

# **COMPLEMENTI DI RICERCA OPERATIVA**

Prof. Marco Trubian  
6 CFU

**Luca Cappelletti**

Lecture Notes  
Year 2017/2018



Magistrale Informatica  
Università di Milano  
Italy  
25 aprile 2018

# Indice

<b>I</b>	<b>Programmazione non lineare</b>	<b>2</b>
<b>1</b>	<b>Alcune definizioni base</b>	<b>3</b>
1.1	La convessità . . . . .	3
1.2	Massimi e minimi locali . . . . .	3
<b>2</b>	<b>Ottimizzazione non vincolata</b>	<b>4</b>
2.1	Condizioni necessarie di ottimalità del primo ordine . . . . .	4
2.2	Condizioni necessarie di ottimalità del secondo ordine . . . . .	4
2.3	Condizioni necessarie di ottimalità in senso stretto del secondo ordine . . . . .	4
<b>3</b>	<b>Programmazione quadratica</b>	<b>5</b>
<b>4</b>	<b>Convergenza</b>	<b>6</b>
4.1	Algoritmo convergente localmente e globalmente . . . . .	6
4.1.1	Convergente localmente . . . . .	6
4.1.2	Convergente globalmente . . . . .	6
4.2	Velocità di convergenza . . . . .	6
4.2.1	Q-lineare . . . . .	6
4.2.2	Q-superlineare . . . . .	6
4.2.3	Q-quadratica . . . . .	6
<b>5</b>	<b>Metodi di ottimizzazione continua</b>	<b>7</b>
5.1	Condizioni di Wolfe . . . . .	7
5.2	Metodo di Armijo per stabilire la stepsize . . . . .	7
5.3	Convergenza dei metodi di ricerca lineare approssimata . . . . .	7
<b>6</b>	<b>Metodi di ottimizzazione</b>	<b>9</b>
6.1	Metodi a discesa rapida . . . . .	9
6.2	Metodi Newton . . . . .	9
6.2.1	Metodi Newton modificati . . . . .	10
6.3	Metodi Quasi-Newton . . . . .	10
6.4	Metodi del gradiente coniugato . . . . .	10
<b>II</b>	<b>Programmazione lineare intera</b>	<b>12</b>
<b>7</b>	<b>Programmazione lineare intera</b>	<b>13</b>

## **Parte I**

# **Programmazione non lineare**

## Alcune definizioni base

### 1.1 La convessità

**Definizione 1.1.1 (Insieme convesso).** Un insieme  $X \subset \mathbb{R}^n$  è convesso se comunque presi due punti  $\underline{x}, \underline{y} \in X$ , allora  $\lambda \underline{x} + (1 - \lambda) \underline{y} \in X$ , per ogni  $\lambda \in [0, 1]$ .

La proprietà di convessità è invariante rispetto alle operazioni di moltiplicazione con uno scalare, unione e intersezione con un altro insieme convesso.

**Definizione 1.1.2 (Funzione convessa).** Una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  è convessa se il suo dominio è un insieme convesso  $X \subseteq \mathbb{R}^n$  e comunque presi due punti  $\underline{x}, \underline{y} \in X$  vale la relazione:

$$f(\lambda \underline{x} + (1 - \lambda) \underline{y}) \leq \lambda f(\underline{x}) + (1 - \lambda) f(\underline{y}) \quad \forall \lambda \in [0, 1]$$

La proprietà di convessità è invariante rispetto a moltiplicazione con uno scalare e somma tra funzioni convesse.

Vale inoltre che la funzione max di una o più funzioni convesse e che il luogo dei punti  $\underline{x}$  per i quali vale che  $f(\underline{x}) \leq \alpha$  è convesso.

**Definizione 1.1.3 (Problema convesso).** Un problema di ottimizzazione con funzione obiettivo e regione ammissibile entrambe convesse viene detto problema convesso.

### 1.2 Massimi e minimi locali

**Definizione 1.2.1 (Minimo globale).** Un punto  $\underline{x}^* \in X$  è un punto di minimo globale di  $f(\underline{x})$  se:

$$f(\underline{x}^*) \leq f(\underline{x}) \quad \forall \underline{x} \in X$$

**Definizione 1.2.2 (Minimo locale).** Un punto  $\underline{x}^* \in X$  è un punto di minimo locale di  $f(\underline{x})$  se esiste un intorno aperto  $I(\underline{x}^*, \epsilon)$  di  $\underline{x}^*$  avente raggio  $\epsilon > 0$  tale che:

$$f(\underline{x}^*) \leq f(\underline{x}) \quad \forall \underline{x} \in X \cap I(\underline{x}^*, \epsilon)$$

# 2

## Ottimizzazione non vincolata

**Definizione 2.0.1 (Direzione di discesa).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , un vettore  $\underline{d} \in \mathbb{R}^n$  si dice direzione di discesa per  $f$  in  $\underline{x}$  se:

$$\exists \lambda > 0 : f(\underline{x} + \lambda \underline{d}) < f(\underline{x})$$

**Definizione 2.0.2 (Derivata direzionale).** Sia data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , un vettore  $\underline{d} \in \mathbb{R}^n$  e un punto dove  $f$  è definita. Se esiste il limite:

$$\lim_{\lambda \rightarrow 0^+} \frac{f(\underline{x} + \lambda \underline{d}) - f(\underline{x})}{\lambda}$$

allora tale limite prende il nome di derivata direzionale della funzione  $f$  nel punto  $\underline{x}$  lungo la direzione  $\underline{d}$

### 2.1 Condizioni necessarie di ottimalità del primo ordine

**Teorema 2.1.1 (Condizioni necessarie di ottimalità del primo ordine).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , derivabile in  $\underline{x}^* \in \mathbb{R}^n$ , condizione necessaria affinché il punto  $\underline{x}^*$  sia un minimo locale per  $f$  è che il gradiente della funzione calcolato in  $\underline{x}^*$  sia nullo.

### 2.2 Condizioni necessarie di ottimalità del secondo ordine

**Teorema 2.2.1 (Condizioni necessarie di ottimalità del secondo ordine).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  di classe  $\underline{C}^2(\underline{x}^*)$ , condizione necessarie affinché il punto  $\underline{x}^*$  sia un minimo locale per  $f$  è che il gradiente della funzione calcolato in  $\underline{x}^*$  sia nullo e che valga la relazione seguente:

$$\underline{d}^T H(\underline{x}^*) \underline{d} \geq 0 \forall \underline{d} \in \mathbb{R}^n$$

Cioè l'hessiana è definita come **semipositiva**.

### 2.3 Condizioni necessarie di ottimalità in senso stretto del secondo ordine

**Teorema 2.3.1 (Condizioni necessarie di ottimalità del secondo ordine).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  di classe  $\underline{C}^2(\underline{x}^*)$ , condizione necessarie affinché il punto  $\underline{x}^*$  sia un minimo locale in **senso stretto** per  $f$  è che il gradiente della funzione calcolato in  $\underline{x}^*$  sia nullo e che valga la relazione seguente:

$$\underline{d}^T H(\underline{x}^*) \underline{d} > 0 \forall \underline{d} \in \mathbb{R}^n$$

Cioè l'hessiana è definita come **positiva**.

# 3

## Programmazione quadratica

Nella programmazione quadratica si approssima  $f$  con il seguente modello quadratico:

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

Esistono pochi casi possibili:

**$Q$  non è semidefinita positiva**  $f$  non ha un punto di minimo.

**$Q$  è definita positiva**  $\underline{x}^* = Q^{-1} \underline{b}$  è l'unico minimo globale.

**$Q$  è definita semi-positiva**  **$Q$  non è singolare**  $\underline{x}^* = Q^{-1} \underline{b}$  è l'unico minimo globale.

**$Q$  è singolare** non esiste una soluzione o esistono infinite soluzioni.

# 4

## Convergenza

Ovviamente un algoritmo è buono se converge rapidamente.

### 4.1 Algoritmo convergente localmente e globalmente

#### 4.1.1 Convergente localmente

Un algoritmo è convergente localmente quando converge solo se il punto da cui parte è in un intorno del punto ottimo.

#### 4.1.2 Convergente globalmente

Un algoritmo è convergente globalmente quando converge partendo da qualsiasi punto del dominio.

### 4.2 Velocità di convergenza

#### 4.2.1 Q-lineare

Quando il rapporto tra lo step  $k$  e lo step  $k + 1$  mantiene un valore costante.

#### 4.2.2 Q-superlineare

Quando il rapporto tra lo step  $k$  e lo step  $k + 1$  all'infinito è nullo.

#### 4.2.3 Q-quadratica

Quando il rapporto tra lo step  $k$  quadro e lo step  $k + 1$  mantiene un valore costante.

# 5

## Metodi di ottimizzazione continua

Si tratta del problema di determinare a partire dallo step  $k$  lo step  $k + 1$  in modo tale da avvicinarsi all'ottimo della funzione.

Esistono due strategie:

**Line search** Si determina una direzione e quindi si stabilisce una distanza con cui muoversi in questa direzione.

**Trust region** Si determina un raggio (una distanza) di confidenza e quindi si stabilisce una direzione nel limite di questa circonferenza verso cui muoversi.

Esiste chiaramente un tradeoff tra velocità e precisione nello stabilire la distanza con cui muoversi.

### 5.1 Condizioni di Wolfe

Per essere efficace, la line-search approssimata richiede che siano rispettate le condizioni di Wolfe. Esse sono due, una di **decremento sufficiente** ed una di **curvatura**, dove i coefficienti  $\underline{c}$  sono tali che  $0 < c_1 < c_2 < 1$

$$\begin{aligned} f(\underline{x} + \alpha \underline{d}) &\leq f(\underline{x}) + c_1 \alpha \nabla f(\underline{x})^T \underline{d} \\ \phi(\alpha) &\leq \phi(0) + \alpha c_1 \phi'(0) \end{aligned}$$

(a) Condizione di decremento sufficiente

$$\begin{aligned} f(\underline{x} + \alpha \underline{d})^T \underline{d} &\geq c_2 \alpha \nabla f(\underline{x})^T \underline{d} \\ \phi'(\alpha) &\geq c_2 \phi'(0) \end{aligned}$$

(b) Condizione di curvatura

Figura 5.1: Condizioni di Wolfe

Le condizioni di Wolfe forti introducono un vincolo di segno sulla curvatura (valore assoluto).

### 5.2 Metodo di Armijo per stabilire la stepsize

Si itera riducendo gradualmente la distanza di un fattore  $\sigma$ , usualmente pari circa a 0.9 sino a che il valore dello step successivo è più vicino all'ottimo dello step corrente. Nei metodi Newton e quasi Newton il coefficiente della distanza è inizializzato usualmente a 1.

### 5.3 Convergenza dei metodi di ricerca lineare approssimata

Se definiamo  $\theta_k$  come l'angolo tra  $\underline{d}_k$  e  $-\nabla f_k$ , allora possiamo determinare il coseno dell'angolo come:

$$\cos \theta_k = \frac{-\nabla f(\underline{x})_k^T \underline{d}_k}{\|\nabla f(\underline{x})_k\| \cdot \|\underline{d}_k\|}$$



**Teorema 5.3.1.** Sia  $\underline{d}_k$  una direzione di discesa e sia  $\alpha_k$  una distanza che rispetti le condizioni di Wolfe. Sia inoltre  $f$  una funzione limitata inferiormente su  $\mathbb{R}^n$ , **differenziabile continuamente** sull'insieme  $M$  che contiene  $L_f = \{\underline{x} : f(\underline{x}) \leq f(\underline{x}_0)\}$ , dove  $\underline{x}_0$  è il punto di inizio. Assumiamo inoltre che  $\nabla f$  sia **lipschitziana** sull'insieme  $M$ . Allora:

$$\sum_{j=0}^{\infty} \cos^2 \theta_j \|\nabla f(\underline{x}_j)\|^2 \leq \infty$$

*Dimostrazione.* Essendo valida la **condizione di curvatura**, allora vale la disequazione:

$$\nabla f_{k+1}^T \underline{d}_k \geq c_2 \nabla f_k^T \underline{d}_k$$

Sottraggo ad ambo i lati  $\nabla f_k \underline{d}_k$  ed ottengo:

$$(\nabla f_{k+1}^T - \nabla f_k^T) \underline{d}_k \geq (c_2 - 1) \nabla f_k^T \underline{d}_k$$

Siccome il gradiente della funzione è **lipschitziano** vale la disequazione:

$$(\nabla f_{k+1}^T - \nabla f_k^T) \underline{d}_k \leq \|\nabla f_{k+1} - \nabla f_k\| \|\underline{d}_k\| \leq L \|\underline{x}_{k+1} - \underline{x}_k\| \|\underline{d}_k\| = \alpha_k L \|\underline{d}_k\|^2$$

Da questa disequazione ricaviamo il valore di  $\alpha_k$ :

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{\nabla f_k^T \underline{d}_k}{\|\underline{d}_k\|^2}$$

Essendo quindi valida la **condizione di decremento sufficiente** vale la disequazione:

$$f_{k+1} \leq f_k + c_1 \alpha_k \underline{d}_k^T \nabla f_k = f_k - c_1 \frac{1 - c_2}{L} \frac{(\nabla f_k^T \underline{d}_k)^2}{\|\underline{d}_k\|^2}$$

Pongo  $c = c_1 \frac{1 - c_2}{L}$  ed ottengo:

$$f_{k+1} \leq f_k + c_1 \alpha_k \underline{d}_k^T \nabla f_k = f_k - c \frac{(\nabla f_k^T \underline{d}_k)^2}{\|\underline{d}_k\|^2}$$

Sostituisco  $\cos \theta_k = \frac{-\nabla f(\underline{x})_k^T \underline{d}_k}{\|\nabla f(\underline{x})_k\| \|\underline{d}_k\|}$  ed ottengo:

$$f_{k+1} \leq f_k + c_1 \alpha_k \underline{d}_k^T \nabla f_k = f_k - c \cos^2 \theta_k \|\nabla f_k\|^2$$

Per ricorsione ottengo la sommatoria:

$$f_{k+1} \leq f_k + c_1 \alpha_k \underline{d}_k^T \nabla f_k = f_0 - c \sum_{j=0}^k \cos^2 \theta_j \|\nabla f_j\|^2$$

Siccome la funzione  $f$  è limitata inferiormente si ottiene la **condizione di Zoutendijk**:

$$c \sum_{j=0}^k \cos^2 \theta_j \|\nabla f_j\|^2 \leq f_0 - f_{k+1} < \infty$$

Questo implica che:

$$\cos^2 \theta_j \|\nabla f_j\|^2 \rightarrow 0$$

Quindi se l'algoritmo soddisfa anche la **condizione angolare** (cioè sceglie una direzione di discesa che la rispetta)  $\cos \theta_k \geq \epsilon > 0$  allora **converge**:

$$\lim_{k \rightarrow \infty} \|\nabla f(\underline{x}_k)\| = 0$$

□

## Metodi di ottimizzazione

### 6.1 Metodi a discesa rapida

Son metodi che usano come funzione di aggiornamento  $\underline{x}_{k+1} = \underline{x}_k - \alpha_k \nabla f_k$ , in cui le direzioni sono ortogonali al contorno della funzione. Non dovendo calcolare l'hessiana lo sforzo computazionale non è eccessivo e converge globalmente, ma estremamente piano quando una funzione è patologica.

**Teorema 6.1.1 (Velocità di convergenza locale dei metodi a discesa rapida).** Data una matrice  $Q$  definita positiva, la seguente relazione vale  $\forall \underline{x} \in \mathbb{R}^n$ :

$$\frac{(\underline{x}^T \underline{x})^2}{(\underline{x}^T Q \underline{x})(\underline{x}^T Q^{-1} \underline{x})} \geq \frac{4\lambda_{\min}\lambda_{\max}}{(\lambda_{\min} + \lambda_{\max})^2}$$

Dove  $\lambda_{\min}$  e  $\lambda_{\max}$  son gli autovalori minimo e massimo di  $Q$ .

Ne segue che la velocità di convergenza dei metodi a discesa rapida è **lineare** per i modelli quadratici.

$$\frac{\|\underline{x}_{k+1} - \underline{x}^*\|_Q}{\|\underline{x}_k - \underline{x}^*\|_Q} \leq \frac{(\lambda_{\max} - \lambda_{\min})}{(\lambda_{\max} + \lambda_{\min})}$$

### 6.2 Metodi Newton

Prendendo in considerazione l'approssimazione di Taylor fermata al secondo ordine, quindi con l'hessiana, otteniamo come direzione di discesa quella che minimizza  $\nabla f^T \underline{d} + \frac{1}{2} \underline{d}^T H \underline{d}$ , cioè  $\underline{d} = -(H)^{-1} \nabla f$ . Quando  $H$  è **definita positiva** vale che:

$$\nabla f^T \underline{d} = -\underline{d}^T H \underline{d} \leq -\sigma \|\underline{d}\|^2$$

Cioè quando  $H$  è **definita positiva** la direzione di Newton è la **direzione di discesa**.

Nei **modelli quadratici** con  $Q$  **definita positiva** il metodo di Newton converge in un'iterazione, altrimenti non converge. Su funzioni generiche, la qualità della direzione dipende da quanto è definita positiva la matrice hessiana.

**Teorema 6.2.1 (Convergenza dei metodi di Newton).** Sia  $f \in C^2$  e sia  $H(x)$  continuamente lipschitziana in un intorno del punto ottimo  $\underline{x}^*$ . Si assuma che valga  $\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$ . Allora:

1. Se  $\underline{x}_0$  è sufficientemente vicino a  $\underline{x}^*$ , allora  $\{\underline{x}_k\} \rightarrow \underline{x}^*$
2.  $\{\underline{x}_k\}$  converge **quadraticamente**

3.  $\{\|\nabla f(\underline{x}_k)\|\}$  converge quadraticamente a zero.

I metodi Newton sono **convergenti localmente**.

La complessità computazionale è di  $O(n^3)$

### 6.2.1 Metodi Newton modificati

Sono metodi che modificano l'Hessiano, o rendendo la matrice positiva o scegliendo una direzione di discesa tramite metodi di discesa rapida quando necessario.

## 6.3 Metodi Quasi-Newton

Sono metodi in cui viene utilizzata un'approssimazione dell'Hessiano, che è computazionalmente costoso da calcolare. Viene utilizzata al suo posto una matrice chiamata  $G_k$  al posto di  $H_k^{-1}$ , e quindi calcolano la direzione di discesa come  $\underline{d}_k = -G_k \nabla f(\underline{x}_k)$ .

**Definizione 6.3.1 (Relazione secante).** Definendo  $\underline{p}_k = \nabla f(\underline{x}_{k+1}) - \nabla f(\underline{x}_k)$  possiamo definire la relazione secante:

$$H(\underline{x}_k)\underline{h}_k \approx \underline{p}_k \quad \text{or} \quad H(\underline{x}_k)^{-1}\underline{p}_k \approx \underline{h}_k$$

Inizializzando  $G_0 = I$ , possiamo imporre che ad ogni iterazione la matrice  $G_{k+1}$  debba soddisfare la relazione secante con la seguente uguaglianza:

$$G_{k+1}\underline{p}_k = \underline{h}_k$$

La realizzazione specifica di come si ottiene  $G_{k+1}$  partendo da  $G_k$  varia dai differenti metodi Quasi-Newton. Questi metodi impongono che  $G_k = G_k^T$  e che  $G_{k+1} - G_k$  abbia un rango basso.

Ne esistono di due categorie:

1. Matrice a rango unitario simmetrico o SR1.
2. A rango due:
  - (a) DFP
  - (b) BFGS

I **metodi a rango due** hanno alcune proprietà interessanti:

1. La matrice  $G_k$  converge a  $H(\underline{x}_k)^{-1}$  sui modelli quadratici.
2. Se  $G_0$  è definita positiva allora tutte le  $G_k$  sono definite positive.
3. Il costo computazionale è di  $O(n^2)$  in ogni iterazione.
4. La velocità di convergenza è **superlineare**.
5. In particolare il metodo BFGS garantisce convergenza globale se lo step-size rispetta le condizioni di Wolfe.

## 6.4 Metodi del gradiente coniugato

Si tratta di una delle tecniche più utili per risolvere sistemi lineari di grandi dimensioni e può essere adattata per risolvere problemi di ottimizzazione non lineare. Si tratta di un metodo iterativo che risolve sistemi della forma  $A\underline{x} = \underline{b}$  dove  $A$  è una matrice definita positiva che è  $A = A^T$ .

**Definizione 6.4.1 (Vettori coniugati).** Un insieme di vettori  $\underline{p}_0, \underline{p}_1, \dots, \underline{p}_h$  è detto **coniugato** rispetto ad una matrice  $A$  se vale la relazione:

$$\underline{p}_i^T A \underline{p}_j = 0 \quad \forall i \neq j$$

Dato un punto di inizio  $\underline{x}_0$  e le **direzioni coniugate**  $\underline{p}_0, \dots, \underline{p}_{n-1}$  generiamo la sequenza  $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$  dove  $\alpha_k$  è un minimizzatore monodimensionale ed è dato da:

$$\alpha_k = -\frac{r_k^T \underline{p}_k}{\underline{p}_k^T A \underline{p}_k}$$

**Teorema 6.4.2.** Convergenza di sequenza con vettori coniugati Dato un qualsiasi  $\underline{x}_0$ , la sequenza generata da  $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$  converge a  $\underline{x}^*$  in al più  $n$  iterazioni.

Il metodo del gradiente coniugato è un metodo a direzione coniugata con una proprietà importante:  $\underline{p}_k$  può essere ottenuta sapendo solo  $\underline{p}_{k-1}$  e  $\underline{p}_k$  risulta coniugato a tutte le direzioni precedenti.

Il vettore  $\underline{p}_k$  è definito come una combinazione lineare del gradiente e del vettore precedente:

$$\underline{p}_k = -\nabla f_k + \beta_k \underline{p}_{k-1}$$

Dove  $\beta$  è definito sfruttando la definizione di vettore coniugato:

$$\beta_k = \frac{\nabla f_k^T A \underline{p}_{k-1}}{\underline{p}_{k+1}^T A \underline{p}_{k-1}}$$

Inizialmente il valore del vettore è  $\underline{p}_0 = -\nabla f_0$ .

Esistono vari metodi del gradiente coniugato che variano in base a come viene calcolato il valore di  $\beta_{k+1}$  a partire dall'iterazione precedente.

In particolare è utile perchè utilizza poco spazio, non è computazionalmente pesante e converge rapidamente. Viene però preferita l'eliminazione gaussiana quando il problema è di piccola dimensione, essendo meno sensibile agli errori di arrotondamento.

## **Parte II**

# **Programmazione lineare intera**

# 7

## Programmazione lineare intera