained Problem . . . . .	18
3.2.1	Lagrange-Newton method . . . . .	18
3.2.2	Sequential Quadratic Programming method . . . . .	19
3.3	General Nonlinear Constrained Problem . . . . .	19

iv CONTENTS

3.3.1	Sequential Quadratic Programming method . . . . .	19
3.3.2	Penalty method . . . . .	20
3.3.3	Extended Lagrange function method . . . . .	22
3.3.4	Barrier method . . . . .	23
<b>4</b>	<b>Convex Optimization</b>	<b>25</b>
4.1	Convex set . . . . .	25
4.1.1	Affine set . . . . .	25
4.1.2	Convex set . . . . .	25
4.1.3	Cone . . . . .	25
4.1.4	Proper cones and generalized inequalities . . . . .	26
4.2	Convex function . . . . .	26
4.2.1	First order condition . . . . .	26
4.3	Convex optimization . . . . .	26
4.3.1	Quadratically constrained quadratic program . . . . .	26
4.3.2	Second-order cone program . . . . .	26
<b>5</b>	<b>Sparse Optimization</b>	<b>29</b>
5.1	Compressed Sensing . . . . .	29
5.1.1	Problem formulation . . . . .	29
5.1.2	Pursuit Algorithms . . . . .	29

# 1 Introduction to Optimization Algorithms

## 1.1 Goal of the Course

- Understand foundations of optimization
- Learn to analyze widely used optimization algorithms
- Be familiar with implementation of optimization algorithms

## 1.2 Basic Concepts

### 1.2.1 Problem Definition

Find the value of the decision variable s.t. objective function is maximized/minimized under certain conditions.

$$\begin{aligned} \min f(x) \\ \text{s.t. } x \in \mathcal{S} \subset \mathbb{R}^n \end{aligned} \quad (1.1)$$

Here, we call  $\mathcal{S}$  *feasible region*.

We often denote constrained optimization Problem as

$$\begin{aligned} \min f(x) \\ \text{s.t. } \quad g_i(x) \geq 0, i = 1, \dots, n \\ \quad \quad b_i(x) = 0, i \in 1, \dots, m \end{aligned} \quad (1.2)$$

**Definition 1.2.1.** *Global Optimality.* For global optimal value  $x^* \in \mathcal{S}$ ,

$$f(x^*) \leq f(x), \forall x \in \mathcal{S} \quad (1.3)$$

**Definition 1.2.2.** *Local Optimality.* For local optimal value  $x^* \in \mathcal{S}$ ,  $\exists U(x^*)$ , such that

$$f(x^*) \leq f(x), \forall x \in \mathcal{S} \cap U(x^*) \quad (1.4)$$

**Definition 1.2.3.** *Feasible direction.* Let  $x \in \mathcal{S}$ ,  $d \in \mathbb{R}^n$  is a non-zero vector. if  $\exists \delta > 0$ , such that

$$x + \lambda d \in \mathcal{S}, \forall \lambda \in (0, \delta) \quad (1.5)$$

Then  $d$  is a **feasible direction** at  $x$ . We denote  $F(x, \mathcal{S})$  as the set of feasible directions at  $x$ .

**Definition 1.2.4.** *Descent direction.*  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $x \in \mathbb{R}^n$ ,  $d$  is a non-zero vector. If  $\exists \delta > 0$ , such that

$$f(x + \lambda d) < f(x), \forall \lambda \in (0, \delta) \quad (1.6)$$

Then  $d$  is a **descent direction** at  $x$ . We denote  $D(x, f) = \{d \mid \nabla f(x)^T d < 0\}$  as the set of descent direction at  $x$ .

## 1.3 Optimal Conditions

### 1.3.1 Unconstrained Optimization

First-order necessary condition:  $f(x)$  is differentiable at  $x$ ,

$$\nabla f(x) = 0 \quad (1.7)$$

Second-order necessary condition:  $f(x)$  is second-order differentiable at  $x$ ,

$$\nabla f(x) = 0 \quad (1.8)$$

$$\nabla^2 f(x) \geq 0 \quad (1.9)$$

### 1.3.2 Constrained Optimization

**Theorem 1.3.1.** *Fritz-John Condition*

*For constrained optimization problem*

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0, i = 1, \dots, n \\ & h_i(x) = 0, i = 1, \dots, m \end{aligned} \quad (1.10)$$

Denote  $I(x) = \{i \in \{1, \dots, n\} \mid g_i(x) = 0\}$ . For  $x \in \mathcal{S}$ ,  $f$  and  $g_i, i \in I(x)$  is differentiable at  $x$ ,  $h_j(x)$  is continuously differentiable at  $x$ . If  $x$  is local optimal, then there exists non-trivial  $\lambda_0, \lambda_i \geq 0, i \in I(x)$  and  $\mu_j$ , such that

$$\lambda_0 \nabla f(x) - \sum_{i \in I(x)} \lambda_i \nabla g_i(x) - \sum_{j=1}^m \mu_j \nabla h_j(x) = 0 \quad (1.11)$$

*Proof.* (i) If  $\{\nabla h_j(x)\}$  is linearly dependent, then there exists non-trivial  $\mu_j$ , such that

$$\sum_{j=1}^m \mu_j \nabla h_j(x) = 0 \quad (1.12)$$

Let  $\lambda_0, \lambda_i, i \in I(x) = 0$ , then (1.10) holds.

(ii) If  $\{\nabla h_j(x)\}$  is linearly independent, Denote

$$F_g = F(x, g) = \{d \mid \nabla g_i(x)^T d > 0, i \in I(x)\} \quad (1.13)$$

$$F_h = F(x, h) = \{d \mid \nabla h_j(x)^T d = 0, j = 1, \dots, m\} \quad (1.14)$$

If  $x$  is a optimal value, then apparently  $F(x, \mathcal{S}) \cap D(x, f) = \emptyset$ . Due to the independence of  $\{\nabla h_j(x)\}$ , we have  $F_g \cap F_h \subset F(x, \mathcal{S})$ , then

$$F_g \cap F_h \cap D(x, f) = \emptyset \quad (1.15)$$

that is

$$\begin{cases} \nabla f(x)^T d < 0 \\ \nabla g_i(x)^T d > 0, i \in I(x) \\ \nabla h_j(x)^T d = 0, j = 1, \dots, m \end{cases} \quad (1.16)$$

has no solution. Let

$$A = \{\nabla f(x)^T, -\nabla g_i(x)^T, i \in I(x)\} \quad (1.17)$$

$$B = \{-\nabla h_j(x)\}, j = 1, \dots, m \quad (1.18)$$

Then (21) is equivalent to

$$\begin{cases} A^T d < 0 \\ B^T d = 0 \end{cases} \quad (1.19)$$

has no solution.

Denote

$$S_1 = \left\{ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \mid y_1 = A^T d, y_2 = B^T d, d \in \mathbb{R}^n \right\} \quad (1.20)$$

$$S_2 = \left\{ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \mid y_1 < 0, y_2 = 0 \right\} \quad (1.21)$$

$S_1, S_2$  are non-trivial convex sets, and  $S_1 \cap S_2 = \emptyset$ . From *Hyperplane Separation Theorem*:  $\exists \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ , such that

$$p_1^T A^T d + p_2^T B^T d \geq p_1^T y_1 + p_2^T y_2, \forall d \in \mathbb{R}^n, \forall \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in CL(S_2) \quad (1.22)$$

Let  $y_2 = 0, d = 0, y_1 < 0$ , we have

$$p_1 \geq 0 \quad (1.23)$$

Let  $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in CL(S_2)$  So that

$$(p_1^T A^T + p_2^T B^T) d \geq 0 \quad (1.24)$$

$$(Ap_1 + Bp_2)^T d \geq 0 \quad (1.25)$$

#### 4 Chapter 1 Introduction to Optimization Algorithms

Let  $d = -(Ap_1 + Bp_2)$ , we have

$$Ap_1 + Bp_2 = 0 \quad (1.26)$$

From above, we have

$$\begin{cases} Ap_1 + Bp_2 = 0 \\ p_1 \geq 0 \end{cases} \quad (1.27)$$

Let  $p_1 = \{\lambda_0, \dots, \lambda_{I(x)}\}$ ,  $p_2 = \{\mu_1, \dots, \mu_m\}$ , i.e.,

$$\begin{cases} \lambda_0 \nabla f(x) - \sum_{i \in I(x)} \lambda_i \nabla g_i(x) - \sum_{j=1}^m \mu_j \nabla h_j(x) = 0 \\ \lambda_i \geq 0 \end{cases} \quad (1.28)$$

□

#### **Theorem 1.3.2.** *Kuhn-Tucker Condition*

*For constrained optimization problem*

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0, i = 1, \dots, n \\ & h_i(x) = 0, i = 1, \dots, m \end{aligned} \quad (1.29)$$

Denote  $I(x) = \{i \in \{1, \dots, n\} | g_i(x) = 0\}$ . For  $x \in \mathcal{S}$ ,  $f$  and  $g_i, i \in I(x)$  is differentiable at  $x$ ,  $h_j(x)$  is continuously differentiable at  $x$ .  $\{\nabla g_i(x), i \in I(x); \nabla h_j(x), j = 1, \dots, m\}$  is linearly independent. If  $x$  is local optimal, then  $\exists \lambda_i \geq 0$  and  $\mu_j$ , such that

$$\nabla f(x) - \sum_{i \in I(x)} \lambda_i \nabla g_i(x) - \sum_{j=1}^m \mu_j \nabla h_j(x) = 0 \quad (1.30)$$

**Remark 1** (K-T condition). *The equation (1.3.2) can be rewritten as*

$$\nabla f(x) - \sum_{i=1}^m \lambda_i \nabla g_i(x) - \sum_{j=1}^m \mu_j \nabla h_j(x) = 0 \quad (1.31)$$

where  $\lambda_i = 0, i \notin I(x)$ . i.e.,

$$\lambda_i g_i(x) = 0, i = 1, \dots, m \quad (1.32)$$

Denote

$$\mathcal{L}(\mathbf{x}, \lambda, \mu) = f(x) - \sum_{i=1}^m \lambda_i g_i(x) - \sum_{j=1}^m \mu_j h_j(x) \quad (1.33)$$



as the Lagrange function, then the K-T condition can be formulated as

$$(K - T) \begin{cases} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda, \mu) = 0 \\ \nabla_{\lambda} \mathcal{L}(\mathbf{x}, \lambda, \mu) = 0 \\ \nabla_{\mu} \mathcal{L}(\mathbf{x}, \lambda, \mu) = 0 \\ \lambda_i \geq 0, i = 1, \dots, m \\ \lambda_i g_i(\mathbf{x}) = 0, i = 1, \dots, m \end{cases} \quad (1.34)$$

## 1.4 Descent function

**Definition 1.4.1.** *Descent function.* Denote solution set  $\Omega \in X$ ,  $\mathcal{A}$  is an algorithm on  $X$ ,  $\psi : X \rightarrow \mathbb{R}$ . If

$$\psi(y) < \psi(x), \quad \forall x \notin \Omega, y \in \mathcal{A}(x) \quad (1.35)$$

$$\psi(y) \leq \psi(x), \quad \forall x \in \Omega, y \in \mathcal{A}(x) \quad (1.36)$$

Then  $\psi$  is a **descent function** of  $(\Omega, \mathcal{A})$ .

## 1.5 Convergence of Algorithm

**Theorem 1.5.1.**  $\mathcal{A}$  is an algorithm on  $X$ ,  $\Omega$  is the solution set,  $x^{(0)} \in X$ . If  $x^{(k)} \in \Omega$ , then the iteration stops. Otherwise set  $x^{(k+1)} = \mathcal{A}(x^{(k)})$ ,  $k := k + 1$ . If

- $\{x^{(k)}\}$  in a compact subset of  $X$
- There exists a continuous function  $\psi$ ,  $\psi$  is a descent function of  $(\Omega, \mathcal{A})$
- $\mathcal{A}$  is closed on  $\Omega^C$

Then, any convergent subsequence of  $\{x^{(k)}\}$  converges to  $x, x \in \Omega$ .

*Proof.*

□

### 1.5.1 Search Methods

#### 1.5.1.1 Line Search

Generate  $d^{(k)}$  from  $x^{(k)}$ ,

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)} \quad (1.37)$$

. search  $\alpha_k$  in 1-D space.

#### 1.5.1.2 Trust Region

Generate local model  $Q_k(s)$  of  $x^{(k)}$ ,

$$s^{(k)} = \arg \min Q_k(s) \quad (1.38)$$

$$x^{(k+1)} = x^{(k)} + s^{(k)} \quad (1.39)$$



# 2

## Unconstrained Optimization

### 2.1 Gradient Based Methods

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2.1)$$

---

**Algorithm 1:** Example of gradient based algorithm

---

**Data:** Solution set  $\Omega$ , cost function  $f$

$x^{(0)} \in \mathbb{R}^n, k := 0;$

**while**  $x^{(k)} \notin \Omega$  **do**

$d^{(k)} = -H_k \nabla f(x^{(k)})$ , ( $H_k$  is a positive definite symmetrical matrix);

    solve  $\min_{\alpha_k \geq 0} f(x^{(k)} + \alpha_k d^{(k)})$ ;

$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, k := k + 1$

**end**

---

#### 2.1.1 Determine Search Direction

##### 2.1.1.1 First-order gradient method

For unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2.2)$$

We have

$$f(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + O(\|x - x^{(k)}\|^2) \quad (2.3)$$

Set  $d^{(k)} = -\nabla f(x^{(k)})$ , when  $\alpha_k$  is sufficiently small,

$$f(x^{(k)} + \alpha_k d^{(k)}) < f(x^{(k)}) \quad (2.4)$$

##### 2.1.1.2 Second-order gradient method – Newton Direction

$$f(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) \quad (2.5)$$

$$+ \frac{1}{2} (x - x^{(k)})^T \nabla^2 f(x^{(k)}) (x - x^{(k)}) + O(\|x - x^{(k)}\|^3) \quad (2.6)$$

Set  $d^{(k)} = -G_k^{-1} \nabla f(x^{(k)})$ , where  $G_k = \nabla^2 f(x^{(k)})$ , i.e., Hesse matrix of  $f$  at  $x^{(k)}$ .

### 2.1.2 Determine Step Factor – Line Search

$$\min_{\alpha \geq 0} \varphi(\alpha) = f(x^{(k)} + \alpha d^{(k)}) \quad (2.7)$$

#### 2.1.2.1 Exact Line Search

Solve Line Search problem in finite iterations.

#### 2.1.2.2 Inexact Line Search

In some cases, the exact solution of Line Search is not necessary, so we can use inexact line search to improve algorithm efficiency.

*Goldstein Conditions*

$$\varphi(\alpha) \leq \varphi(0) + \rho \alpha \varphi'(0) \quad (2.8)$$

$$\varphi(\alpha) \geq \varphi(0) + (1 - \rho) \alpha \varphi'(0) \quad (2.9)$$

where  $\rho \in (\frac{1}{2}, 1)$  is a fixed parameter.

However, the downside of Goldstein Conditions is that the optimal value might not lie in the valid area.

*Wolfe-Powell Conditions*

$$\varphi(\alpha) \leq \varphi(0) + \rho \alpha \varphi'(0) \quad (2.10)$$

$$\varphi'(\alpha) \geq \sigma \varphi'(0) \quad (2.11)$$

where  $\sigma \in (\rho, 1)$ .

### 2.1.3 Global Convergence

**Theorem 2.1.1.** Assume  $\nabla f(x)$  exists and uniformly continuous on level set  $L(x^{(0)}) = \{x | f(x) \leq f(x^{(0)})\}$ . Denote  $\theta^{(k)}$  as the angle between  $d^{(k)}$  and  $-\nabla f(x^{(k)})$ .

$$\theta^{(k)} \leq \frac{\pi}{2} - \mu \quad (2.12)$$

If step factor is determined by following methods

- Exact Line Search
- Goldstein Conditions
- Wolfe-Powell Conditions

Then, there exists  $k$ , such that  $\nabla f(x^{(k)}) = 0$ , or  $f(x^{(k)}) \rightarrow 0$  or  $f(x^{(k)}) \rightarrow -\infty$ .

*Proof.*

□

### 2.1.4 Steepest Descent Method

Steepest Descent Method is a Line Search Method.

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)}) \quad (2.13)$$

---

**Algorithm 2:** Steepest Descent Algorithm

---

**Data:** Termination error  $\epsilon$ , cost function  $f$

$x^{(0)} \in \mathbb{R}^n, k := 0;$

**while**  $\|g^{(k)}\| \geq \epsilon$  **do**

$d^{(k)} = -g^{(k)};$

    solve  $\min_{\alpha_k \geq 0} f(x^{(k)} + \alpha_k d^{(k)});$

$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, k := k + 1;$

    Compute  $g^{(k)} = \nabla f(x^{(k)})$

**end**

---

Steepest Descent Method has linear convergence rate generally.

### 2.1.5 Newton Method

Newton Method is also a Line Search Method.

$$f(x^{(k)} + s) \approx q^{(k)}(s)f(x^{(k)}) + g^{(k)T}s + \frac{1}{2}s^T G_k s \quad (2.14)$$

where  $g^{(k)} = \nabla f(x^{(k)})$ ,  $G_k = \nabla^2 f(x^{(k)})$ . To minimize  $q^{(k)}(s)$ , we have

$$s = G_k^{-1} g^{(k)} \quad (2.15)$$

Notice that  $G_k^{-1} g^{(k)}$  is the Newton Direction.

*Analysis on quadratic function*

For positive definite quadratic function

$$f(x) = \frac{1}{2}x^T Gx - c^T x \quad (2.16)$$

In this case,  $\nabla^2 f(x) = G$ . Let  $H_0 = G^{-1}$ , then we have

$$d^{(0)} = H_0 \nabla f(x^{(0)}) \quad (2.17)$$

$$= G^{-1}(Gx^{(0)} - c) \quad (2.18)$$

$$= x^{(0)} - G^{-1}c \quad (2.19)$$

$$= x^{(0)} - x^* \quad (2.20)$$

So that Newton Method can reach global optimal in 1 iteration for quadratic functions.

For general non-linear functions, if we follow

$$x^{(k+1)} = x^{(k)} - G_k^{-1} g^{(k)} \quad (2.21)$$

we called it Newton Method.

#### Convergence Rate of Newton Method

**Theorem 2.1.2.**  $f \in \mathcal{C}^2$ ,  $x^{(k)}$  is sufficiently closed to optimal point  $x^*$ , where  $\nabla f(x^*) = 0$ . If  $\nabla^2 f(x^*)$  is positive definite, Hesse matrix of  $f$  satisfies Lipschitz Condition, i.e.,  $\exists \beta > 0$ , such that for all  $(i, j)$ ,

$$|G_{ij}(x) - G_{ij}(y)| \leq \beta \|x - y\| \quad (2.22)$$

Then  $\{x^{(k)}\} \rightarrow x^*$ , and have quadratic convergence rate.

*Proof.* Denote  $g(x) = \nabla f(x)$ , then we have

$$g(x - h) = g(x) - G(x)h + O(\|h\|^2) \quad (2.23)$$

Let  $x = x^{(k)}$ ,  $h = h^{(k)} = x^{(k)} - x^*$ , then

$$g(x^*) = g(x^{(k)}) - G(x^{(k)})(h^{(k)}) + O(\|h^{(k)}\|^2) = 0 \quad (2.24)$$

From Lipschitz Condition, we can easily get  $G(x^{(k)})^{-1}$  is finite. Then we left multiply  $G(x^{(k)})^{-1}$  to Equation (2.24)

$$0 = G(x^{(k)})^{-1}g(x^{(k)}) - h^{(k)} + O(\|h^{(k)}\|^2) \quad (2.25)$$

$$= x^* - x^{(k)} + G(x^{(k)})^{-1}g(x^{(k)}) + O(\|h^{(k)}\|^2) \quad (2.26)$$

$$= x^* - x^{(k+1)} + O(\|h^{(k)}\|^2) \quad (2.27)$$

$$= -h^{(k+1)} + O(\|h^{(k)}\|^2) \quad (2.28)$$

i.e.,

$$\|h^{(k+1)}\| = O(\|h^{(k)}\|^2) \quad (2.29)$$

□

#### 2.1.6 Quasi-Newton Method

Newton Method has a fast convergence rate. However, Newton Method requires second-order derivative, if Hesse matrix is not positive definite, Newton Method might not work well.

In order to overcome the above difficulties, Quasi-Newton Method is introduced. Its basic idea is that: Using second-order derivative free matrix  $H_k$  to approximate  $G(x^{(k)})^{-1}$ . Denote  $s^{(k)} = x^{(k+1)} - x^{(k)}$ ,  $y^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$ , then we have

$$\nabla^2 f(x^{(k)})s^{(k)} \approx y^{(k)} \quad (2.30)$$

or

$$\nabla^2 f(x^{(k)})^{-1}y^{(k)} \approx s^{(k)} \quad (2.31)$$

So we need to construct  $H_{k+1}$  such that

$$H_{k+1}y^{(k)} \approx s^{(k)} \quad (2.32)$$

or

$$y^{(k)} \approx B_{k+1}s^{(k)} \quad (2.33)$$

we called (2.32), (2.33) *Quasi-Newton Conditions* or *Secant Conditions*.

---

**Algorithm 3:** Quasi-Newton Algorithm

---

**Data:** Cost function  $f$

$x^{(0)} \in \mathbb{R}^n, H_0 = I, k := 0;$

**while** *some conditions* **do**

$d^{(k)} = -H_k g^{(k)};$   
 solve  $\min_{\alpha_k \geq 0} f(x^{(k)} + \alpha_k d^{(k)});$   
 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)};$   
 generate  $H_{k+1}, k := k + 1$

**end**

---

### 2.1.6.1 How to generate $H_k$

$H_k$  is the approximation matrix in  $k$ th iteration, we want to generate  $H_{k+1}$  from  $H_k$

**Symmetric Rank 1 Update** Assume

$$H_{k+1} = H_k + a\mathbf{u}\mathbf{u}^T, \quad a \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^n \quad (2.34)$$

From the Quasi-Newton Conditions, we have

$$H_{k+1}\mathbf{y}^{(k)} = \mathbf{s}^{(k)} \quad (2.35)$$

$$H_k\mathbf{y}^{(k)} + a\mathbf{u}\mathbf{u}^T\mathbf{y}^{(k)} = \mathbf{s}^{(k)} \quad (2.36)$$

$$H_k\mathbf{y}^{(k)} + a\mathbf{u}^T\mathbf{y}^{(k)}\mathbf{u} = \mathbf{s}^{(k)} \quad (2.37)$$

Let  $\mathbf{u} = \mathbf{s}^{(k)} - H_k\mathbf{y}^{(k)}, a = \frac{1}{\mathbf{u}^T\mathbf{y}^{(k)}}$ , clearly this is a solution of the equation. Here we have

$$H_{k+1} = \frac{(\mathbf{s}^{(k)} - H_k\mathbf{y}^{(k)})(\mathbf{s}^{(k)} - H_k\mathbf{y}^{(k)})^T}{(\mathbf{s}^{(k)} - H_k\mathbf{y}^{(k)})^T\mathbf{y}^{(k)}} \quad (2.38)$$

(2.38) is *Symmetric Rank 1 Update*. The problem of Symmetric Rank 1 Update is that the positive-definite property of  $H_k$  can not be preserved.

**Symmetric Rank 2 Update** Assume

$$H_{k+1} = H_k + a\mathbf{u}\mathbf{u}^T + b\mathbf{v}\mathbf{v}^T, \quad a, b \in \mathbb{R}, \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \quad (2.39)$$

such that Quasi-Newton Conditions stand. We can find a solution of  $a, b, \mathbf{u}, \mathbf{v}$  that is

$$\begin{cases} \mathbf{u} = \mathbf{s}^{(k)}, & a\mathbf{u}^T \mathbf{y} = 1 \\ \mathbf{v} = H_k \mathbf{y}^{(k)}, & b\mathbf{v}^T \mathbf{y} = -1 \end{cases} \quad (2.40)$$

So that we have

$$H_{k+1} = H_k + \frac{\mathbf{s}^{(k)} \mathbf{s}^{(k)T}}{\mathbf{s}^{(k)T} \mathbf{y}^{(k)}} - \frac{H_k \mathbf{y}^{(k)} \mathbf{y}^{(k)T} H_k}{\mathbf{y}^{(k)T} H_k \mathbf{y}^{(k)}} \quad (2.41)$$

We called (2.41) the DFP (Davidon-Fletcher-Powell) update.

From Quasi-Newton Condition (2.33), we can get the BFGS (Broyden-Fletcher-Goldfarb-Shanno) update

$$B_{k+1}^{(BFGS)} = B_k + \frac{\mathbf{y}^{(k)} \mathbf{y}^{(k)T}}{\mathbf{y}^{(k)T} \mathbf{s}^{(k)}} - \frac{B_k \mathbf{s}^{(k)} \mathbf{s}^{(k)T} B_k}{\mathbf{s}^{(k)T} B_k \mathbf{s}^{(k)}} \quad (2.42)$$

*Inverse of SRI update*

**Theorem 2.1.3** (Sherman-Morrison).  $A \in \mathbb{R}^n \times \mathbb{R}^n$  is a non-singular matrix,  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ . If  $1 + \mathbf{v}^T A^{-1} \mathbf{u} \neq 0$ , then SRI update of  $A$  is non-singular, and its inverse can be represented as

$$(A + a\mathbf{u}\mathbf{v}^T)^{-1} = A^{-1} - \frac{A^{-1} \mathbf{u} \mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1} \mathbf{u}} \quad (2.43)$$

### 2.1.7 Conjugate Gradient Method

**Definition 2.1.1.** *Conjugate Direction.*  $G$  is a  $n \times n$  positive definite matrix, for non-zero vector set  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\} \in \mathbb{R}^n$ , if  $\mathbf{d}^{(i)T} G \mathbf{d}^{(j)} = 0, (i \neq j)$ , then we called  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\}$  is  $G$ -Conjugate.

**Lemma 2.1.4.** *For non-zero conjugate vector set  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\} \in \mathbb{R}^n$ ,  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\}$  are linearly independent.*

*Proof.* From Definition 2.1.1, we have

$$\mathbf{d}^{(i)T} G \mathbf{d}^{(j)} = 0, \forall i, j, i \neq j \quad (2.44)$$

if  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\}$  is linearly dependent, there exists

$$\mathbf{d}^{(t)} = \sum_{j=0}^k c_j \mathbf{d}^{(j)} \quad (2.45)$$

then

$$\mathbf{d}^{(t)T} G \mathbf{d}^{(i)} = \sum_{j=0}^k c_j \mathbf{d}^{(j)T} G \mathbf{d}^{(i)} = c_i \mathbf{d}^{(i)T} G \mathbf{d}^{(i)} \neq 0 \quad (2.46)$$



so that  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\}$  are linearly independent. □

---

**Algorithm 4:** Conjugate Gradient Algorithm

---

**Data:** Cost function  $f$

$x^{(0)} \in \mathbb{R}^n$ , positive definite matrix  $G$ ,  $k := 0$ ;

Construct  $\mathbf{d}^{(0)}$  such that  $\mathbf{g}^{(0)T} \mathbf{d}^{(0)} < 0$ ;

**while** *some conditions* **do**

solve  $\min_{\alpha_k \geq 0} f(\mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)})$ ;

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ ;

Construct  $\mathbf{d}^{(k+1)}$  such that  $\mathbf{d}^{(k+1)T} G \mathbf{d}^{(j)} = 0, j = 0, \dots, k$ ;

$k := k + 1$

**end**

---

**Theorem 2.1.5** (Conjugate Gradient). *For strictly convex quadratic function*

$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x}$ , *apply conjugate gradient method combined with exact line search, then  $\mathbf{x}^{(k+1)}$  is the global minima in manifold*

$$\mathcal{V} = \{\mathbf{x} | \mathbf{x} = \mathbf{x}^{(0)} + \sum_{j=0}^k \beta_j \mathbf{d}^{(j)}, \forall \beta_j \in \mathbb{R}\} \quad (2.47)$$

*Proof.* Firstly, from Lemma 2.1.6, we have  $\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\}$  are linearly independent. So we only need to prove that for all  $k < n$

$$\mathbf{g}^{(k+1)T} \mathbf{d}^{(j)} = 0, j = 0, \dots, k \quad (2.48)$$

i.e.,  $\mathbf{g}^{(k+1)}$  is orthogonal with subspace  $\text{span}\{\mathbf{d}^{(0)}, \dots, \mathbf{d}^{(k)}\}$ .

Due to the exact line search,  $\forall j$

$$\mathbf{g}^{(j+1)T} \mathbf{d}^{(j)} = 0 \quad (2.49)$$

especially  $\mathbf{g}^{(k+1)T} \mathbf{d}^{(k)} = 0$ .

Notice that

$$\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} = G(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \alpha_k G \mathbf{d}^{(k)} \quad (2.50)$$

so that we have  $\forall j \leq k$

$$\mathbf{g}^{(k+1)T} \mathbf{d}^{(j)} = \left( \sum_{m=j+1}^k (\mathbf{g}^{(m+1)T} - \mathbf{g}^{(m)T}) + \mathbf{g}^{(j+1)T} \right) \mathbf{d}^{(j)} \quad (2.51)$$

$$= \sum_{m=j+1}^k \alpha_m \mathbf{d}^{(m)T} G \mathbf{d}^{(j)} + \mathbf{g}^{(j+1)T} \mathbf{d}^{(j)} \quad (2.52)$$

$$= 0 \quad (2.53)$$

□

**Lemma 2.1.6.** For strictly convex quadratic function  $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x}$ , apply conjugate gradient method combined with exact line search,  $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = G \mathbf{x} + \mathbf{c}$ , we have

$$\mathbf{g}^{(k)T} \mathbf{g}^{(j)} = 0, \forall j = 0, \dots, k-1 \quad (2.54)$$

*Proof.* From Theorem 2.1.5, we have

$$\mathbf{g}^{(k)T} \mathbf{g}^{(j)} = \mathbf{g}^{(k)T} (-\mathbf{d}^{(j)} + \sum_{i=0}^{j-1} \beta_i^{(j)} \mathbf{d}^{(i)}) = 0 \quad (2.55)$$

□

### 2.1.7.1 Quadratic function case

For  $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x}$ ,  $G$  is a  $n \times n$  positive definite matrix.

$$\mathbf{g}(\mathbf{x}) = G \mathbf{x} + \mathbf{c} \quad (2.56)$$

Set  $\mathbf{d}^{(0)} = -\mathbf{g}^{(0)}$ , exact line search for  $\alpha_0$  such that  $\mathbf{g}^{(1)T} \mathbf{d}^{(0)} = 0$ . Assume  $\mathbf{d}^{(1)} = -\mathbf{g}^{(1)} + \beta_0^{(1)} \mathbf{d}^{(0)}$ , select  $\beta_0^{(1)}$  such that  $\mathbf{d}^{(1)T} G \mathbf{d}^{(0)} = 0$

$$\beta_0^{(1)} = \frac{\mathbf{g}^{(1)T} \mathbf{g}^{(1)}}{\mathbf{g}^{(0)T} \mathbf{g}^{(0)}} \quad (2.57)$$

*Proof.* From (92), we have

$$\mathbf{d}^{(1)T} G \mathbf{d}^{(0)} = 0 \quad (2.58)$$

$$\Leftrightarrow \mathbf{d}^{(1)T} (\mathbf{g}^{(1)} - \mathbf{g}^{(0)}) = 0 \quad (2.59)$$

$$\Leftrightarrow (\mathbf{g}^{(1)} + \beta_0^{(1)} \mathbf{g}^{(0)})^T (\mathbf{g}^{(1)} - \mathbf{g}^{(0)}) = 0 \quad (2.60)$$

$$\Leftrightarrow \mathbf{g}^{(1)T} \mathbf{g}^{(1)} - \beta_0^{(1)} \mathbf{g}^{(0)T} \mathbf{g}^{(0)} = 0 \quad (2.61)$$

$$\Leftrightarrow \beta_0^{(1)} = \frac{\mathbf{g}^{(1)T} \mathbf{g}^{(1)}}{\mathbf{g}^{(0)T} \mathbf{g}^{(0)}} \quad (2.62)$$

□

Generally, we can select  $\beta_j^{(k)}$  such that  $\mathbf{d}^{(k)T} G \mathbf{d}^{(j)} = 0, j = 0, 1, \dots, k-1$  that is

$$\mathbf{d}^{(k)T} G \mathbf{d}^{(j)} = 0 \quad (2.63)$$

$$(-\mathbf{g}^{(k)T} + \sum_{i=0}^{k-1} \beta_i^{(k)} \mathbf{d}^{(i)T}) G \mathbf{d}^{(j)} = 0 \quad (2.64)$$

$$-\mathbf{g}^{(k)T} G \mathbf{d}^{(j)} + \beta_j^{(k)} \mathbf{d}^{(j)T} G \mathbf{d}^{(j)} = 0 \quad (2.65)$$

so we have

$$\beta_j^{(k)} = \frac{\mathbf{g}^{(k)T} G \mathbf{d}^{(j)}}{\mathbf{d}^{(j)T} G \mathbf{d}^{(j)}} = \frac{\mathbf{g}^{(k)T} (\mathbf{g}^{(j+1)} - \mathbf{g}^{(j)})}{\mathbf{d}^{(j)T} (\mathbf{g}^{(j+1)} - \mathbf{g}^{(j)})} \quad (2.66)$$

From Lemma 2.1.6, we have

$$\mathbf{g}^{(k)T} \mathbf{g}^{(j)} = 0, \forall j = 0, \dots, k-1 \quad (2.67)$$

So

$$\beta_j^{(k)} = 0, j = 0, \dots, k-2 \quad (2.68)$$

$$\beta_{k-1}^{(k)} = \frac{\mathbf{g}^{(k)T} (\mathbf{g}^{(k)} - \mathbf{g}^{(k-1)})}{\mathbf{g}^{(k-1)T} (\mathbf{g}^{(k)} - \mathbf{g}^{(k-1)})} \quad (2.69)$$

## 2.2 Trust Region Method

Previously, we use a direction search strategy to determine a search direction, then use line search method to determine step length.

Now we discuss a new global convergence strategy – Trust-Region Method.

**Definition 2.2.1** (Trust Region).

$$\Omega_k = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}^{(k)}\| \leq e_k\} \quad (2.70)$$

We called  $\Omega_k$  Trust Region,  $e_k$  is the Trust radius.

Suppose in this neighborhood, quadratic model  $q^{(k)}(\mathbf{s})$  is a proper approximation of  $f(\mathbf{x})$ . We minimize the quadratic model in trust region, derive approximate minima  $\mathbf{s}^{(k)}$ , and set  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$ .

### 2.2.1 Trust Region Subproblem

$$\min_{\|\mathbf{s}\| \leq e_k} q^{(k)}(\mathbf{s}) = f(\mathbf{x}^{(k)}) + \mathbf{g}^{(k)T} \mathbf{s} + \frac{1}{2} \mathbf{s}^T B_k \mathbf{s} \quad (2.71)$$

Where  $\mathbf{s} = \mathbf{x} - \mathbf{x}^{(k)}$ ,  $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)})$ ,  $B_k = \nabla^2 f(\mathbf{x}^{(k)})$ .  $e_k$  is the trust region radius.

**2.2.2 How to select  $e_k$** 

Denote the solution of the subproblem as  $\mathbf{s}^{(k)}$ , then let

$$\text{Act}_k = f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)}) \quad (2.72)$$

$$\text{Pre}_k = q^{(k)}(\mathbf{0}) - q^{(k)}(\mathbf{s}^{(k)}) \quad (2.73)$$

Define

$$r_k = \frac{\text{Act}_k}{\text{Pre}_k} = \frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})}{q^{(k)}(\mathbf{0}) - q^{(k)}(\mathbf{s}^{(k)})} \quad (2.74)$$

to measure the difference between objective function and the quadratic approximate model.

We can update  $e_k$  according to  $r_k$ . If  $r_k$  is too small, that means our model can not fit the objective function well, so we need to decrease  $e_k$ . If  $r_k$  is close to 1, that means our model is good and we can increase  $r_k$ . Set the parameters  $0 < \gamma_1 < \gamma_2 < 1$  and  $0 < \eta_1 < 1 < \eta_2$ , we can have the following update rule

$$e_{k+1} = \begin{cases} \eta_1 e_k & \text{if } r_k < \gamma_1 \\ e_k & \text{if } \gamma_1 < r_k < \gamma_2 \\ \min(\eta_2 e_k, \bar{e}) & \text{if } r_k \geq \gamma_2 \end{cases} \quad (2.75)$$

---

**Algorithm 5:** Trust Region Algorithm

---

**Data:** Cost function  $f$

$\mathbf{x}^{(0)} \in \mathbb{R}^n$ ,  $e_0 \in (0, \bar{e})$ ,  $\epsilon > 0$ ,  $0 < \gamma_1 < \gamma_2 < 1$ ,  $0 < \eta_1 < 1 < \eta_2$ ,  $k := 0$ ;

**while**  $\|\mathbf{g}^{(k)}\| \geq \epsilon$  **do**

    solve the subproblem to derive  $\mathbf{s}^{(k)}$ ;

    calculate  $r_k$ , update  $\mathbf{x}$ ;

**if**  $r_k > 0$  **then**

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$

**else**

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$

**end**

    update  $e_k$  following (117);

$k := k + 1$ ;

**end**

---

# 3

## Constrained Optimization

### 3.1 Quadratic Programming

$$\begin{aligned} \min \quad & Q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_i^T \mathbf{x} = b_i, i \in \mathcal{E} = \{1, \dots, m_e\} \\ & \mathbf{a}_i^T \mathbf{x} \geq b_i, i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \quad (3.1)$$

We assume that  $G$  is a symmetric matrix and  $\mathbf{a}_i, i \in \mathcal{E}$  be linearly independent.

#### 3.1.1 Solution of Quadratic Programming

If  $G$  be positive semi-definite matrix, the Quadratic Programming problem is a convex optimization problem, so any of its local minima is a global minima.

If  $G$  be positive definite matrix, the solution to the Quadratic Programming problem is unique, if exists.

If  $G$  be indefinite, there is no guarantee to the solution.

#### 3.1.2 Equality Constrained Quadratic Programming

$$\begin{aligned} \min \quad & Q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \end{aligned} \quad (3.2)$$

#### 3.1.3 General Quadratic Programming

$$\begin{aligned} \min \quad & Q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{a}_i^T \mathbf{x} = b_i, i \in \mathcal{E} = \{1, \dots, m_e\} \\ & \mathbf{a}_i^T \mathbf{x} \geq b_i, i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \quad (3.3)$$

The idea is to remove or transform the inequality constraints. If the inequality constraint is not active near the solution, we can ignore the constraint; For the active inequality constraints, we can use equality constraints to replace them.

**Theorem 3.1.1** (Active Set). *Denote  $\mathbf{x}^*$  as a local minima of general quadratic problem (3.3), then  $\mathbf{x}^*$  must be a local minima of the equality constrained problem*

$$(\text{EQ}) \begin{cases} \min \quad Q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T G \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad \mathbf{a}_i^T \mathbf{x} = b_i, i \in \mathcal{E} \cup \mathcal{I}(\mathbf{x}^*) \end{cases} \quad (3.4)$$

Meanwhile, if  $\mathbf{x}^*$  is a feasible point of (3.3), and the K-T point of (EQ),  $\lambda_i^* \geq 0, i \in \mathcal{I}(\mathbf{x}^*)$ , then  $\mathbf{x}^*$  must be the K-T point of (3.3).

*Proof.* Recall the K-T condition, we can get that there exists  $\lambda_i \geq 0, i \in \mathcal{I}(\mathbf{x}^*)$  and  $\mu_j$  s.t.

$$\nabla Q(\mathbf{x}^*) - \sum_{i \in \mathcal{I}(\mathbf{x}^*)} \lambda_i \mathbf{a}_i - \sum_{j \in \mathcal{E}} \mu_j \mathbf{a}_j = 0 \quad (3.5)$$

the K-T condition of (EQ) is there exists  $\lambda_i, i \in \mathcal{E} \cup \mathcal{I}(\mathbf{x}^*)$ , s.t.

$$\nabla Q(\mathbf{x}^*) - \sum_{j \in \mathcal{E} \cup \mathcal{I}(\mathbf{x}^*)} \lambda_j \mathbf{a}_j = 0 \quad (3.6)$$

Apparently If  $\mathbf{x}^*$  satisfies (3.5), then it also satisfies (3.6). On the other hand, if  $\mathbf{x}^*$  satisfies (3.6) and  $\lambda_i \geq 0, i \in \mathcal{I}(\mathbf{x}^*)$ , we have

$$\nabla Q(\mathbf{x}^*) - \sum_{j \in \mathcal{E} \cup \mathcal{I}(\mathbf{x}^*)} \lambda_j \mathbf{a}_j = 0 \quad (3.7)$$

$$\Leftrightarrow \nabla Q(\mathbf{x}^*) - \sum_{i \in \mathcal{I}(\mathbf{x}^*)} \lambda_i \mathbf{a}_i - \sum_{j \in \mathcal{E}} \lambda_j \mathbf{a}_j = 0 \quad (3.8)$$

i.e.,  $\mathbf{x}^*$  satisfies (3.5).

□

## 3.2 Equality Constrained Problem

### 3.2.1 Lagrange-Newton method

$$\min f(\mathbf{x}) \quad (3.9)$$

$$s.t. \mathbf{c}(\mathbf{x}) = \mathbf{0} \quad (3.10)$$

where  $\mathbf{c}(\mathbf{x}) = (c_1(\mathbf{x}), \dots, c_m(\mathbf{x}))^T$ .

Denote  $A(\mathbf{x}) = [\nabla \mathbf{c}(\mathbf{x})]^T = (\nabla c_1(\mathbf{x}), \dots, \nabla c_m(\mathbf{x}))^T$ . The K-T condition of the problem is there exists  $\lambda \in \mathbb{R}^m$  s.t.

$$\nabla f(\mathbf{x}) - A(\mathbf{x})^T \lambda = \mathbf{0} \quad (3.11)$$

and  $\mathbf{c}(\mathbf{x}) = \mathbf{0}$ .

We can use Newton-Raphson method to solve the equations by

$$\begin{pmatrix} W(\mathbf{x}, \lambda) & -A(\mathbf{x})^T \\ -A(\mathbf{x}) & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}) - A(\mathbf{x})^T \lambda \\ \mathbf{c}(\mathbf{x}) \end{pmatrix} \quad (3.12)$$

where  $W(\mathbf{x}, \lambda) = \nabla^2 f(\mathbf{x}) - \sum_{i=1}^m \lambda_i \nabla^2 c_i(\mathbf{x})$ .

We called the method above as *Lagrange-Newton Method*.

Here we can define

$$\psi(\mathbf{x}, \lambda) = \|\nabla f(\mathbf{x}) - A(\mathbf{x})^T \lambda\|^2 + \|\mathbf{c}(\mathbf{x})\|^2 \quad (3.13)$$

so that  $\psi$  is a descent function to Lagrange-Newton method.

$$\nabla \psi(\mathbf{x}, \lambda)^T \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = -2\psi(\mathbf{x}, \lambda) \neq 0 \quad (3.14)$$

### 3.2.2 Sequential Quadratic Programming method

(3.12) can be rewritten into

$$\begin{cases} W(\mathbf{x}, \lambda)\delta_x + \nabla f(\mathbf{x}) &= A(\mathbf{x})^T(\lambda + \delta_\lambda) \\ \mathbf{c}(\mathbf{x}) + A(\mathbf{x})\delta_x &= \mathbf{0} \end{cases} \quad (3.15)$$

From K-T condition, we notice that  $\delta_x$  is the K-T point of the following Quadratic Programming problem

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{d}^T W(\mathbf{x}, \lambda) \mathbf{d} + \nabla f(\mathbf{x})^T \mathbf{d} \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{x}) + A(\mathbf{x}) \mathbf{d} = \mathbf{0} \end{aligned} \quad (3.16)$$

So we can solve a Quadratic Programming subproblem to derive  $\delta_x$ , we called this method *Sequential Quadratic Programming*.

## 3.3 General Nonlinear Constrained Problem

### 3.3.1 Sequential Quadratic Programming method

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E} = \{1, \dots, m_e\} \\ & c_i(\mathbf{x}) \geq 0, \quad i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \quad (3.17)$$

Similarly, we can construct subproblem

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{d}^T W \mathbf{d} + \mathbf{g}^T \mathbf{d} \\ \text{s.t.} \quad & c_i(\mathbf{x}) + \mathbf{a}_i(\mathbf{x})^T \mathbf{d} = 0, i \in \mathcal{E} \\ & c_i(\mathbf{x}) + \mathbf{a}_i(\mathbf{x})^T \mathbf{d} \geq 0, i \in \mathcal{I} \end{aligned} \quad (3.18)$$

Here,  $W$  is the Hesse matrix (or its approximation) of the Lagrange function of (3.17),  $\mathbf{g} = \nabla f(\mathbf{x})$ ,  $A(\mathbf{x}) = (\mathbf{a}_1(\mathbf{x}), \dots, \mathbf{a}_m(\mathbf{x}))$ .

Denote the solution to subproblem (3.18) as  $\mathbf{d}$ , the corresponding Lagrange multiplier vector  $\bar{\lambda}$ , so we have

$$\begin{cases} W \mathbf{d} + \mathbf{g} = A(\mathbf{x})^T \bar{\lambda} \\ \bar{\lambda}_i \geq 0, i \in \mathcal{I} \\ \mathbf{c}(\mathbf{x}) + A(\mathbf{x}) \mathbf{d} = 0, i \in \mathcal{E} \\ \mathbf{c}(\mathbf{x}) + A(\mathbf{x}) \mathbf{d} \geq 0, i \in \mathcal{I} \end{cases} \quad (3.19)$$

### 3.3.2 Penalty method

For nonlinear constrained problem (3.17), we can use objective function  $f(\mathbf{x})$  and constraint function  $\mathbf{c}(\mathbf{x})$  to construct *Penalty function*

$$P(\mathbf{x}) = P(f(\mathbf{x}), \mathbf{c}(\mathbf{x})) \quad (3.20)$$

We need the penalty function have the property that: for feasible points,  $P(\mathbf{x}) = f(\mathbf{x})$ , otherwise,  $P(\mathbf{x}) > f(\mathbf{x})$ .

To measure the destructiveness to the constraints, we define  $\mathbf{c}(\mathbf{x})_-$

$$\begin{cases} c_i(\mathbf{x})_- = c_i(\mathbf{x}), & i \in \mathcal{E} \\ c_i(\mathbf{x})_- = |\min\{0, c_i(\mathbf{x})\}|, & i \in \mathcal{I} \end{cases} \quad (3.21)$$

Consider simple penalty function

$$P_\sigma(\mathbf{x}) = f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})_-\|^2 \quad (3.22)$$

Denote  $\mathbf{x}(\sigma)$  as the solution to unconstrained problem  $\min P_\sigma(\mathbf{x})$ , we have the following lemma:

**Lemma 3.3.1** (Penalty method). *If  $\mathbf{x}(\sigma)$  is a feasible point of nonlinear constrained problem (3.17), then  $\mathbf{x}(\sigma)$  aslo is the solution to (3.17).*

*Proof.* From the definition of penalty function, we have  $P(\mathbf{x}) = f(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{S}$ . If  $\mathbf{x}(\sigma)$  is the solution to  $\min P(\mathbf{x})$ , i.e.,

$$P(\mathbf{x}(\sigma)) \leq P(\mathbf{x}_0), \forall \mathbf{x}_0 \in \mathbb{R}^n \quad (3.23)$$

$$f(\mathbf{x}(\sigma)) \leq f(\mathbf{x}_0), \forall \mathbf{x}_0 \in \mathcal{S} \quad (3.24)$$

that is,  $\mathbf{x}(\sigma)$  is the solution to (3.17). □

---

#### Algorithm 6: Penalty Method Algorithm

---

**Data:** Cost function  $f$

$x^{(0)} \in \mathbb{R}^n, \sigma_0 > 0, \beta > 1, \epsilon > 0, k := 0;$

**while**  $\|\mathbf{c}(\mathbf{x}(\sigma_{k-1}))_-\| \geq \epsilon$  **do**

solve the subproblem  $\min_{\mathbf{x} \in \mathbb{R}^n} P_{\sigma_k}(\mathbf{x})$  to get the solution  $\mathbf{x}(\sigma_k);$

$\mathbf{x}^{(k+1)} = \mathbf{x}(\sigma_k), \sigma_{k+1} = \beta \sigma_k;$

$k := k + 1;$

**end**

**return:**  $\mathbf{x}(\sigma_{k-1})$

---

**Theorem 3.3.2** (Convergence of Penalty method). *If  $\epsilon > \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{c}(\mathbf{x})_-\|$ , then the algorithm can terminate in finite steps.*



**Lemma 3.3.3.** *Let  $\sigma_{k+1} > \sigma_k > 0$ , then we have  $P_{\sigma_k}(\mathbf{x}(\sigma_k)) \leq P_{\sigma_{k+1}}(\mathbf{x}(\sigma_{k+1}))$ ,  $\|\mathbf{c}(\mathbf{x}(\sigma_k))_-\| \geq \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|$ ,  $f(\mathbf{x}(\sigma_k)) \leq f(\mathbf{x}(\sigma_{k+1}))$ .*

*Proof.*

$$P_{\sigma_{k+1}}(\mathbf{x}(\sigma_{k+1})) = f(\mathbf{x}(\sigma_{k+1})) + \sigma_{k+1} \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2 \quad (3.25)$$

$$\geq f(\mathbf{x}(\sigma_{k+1})) + \sigma_k \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2 \quad (3.26)$$

$$\geq \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \sigma_k \|\mathbf{c}(\mathbf{x})_-\|^2 \quad (3.27)$$

$$= P_{\sigma_k}(\mathbf{x}(\sigma_k)) \quad (3.28)$$

From the definition, we have

$$f(\mathbf{x}(\sigma_k)) + \sigma_{k+1} \|\mathbf{c}(\mathbf{x}(\sigma_k))_-\|^2 \quad (3.29)$$

$$\geq f(\mathbf{x}(\sigma_{k+1})) + \sigma_{k+1} \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2 \quad (3.30)$$

$$\geq f(\mathbf{x}(\sigma_{k+1})) + \sigma_k \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2 \quad (3.31)$$

$$\geq f(\mathbf{x}(\sigma_k)) + \sigma_k \|\mathbf{c}(\mathbf{x}(\sigma_k))_-\|^2 \quad (3.32)$$

From the inequalities above, we have

$$\sigma_k (\|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2 - \|\mathbf{c}(\mathbf{x}(\sigma_k))_-\|^2) \quad (3.33)$$

$$\leq f(\mathbf{x}(\sigma_{k+1})) - f(\mathbf{x}(\sigma_k)) \quad (3.34)$$

$$\leq \sigma_{k+1} (\|\mathbf{c}(\mathbf{x}(\sigma_k))_-\|^2 - \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2) \quad (3.35)$$

So that

$$\|\mathbf{c}(\mathbf{x}(\sigma_k))_-\| \geq \|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\| \quad (3.36)$$

Then

$$0 \leq \sigma_k (\|\mathbf{c}(\mathbf{x}(\sigma_{k+1}))_-\|^2 - \|\mathbf{c}(\mathbf{x}(\sigma_k))_-\|^2) \leq f(\mathbf{x}(\sigma_{k+1})) - f(\mathbf{x}(\sigma_k)) \quad (3.37)$$

i.e.,

$$f(\mathbf{x}(\sigma_{k+1})) \geq f(\mathbf{x}(\sigma_k)) \quad (3.38)$$

□

**Lemma 3.3.4.** *Denote  $\bar{\mathbf{x}}$  as the solution to problem (3.17), then for all  $\sigma_k > 0$ ,*

$$f(\bar{\mathbf{x}}) \geq P_{\sigma_k}(\mathbf{x}(\sigma_k)) \geq f(\mathbf{x}(\sigma_k)) \quad (3.39)$$

*Proof.* For all  $\sigma_k > 0$ ,

$$f(\bar{\mathbf{x}}) = \min_{\mathbf{x} \in \mathbb{R}^n} \lim_{\sigma \rightarrow \infty} f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})_-\|^2 \quad (3.40)$$

$$\geq \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \sigma_k \|\mathbf{c}(\mathbf{x})_-\|^2 \quad (3.41)$$

$$= f(\mathbf{x}(\sigma_k)) + \sigma_k \|\mathbf{c}(\mathbf{x}(\sigma_k))_-\|^2 \quad (3.42)$$

$$\geq f(\mathbf{x}(\sigma_k)) \quad (3.43)$$

□

**Lemma 3.3.5.** Let  $\delta = \|\mathbf{c}(\mathbf{x}(\sigma))_-\|$ , then  $\mathbf{x}(\sigma)$  is also the solution to the problem

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \|\mathbf{c}(\mathbf{x})_-\| \leq \delta \end{aligned} \quad (3.44)$$

*Proof.* The problem is equivalent to

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \|\mathbf{c}(\mathbf{x})_-\| \leq \|\mathbf{c}(\mathbf{x}(\sigma))_-\| \end{aligned} \quad (3.45)$$

$$f(\mathbf{x}(\sigma)) + \sigma \|\mathbf{c}(\mathbf{x}(\sigma))_-\|^2 = \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})_-\|^2 \quad (3.46)$$

Then for all  $\mathbf{x} \in \mathbb{R}^n$ , we have

$$f(\mathbf{x}(\sigma)) + \sigma \|\mathbf{c}(\mathbf{x}(\sigma))_-\|^2 \leq f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})_-\|^2 \quad (3.47)$$

$$f(\mathbf{x}(\sigma)) - f(\mathbf{x}) \leq \sigma (\|\mathbf{c}(\mathbf{x})_-\|^2 - \|\mathbf{c}(\mathbf{x}(\sigma))_-\|^2) \quad (3.48)$$

That is, if  $\|\mathbf{c}(\mathbf{x})_-\| \leq \|\mathbf{c}(\mathbf{x}(\sigma))_-\|$ , then

$$f(\mathbf{x}(\sigma)) - f(\mathbf{x}) \leq \sigma (\|\mathbf{c}(\mathbf{x})_-\|^2 - \|\mathbf{c}(\mathbf{x}(\sigma))_-\|^2) \leq 0 \quad (3.49)$$

i.e., for all  $\mathbf{x} \in \mathbb{R}^n$ ,  $f(\mathbf{x}(\sigma)) \leq f(\mathbf{x})$ .

□

### 3.3.3 Extended Lagrange function method

#### 3.3.3.1 Revisit Penalty method

Consider equality constrained problem

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{c}(\mathbf{x}) = 0 \end{aligned} \quad (3.50)$$

The Lagrange function of (3.50) is

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda^T \mathbf{c}(\mathbf{x}) \quad (3.51)$$

From K-T condition, we have for global optimal point  $\mathbf{x}^*$ ,

$$\begin{cases} \nabla \mathbf{x} \mathcal{L}(\mathbf{x}, \lambda) = 0 \\ \nabla \lambda \mathcal{L}(\mathbf{x}, \lambda) = 0 \end{cases} \quad (3.52)$$

i.e.,  $\mathbf{x}^*$  is a stable point of  $\mathcal{L}(\mathbf{x}, \lambda)$ . Notice that

$$\nabla \mathbf{x} \mathcal{L}(\mathbf{x}, \lambda) = \nabla f(\mathbf{x}) - \sum_i \lambda_i \nabla c_i(\mathbf{x}) \quad (3.53)$$

For the corresponding penalty function

$$P_\sigma(\mathbf{x}) = f(\mathbf{x}) + \sigma \|\mathbf{c}(\mathbf{x})\|^2 \quad (3.54)$$

we have the K-T condition is

$$\nabla P_\sigma(\mathbf{x}) = \nabla f(\mathbf{x}) + 2\sigma \mathbf{c}^T(\mathbf{x}) \nabla \mathbf{c}(\mathbf{x}) \quad (3.55)$$

$$= \nabla f(\mathbf{x}) + \sum_i 2\sigma c_i(\mathbf{x}) \nabla c_i(\mathbf{x}) \quad (3.56)$$

If we want (3.56) to be a good approximation of (3.53), i.e.,

$$\lambda_i \approx -2\sigma c_i(\mathbf{x}) \quad (3.57)$$

Notice that  $c_i(\mathbf{x}) \approx 0$ , so we need  $|\sigma| \rightarrow \infty$ .

### 3.3.3.2 Argmented Lagrange function method

Consider *Argmented Lagrange function*

$$P(\mathbf{x}, \lambda, \sigma) = \mathcal{L}(\mathbf{x}, \lambda) + \frac{\sigma}{2} \|\mathbf{c}(\mathbf{x})\|^2 \quad (3.58)$$

The K-T condition of the function is

$$\nabla P(\mathbf{x}, \lambda, \sigma) = \nabla f(\mathbf{x}) - \sum_i (\lambda_i - \sigma c_i(\mathbf{x})) \nabla c_i(\mathbf{x}) \quad (3.59)$$

$$= \nabla f(\mathbf{x}) - \sum_i \lambda_i \nabla c_i(\mathbf{x}) \quad (3.60)$$

i.e., the K-T condition of  $P$  is similar to the original problem (3.50).

**Theorem 3.3.6.** Suppose  $\mathbf{x}^*$  and  $\lambda^*$  satisfy the K-T condition of (3.50), then there exists  $\bar{\sigma}$  such that when  $\sigma > \bar{\sigma}$ ,  $\mathbf{x}^*$  is the strict local minima of  $P(\mathbf{x}, \lambda^*, \sigma)$ .

However, the optimal value of  $\lambda$  remains unknown.

### 3.3.4 Barrier method



# 4

## Convex Optimization

### 4.1 Convex set

#### 4.1.1 Affine set

**Definition 4.1.1** (Affine set). A set  $\mathcal{C} \subset \mathbb{R}^n$  is affine if  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$  and  $\theta \in \mathbb{R}$ , we have

$$\theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2 \in \mathcal{C} \quad (4.1)$$

**Definition 4.1.2** (Affine hull). The set of all affine combinations of points in some set  $\mathcal{C} \subset \mathbb{R}^n$  is called the affine hull of  $\mathcal{C}$ , denoted  $\text{aff}\mathcal{C}$ :

$$\text{aff}\mathcal{C} = \left\{ \sum_{i=1}^k \theta_i \mathbf{x}_i \mid \mathbf{x}_1, \dots, \mathbf{x}_k \in \mathcal{C}, \theta_1 + \dots + \theta_k = 1 \right\} \quad (4.2)$$

**Remark 2.** The affine hull is the smallest affine set that contains  $\mathcal{C}$ .

*Proof.* For any affine set  $\mathcal{A}$  contains  $\mathcal{C}$ , we have

$$\sum_{i=1}^k \theta_i \mathbf{x}_i \in \mathcal{A}, \forall \mathbf{x}_1, \dots, \mathbf{x}_k \in \mathcal{C}, \theta_1 + \dots + \theta_k = 1 \quad (4.3)$$

i.e.,  $\text{aff}\mathcal{C} \subset \mathcal{A}$ . □

#### 4.1.2 Convex set

**Definition 4.1.3** (Convex set). A set  $\mathcal{C} \subset \mathbb{R}^n$  is convex if  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$  and  $0 \leq \theta \leq 1$ , we have

$$\theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2 \in \mathcal{C} \quad (4.4)$$

**Definition 4.1.4** (Convex hull). The set of all convex combinations of points in some set  $\mathcal{C} \subset \mathbb{R}^n$  is called the convex hull of  $\mathcal{C}$ , denoted  $\text{conv}\mathcal{C}$ :

$$\text{conv}\mathcal{C} = \left\{ \sum_{i=1}^k \theta_i \mathbf{x}_i \mid \mathbf{x}_1, \dots, \mathbf{x}_k \in \mathcal{C}, \theta_i \geq 0, \theta_1 + \dots + \theta_k = 1 \right\} \quad (4.5)$$

**Remark 3.** The convex hull is the smallest convex set that contains  $\mathcal{C}$ .

#### 4.1.3 Cone

**Definition 4.1.5** (Cone). A set  $\mathcal{C}$  is called a cone, if  $\forall \mathbf{x} \in \mathcal{C}$  and  $\theta \geq 0$  we have  $\theta \mathbf{x} \in \mathcal{C}$ . A set  $\mathcal{C}$  is called a convex cone if it is convex and a cone, i.e.,  $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$  and  $\theta_1, \theta_2 \geq 0$ , we

have

$$\theta_1 \mathbf{x}_1 + \theta_2 \mathbf{x}_2 \in \mathcal{C} \quad (4.6)$$

**Definition 4.1.6** (Conic hull). *The conic hull of set  $\mathcal{C}$  is the set of all conic combinations of points in  $\mathcal{C}$ , i.e.,*

$$\left\{ \sum_{i=1}^k \theta_i \mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{C}, \theta_i \geq 0, i = 1, \dots, k \right\} \quad (4.7)$$

#### 4.1.4 Proper cones and generalized inequalities

### 4.2 Convex function

**Definition 4.2.1** (Convex function). *A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex if  $\text{dom} f$  is a convex set and if  $\forall x, y \in \text{dom} f$  and  $\theta$  with  $0 \leq \theta \leq 1$ , we have*

$$f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2) \quad (4.8)$$

#### 4.2.1 First order condition

Suppose  $f$  is differentiable

**Theorem 4.2.1.** *Function  $f$  is convex if and only if  $\text{dom} f$  is a convex set and for  $\forall x, y \in \text{dom} f$ , the following holds:*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad (4.9)$$

**Remark 4.** *If  $\nabla f(x^*) = 0$ , then for  $\forall y \in \text{dom} f$ ,  $f(y) \geq f(x^*)$ , i.e.,  $x^*$  is the global minimizer of  $f$ .*

## 4.3 Convex optimization

### 4.3.1 Quadratically constrained quadratic program

$$\begin{aligned} \min \quad & \frac{1}{2} \mathbf{x}^T P_0 \mathbf{x} + \mathbf{q}_0^T \mathbf{x} + r_0 \\ \text{s.t.} \quad & \frac{1}{2} \mathbf{x}^T P_i \mathbf{x} + \mathbf{q}_i^T \mathbf{x} + r_i \leq 0, \quad i = 1, \dots, m \\ & A\mathbf{x} = \mathbf{b} \end{aligned} \quad (4.10)$$

### 4.3.2 Second-order cone program

$$\begin{aligned} \min \quad & \mathbf{f}^T \mathbf{x} \\ \text{s.t.} \quad & \|A_i \mathbf{x} + \mathbf{b}_i\| \leq \mathbf{c}_i^T \mathbf{x} + \mathbf{d}_i, \quad i = 1, \dots, m \\ & F\mathbf{x} = \mathbf{g} \end{aligned} \quad (4.11)$$

**Lemma 4.3.1.** *Any QCQP problem can be formulated as a SOCP problem.*

*Proof.* The QCQP problem is equivalent to

$$\min -r_0 \quad (4.12)$$

$$s.t. \frac{1}{2} \mathbf{x}^T P_i \mathbf{x} + \mathbf{q}_i^T \mathbf{x} + r_i \leq 0, \quad i = 0, \dots, m \quad (4.13)$$

$$A\mathbf{x} = \mathbf{b} \quad (4.14)$$

Then we need to prove that (121) can be formulated as (118).

$$\frac{1}{2} \mathbf{x}^T P_i \mathbf{x} + \mathbf{q}_i^T \mathbf{x} + r_i \leq 0 \quad (4.15)$$

$$\Leftrightarrow \mathbf{x}^T P_i \mathbf{x} + 2(\mathbf{q}_i^T \mathbf{x} + r_i) \leq 0 \quad (4.16)$$

$$\Leftrightarrow \mathbf{x}^T P_i \mathbf{x} + 2(\mathbf{q}_i^T \mathbf{x} + r_i) + (\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2})^2 \leq (\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2})^2 \quad (4.17)$$

$$\Leftrightarrow \mathbf{x}^T P_i \mathbf{x} + (\mathbf{q}_i^T \mathbf{x} + r_i + \frac{1}{2})^2 \leq (\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2})^2 \quad (4.18)$$

Since  $P_i$  is positive semi-definite,  $P_i = A_i^T A_i$ , then

$$\Leftrightarrow \mathbf{x}^T P_i \mathbf{x} + (\mathbf{q}_i^T \mathbf{x} + r_i + \frac{1}{2})^2 \leq (\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2})^2 \quad (4.19)$$

$$\Leftrightarrow \|A_i \mathbf{x}\|^2 + \|\mathbf{q}_i^T \mathbf{x} + r_i + \frac{1}{2}\|^2 \leq (\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2})^2 \quad (4.20)$$

Let

$$A'_i = \begin{pmatrix} A_i \\ \mathbf{q}_i^T \end{pmatrix} \quad (4.21)$$

$$\mathbf{b}_i = \begin{pmatrix} \mathbf{0}_{n \times 1} \\ r_i + \frac{1}{2} \end{pmatrix} \quad (4.22)$$

From (123) and  $\mathbf{x}^T P_i \mathbf{x} \geq 0$ , we can derive that  $\mathbf{q}_i^T \mathbf{x} + r_i \leq 0$ , i.e.,  $\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2} \leq 0$ .

Then (128) can be formulated as

$$\|A'_i \mathbf{x} + \mathbf{b}_i\|^2 \leq (\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2})^2 \quad (4.23)$$

$$\Leftrightarrow \|A'_i \mathbf{x} + \mathbf{b}_i\| \leq -(\mathbf{q}_i^T \mathbf{x} + r_i - \frac{1}{2}) \quad (4.24)$$

□





# 5

## Sparse Optimization

### 5.1 Compressed Sensing

#### 5.1.1 Problem formulation

$$(P_0) \quad \begin{array}{ll} \min_{\mathbf{x} \in \mathbb{R}^n} & \|\mathbf{x}\|_0 \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \end{array} \quad (5.1)$$

The definition above means to find the sparsest solution for underdetermined linear equation  $A\mathbf{x} = \mathbf{b}$  ( $A \in \mathbb{R}^{m \times n}$ ,  $m \ll n$ ).

**Definition 5.1.1** (spark). *The spark of a given matrix  $A$  is the smallest number of columns from  $A$  that are linearly dependent.*

**Theorem 5.1.1.** *If a system of linear equations  $A\mathbf{x} = \mathbf{b}$  has a solution  $\mathbf{x}$  obeying  $\|\mathbf{x}\|_0 < \frac{\text{spark}(A)}{2}$ , this solution is necessarily the sparsest possible.*

**Definition 5.1.2.** *The mutual coherence of a given matrix  $A$  is the largest absolute normalized inner product between different columns from  $A$ . Denoting the  $k$ -th column in  $A$  by  $\mathbf{a}_k$ , the mutual coherence is given by*

$$\mu(A) = \max_{1 \leq i \neq j \leq n} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2} \quad (5.2)$$

**Lemma 5.1.2.** *For any matrix  $A \in \mathbb{R}^{m \times n}$ , the following relationship holds:*

$$\text{spark}(A) \geq 1 + \frac{1}{\mu(A)} \quad (5.3)$$

Then we have the following theorem:

**Theorem 5.1.3.** *If a system of linear equations  $A\mathbf{x} = \mathbf{b}$  has a solution  $\mathbf{x}$  obeying  $\|\mathbf{x}\|_0 < (1 + \frac{1}{\mu(A)})/2$ , this solution is necessarily the sparsest possible.*

#### 5.1.2 Pursuit Algorithms

##### 5.1.2.1 Orthogonal Matching Pursuit

##### 5.1.2.2 Basis Pursuit