



UNIVERSITÀ DI PISA  
MASTER DEGREE IN COMPUTER SCIENCE  
AI AND BDT CURRICULA

Computational Mathematics  
for Learning and Data Analysis

ANNO ACCADEMICO 2022/2023

# Indice

<b>I</b>	<b>Ottimizzazione matematica</b>	<b>4</b>
<b>1</b>	<b>Ottimizzazione in una dimensione</b>	<b>5</b>
1.1	Introduzione ai problemi di ottimizzazione . . . . .	5
1.2	Ottimizzazione in una dimensione . . . . .	6
1.2.1	Ok I'm gonna optimize these functions... Damn this hard . . . . .	6
1.2.2	Minimi locali . . . . .	7
1.2.3	Modello del primo ordine . . . . .	9
1.2.4	Modello del secondo ordine e metodo di Newton . . . .	11
1.2.5	Misurare la velocità di convergenza . . . . .	14
<b>2</b>	<b>Ottimizzazione non vincolata</b>	<b>16</b>
2.1	Richiami di analisi in più variabili . . . . .	16
2.1.1	Prodotti scalari, norme, distanze . . . . .	16
2.1.2	Gradiente, Hessiana, Jacobiana . . . . .	18
2.1.3	Condizioni di ottimalità al primo e secondo ordine . . .	20
2.1.4	Funzioni convesse . . . . .	22
2.2	Algoritmo della discesa del gradiente . . . . .	24
2.2.1	Metodo del gradiente con passo costante . . . . .	25
2.2.2	Metodo del gradiente con line-search "esatta" . . . . .	28
2.2.3	Condizioni di Armijo e Wolfe . . . . .	29
2.3	Metodo di Newton in più variabili . . . . .	35
2.3.1	Newton per funzioni convesse . . . . .	36
2.3.2	Newton per funzioni non convesse . . . . .	39
2.4	Metodi quasi-Newton . . . . .	41
2.4.1	DFP . . . . .	42
2.4.2	BFGS . . . . .	43
2.4.3	Limited-memory BFGS . . . . .	44

2.5	Gradiente coniugato e Heavy Ball . . . . .	45
2.5.1	Metodo del gradiente coniugato . . . . .	45
2.5.2	Heavy Ball . . . . .	47
2.6	Ottimizzazione di funzioni non differenziabili . . . . .	47
2.6.1	Subgradiente e subdifferenziale . . . . .	48
2.6.2	Metodi subgradiente . . . . .	48
2.6.3	Smoothed Gradient . . . . .	51
2.6.4	Bundle Methods . . . . .	51
<b>3</b>	<b>Ottimizzazione vincolata</b>	<b>52</b>
3.1	Introduzione all'ottimizzazione vincolata . . . . .	52
3.1.1	Cenni di topologia . . . . .	53
3.2	Condizioni di ottimalità . . . . .	54
3.2.1	Cono Tangente . . . . .	54
3.2.2	Insiemi convessi . . . . .	56
3.2.3	Lemma di Farkas . . . . .	58
3.2.4	Condizioni di Karush–Kuhn–Tucker . . . . .	61
3.3	Dualità Lagrangiana . . . . .	63
3.3.1	Funzione Lagrangiana e dualità . . . . .	63
3.3.2	Duali speciali . . . . .	65
3.4	Algoritmi per l'ottimizzazione vincolata . . . . .	66
3.4.1	Metodo Active Set per problemi quadratici . . . . .	67
3.4.2	Gradiente proiettato . . . . .	69
3.4.3	Metodo di Frank-Wolfe . . . . .	71
3.4.4	Metodo duale . . . . .	72
3.4.5	Barrier Methods . . . . .	73

# Introduzione

Ho deciso di riassumere in questa dispensa i principali contenuti delle lezioni del professor Antonio Frangioni e del professor Federico Poloni del corso di Computational Mathematics dell' a.a. 2022/2023 del corso di laurea in Artificial Intelligence.

Queste dispense sono intese prima di tutto come uno strumento per aiutare me stesso a studiare, perciò sono possibili e anzi probabili eventuali errori, imprecisioni o typo. Nel caso decida un giorno di pubblicare tali dispense, sono gradite correzioni di tali errori, potete scrivermi alla mail [g.lagomarsini@studenti.unipi.it](mailto:g.lagomarsini@studenti.unipi.it) .

Per una buona fruibilità del documento è consigliata un po' di familiarità dei concetti di un corso standard di Analisi I, quali continuità e derivabilità, e qualche nozione di algebra lineare (faccio comunque notare che nelle loro lezioni i professori ridefiniscono praticamente ogni concetto). Saranno presenti qualche nozione di continuità, derivabilità ecc... in più variabili visto che nella triennale di informatica almeno fino a qualche anno fa non si toccavano tali argomenti.

Siccome il corso è diviso in due sezioni interconnesse ma distinte, ho deciso anch'io di dividere le dispense nelle due parti principali, ovvero quella di Ottimizzazione e quella di Analisi Numerica. Le varie figure presenti nel documento sono state prese per la maggior parte dei casi direttamente dalle slide dei professori. In alcuni casi ho preso alcune figure da internet evitando di citare le fonti tanto non è che lo vendo sto robo.

Per il momento mi concentrerò sulla parte di ottimizzazione, ma spero di riuscire a scrivere pure la parte di analisi numerica. Il motivo di questa scelta è che le matrici sono pallose da scrivere in  $\text{\LaTeX}$ .

Giacomo Lagomarsini

# Parte I

## Ottimizzazione matematica

# Capitolo 1

## Ottimizzazione in una dimensione

### 1.1 Introduzione ai problemi di ottimizzazione

La prima parte delle dispense sarà dedicata ad una trattazione di alcune tecniche di ottimizzazione, ovvero della ricerca di minimi o massimi di funzioni.

**Definizione.** Dato un insieme  $X$ , detto *regione ammissibile* (del problema) e una funzione  $f : X \rightarrow \mathbb{R}$ , un problema di ottimizzazione è trovare

$$(P) \quad f_* = \min_{x \in X} f(x). \quad (1.1)$$

Osserviamo che se  $f_*$  esiste, è unico, ma in generale non è unico  $x_*$  tale per cui  $f(x_*) = f_*$ . In generale possiamo formulare in maniera equivalente il problema come la ricerca del valore  $x_*$  che minimizza  $f$ , ovvero di

$$(P) \quad x_* = \arg \min_{x \in X} f(x). \quad (1.2)$$

Osserviamo anche che non è restrittivo supporre di voler cercare il minimo anziché il massimo: infatti

$$\min_{x \in X} f(x) = - \max_{x \in X} -f(x),$$

perciò se dobbiamo massimizzare un'altra funzione  $g(x)$ , possiamo minimizzare  $-g(x)$ . Se  $f$  è definita su un insieme più grande  $F$  con  $X \subset F$ , chiamiamo *inammissibili* i punti di  $F \setminus X$ .

Studieremo funzioni reali in una o più variabili, ovvero funzioni  $f : X \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$ , con  $d \geq 1$ . Iniziamo a trattare il caso unidimensionale, ovvero il caso in cui  $d = 1$ . Tale caso è interessante perché come vedremo spesso potremo ridurre singoli passaggi in algoritmi per la risoluzione di casi più complicati, ovvero casi di dimensione maggiore, a problemi di ottimizzazione unidimensionale (detta anche *line search*).

## 1.2 Ottimizzazione in una dimensione

### 1.2.1 Ok I'm gonna optimize these functions... Damn this hard

Anche nel caso più semplice possibile, quello di una funzione  $f : \mathbb{R} \rightarrow \mathbb{R}$ , non è sempre facile trovare un minimo locale. Pensiamo ad una funzione costante in  $\mathbb{R} \setminus x_*$ , con valore  $f(x) = 1$  e con  $f(x_*) = 0$ : in tale caso se non conosciamo nulla su dove possa trovarsi  $x_*$ , è praticamente impossibile trovare tale minimo. La funzione appena descritta ha una discontinuità in  $x_*$ , perciò si potrebbe supporre che per funzioni continue la situazione migliori: immaginiamo una funzione che “scende” con velocità arbitraria, come nella figura 1.1.

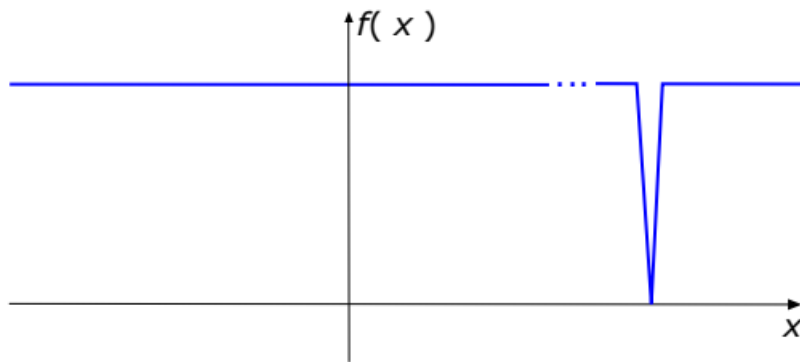


Figura 1.1: Funzione continua molto difficile da ottimizzare :(

Facciamo perciò delle assunzioni sul nostro problema:

- Scegliamo una regione ammissibile chiusa e limitata  $X = [x_1, x_2]$ : questa ipotesi ci garantisce che esista un minimo (teorema di Weierstrass).
- Cerchiamo una soluzione  $\epsilon$ -approssimata per qualche  $\epsilon > 0$ , cioè una soluzione  $f(x_\epsilon) \leq f_* + \epsilon$ . A volte si sceglie  $\epsilon \approx 10^{-16}$ , cioè dell'ordine di grandezza della precisione di macchina di un computer, ma spesso  $\epsilon$  può essere anche molto più grande.
- Assumiamo che la funzione da ottimizzare sia *L-lipschitziana*:

**Definizione.** Diciamo che una funzione  $f : X \rightarrow \mathbb{R}$  è lipschitziana con costante di Lipschitz  $L > 0$ , o in breve *L-lipschitz*, se per ogni  $x, y \in X$  si ha che

$$|f(x) - f(y)| \leq L|x - y|.$$

Questa definizione formalizza il fatto che se due punti sono molto vicini anche le loro immagini saranno ragionevolmente vicine.

Con queste ipotesi è facile trovare un algoritmo per trovare una soluzione  $\epsilon$ -approssimata: basta campionare uniformemente l'intervallo prendendo punti a distanza sempre uguale e minore o uguale a  $\epsilon/L$ .

Osserviamo che in questo modo ci sono circa  $DL/\epsilon$  punti, dove  $D$  è la lunghezza dell'intervallo  $X$  iniziale. Perciò dobbiamo calcolare  $f$  in  $O\left(\frac{DL}{\epsilon}\right)$  punti. In pratica perciò l'algoritmo non è per niente efficiente: è infatti esponenziale in  $\log \frac{1}{\epsilon}$ , cioè nel numero di cifre significative dell'approssimazione. Si può dimostrare però che, con solamente queste ipotesi, nessun altro algoritmo può fare meglio di così su ogni istanza.

### 1.2.2 Minimi locali

Fin'ora abbiamo cercato di trovare il minimo *globale* di una funzione. Abbassiamo un po' le nostre pretese e cerchiamo un punto di minimo *locale* di  $f$ :

**Definizione.** Un minimo locale di  $f$  è  $x_*$  tale che esiste  $\delta > 0$  tale per cui per ogni altro  $x$  in  $(x_* - \delta, x_* + \delta)$ ,  $f(x_*) \leq f(x)$ .

Se  $f$  è un minimo locale, “tipicamente” (ma non sempre!) esiste un  $\delta'$  (in generale non lo stesso nella definizione di minimo locale) tale per cui  $f$  è *unimodale* in  $(x_* - \delta', x_* + \delta')$ :



**Definizione.**  $f$  è unimodale attorno a  $x_*$  se esistono due punti  $x_- \leq x_* \leq x_+$  tale per cui  $f$  è decrescente in  $[x_-, x_*]$  e crescente in  $[x_*, x_+]$ .

L'intervallo  $[x_-, x_+]$  della definizione di funzione unimodale è detto *bacino di attrazione*.

In un bacino di attrazione abbiamo un algoritmo per trovare un minimo approssimato molto semplice: dato  $X = [x_-, x_+]$ ,

1. Scegliamo due punti  $x'_-$  e  $x'_+$ , con  $x_- < x'_- < x'_+ < x_+$ .
2. Se  $f(x'_-) < f(x'_+)$  scartiamo  $[x'_+, x_+]$ , ovvero reiteriamo su  $[x_-, x'_+]$ .
3. Altrimenti reiteriamo su  $[x'_-, x_+]$ .

La domanda che ci poniamo subito è come scegliere  $x'_-$  e  $x'_+$ ?

Se  $D$  è la lunghezza dell'intervallo, possiamo scegliere  $r \in (\frac{1}{2}, 1)$  e prendere  $x'_- = x_- + (1 - r)D$ ,  $x'_+ = x_+ - (1 - r)D$ . In tal modo ad ogni passo la lunghezza dell'intervallo passa da  $D$  a  $rD$ . Perciò l'intuizione ci dice di prendere  $r$  molto vicino a  $\frac{1}{2}$  per andare il più veloce possibile: in questo modo, ad ogni iterazione, la lunghezza dell'intervallo viene dimezzata.

### Golden ratio search

Se vogliamo minimizzare il numero di valutazioni della funzione  $f$ , possiamo altrimenti prendere come uno dei due  $x'_-$  e  $x'_+$  il rimanente tra i due punti. In questo caso deve valere che, nel nuovo intervallo di lunghezza  $rD$ , il punto rimanente tra i due diventi uno dei due nuovi punti mediani scelti (se abbiamo scartato  $[x_-, x'_-]$ , il vecchio  $x_+$  diventerà il nuovo  $x'_-$  e viceversa) (Figura 1.2). Vogliamo cioè che valga la proporzione

$$1 : r = r : 1 - r,$$

perciò si trova che  $r$  deve risolvere l'equazione

$$r^2 + r - 1 = 0$$

Otteniamo perciò

$$r = \frac{\sqrt{5} - 1}{2} = \frac{1}{g} = 1 - g$$

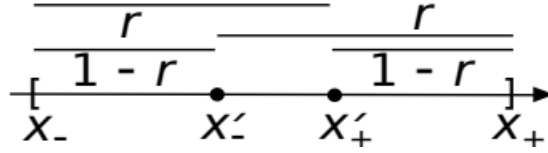


Figura 1.2: Golden Ratio Search

dove  $g = \frac{\sqrt{5}+1}{2}$  è la sezione aurea. Tale metodo di ricerca si chiama *golden ratio search*.

Quando l'intervallo è più piccolo di  $\delta = \frac{\epsilon}{L}$ , dove  $L$  è la costante di Lipschitz della funzione, siamo sicuri che un qualsiasi punto nell'intervallo ci fornisca una soluzione  $\epsilon$ -approssimata. Questo algoritmo fa un numero  $O(\log \frac{1}{\epsilon})$  di iterazioni: ha cioè complessità lineare nel numero di cifre significative che richiediamo all'approssimazione.

### 1.2.3 Modello del primo ordine

Analizziamo ora un metodo che tiene conto della derivata. Ricordiamo preliminarmente il teorema di Taylor:

**Teorema 1.2.1. (Polinomio di Taylor)** *Sia  $f : X \rightarrow \mathbb{R}$  una funzione derivabile  $n$  volte in  $x_0 \in X$ . Chiamiamo  $f^{(k)}$  la  $k$ -esima derivata di  $f$  in  $x_0$ , e denotiamo  $f^{(0)}(x_0) = f(x_0)$ . Allora vale per ogni  $x \in X$ :*

$$f(x) = \sum_{k=0}^n f^{(k)}(x_0) \frac{(x - x_0)^k}{k!} + o((x - x_0)^n)$$

In pratica possiamo approssimare la funzione in un punto  $x$  vicino a  $x_0$  con un polinomio i cui coefficienti sono dati dalle derivate successive di  $f$ , e l'errore che commettiamo in tale approssimazione è più piccolo della distanza tra  $x$  e  $x_0$  elevata alla  $n$ .

Per  $n = 1$  si ha che

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + o(x - x_0),$$

perciò la funzione lineare  $f(x_0) + f'(x_0)(x - x_0)$  ci fornisce un *modello* (al primo ordine) della funzione  $f$  da minimizzare. La derivata in un punto esprime il coefficiente angolare della retta tangente in quel punto: ci dice cioè in quale direzione la funzione sta crescendo e in quale sta decrescendo, e con quale

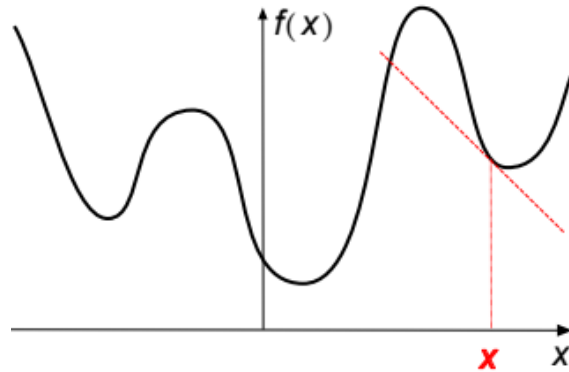


Figura 1.3: La derivata di una funzione in un punto ci dice la pendenza della retta tangente in quel punto

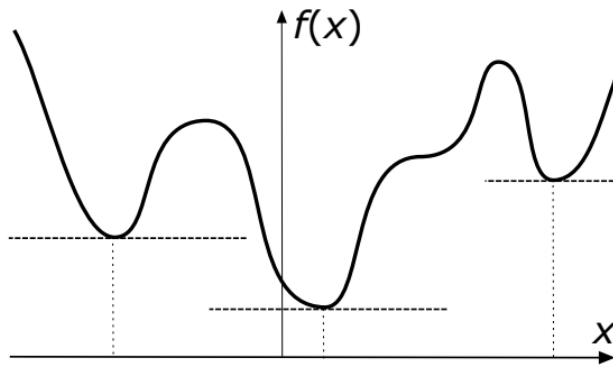


Figura 1.4: La derivata si annulla nei minimi locali

velocità (Figura 1.3). Nei minimi locali in particolare la derivata è sempre nulla (Figura 1.4), per il noto Teorema di Fermat sui punti stazionari. La condizione che la derivata si annulli è solo necessaria, ma non sufficiente: la derivata può essere nulla anche in altri punti, come ad esempio nei massimi locali.

### Ricerca dicotomica:

L'ultima osservazione ci porta a sviluppare un nuovo algoritmo per minimizzare una funzione:

- Supponiamo innanzitutto che  $f'$  sia continua in  $X$ , ovvero che  $f \in C^1$  (cioè  $f$  è derivabile una volta e la sua derivata è continua).
- Supponiamo inoltre di trovare due punti  $x_- < x_+$  con  $f'(x_-) < -\epsilon$  e  $f'(x_+) > \epsilon$ .

Allora possiamo procedere in questo modo:

1. Ad ogni passo scegliamo un punto  $x_c \in (x_-, x_+)$ . Se  $-\epsilon < f'(x_c) < \epsilon$ , fermiamo il procedimento.
2. Altrimenti se  $f'(x_c) < -\epsilon$  ricominciamo con  $x_- = x_c$
3. Altrimenti (  $f'(x_c) > \epsilon$  )ricominciamo con  $x_+ = x_c$

Il vantaggio di questo algoritmo è che ad ogni passo dobbiamo fare solo una valutazione di funzione (in questo caso  $f'$ ). Possiamo inoltre scegliere un passo di esattamente  $\frac{1}{2}$ , cioè  $x_c = (x_- + x_+)/2$ , perciò questo metodo ad ogni passo dimezziamo esattamente la lunghezza dell'intervallo considerato

*Osservazione 1.* Il criterio di stop che usiamo è  $|f'(x_c)| < \epsilon$ , che ci garantisce che  $|f(x_*) - f(x_c)| \leq \epsilon|x_* - x_c| + o(|x_* - x_c|)$ , ovvero  $f(x_c)$  è ragionevolmente vicino al minimo locale.

### 1.2.4 Modello del secondo ordine e metodo di Newton

**Idea:** Scegliere  $x_c$  esattamente al centro di  $[x_-, x_+]$  non è la scelta più furba da fare: notiamo che stiamo usando infatti solo il segno della derivata nel passo 1. (o meglio il fatto che sia maggiore di  $\epsilon$  in valore assoluto) e non il suo valore effettivo.

#### Interpolazione quadratica

Invece che un modello lineare vogliamo ora un modello quadratico  $p$  (un polinomio di secondo grado) che approssimi la nostra funzione nel nostro intervallo. Idealmente vorremmo un modello che coincida con  $f$  negli estremi  $x_-$  e  $x_+$ : vorremmo cioè  $p(x) = ax^2 + bx + c$ , dove  $a, b, c$ , sono le nostre incognite, tale che

$$\begin{cases} p(x_-) = f(x_-) \\ p(x_+) = f(x_+) \\ p'(x_-) = f'(x_-) \\ p'(x_+) = f'(x_+). \end{cases}$$

Visto che le incognite sono 3 e le equazioni sono 4, il sistema è “sovradeterminato” (anche se non è un sistema lineare) perciò dobbiamo lasciar cadere qualche ipotesi. In particolare richiediamo solo che  $p'(x_-) = f'(x_-)$  e  $p'(x_+) = f'(x_+)$ . In questo caso quindi

$$\begin{aligned} p'(x_-) &= 2ax_- + b = f'(x_-) \\ p'(x_+) &= 2ax_+ + b = f'(x_+) \end{aligned}$$

da cui otteniamo svolgendo i sistemi che

$$\begin{aligned} a &= \frac{f'(x_+) - f'(x_-)}{2(x_+ - x_-)} \\ b &= \frac{x_+ f'(x_-) - x_- f'(x_+)}{x_+ - x_-}. \end{aligned}$$

Per minimizzare  $p$  basta porre la derivata  $ax + b = 0$ . ciò accade in

$$x = \frac{x_+ f'(x_-) - x_- f'(x_+)}{f'(x_+) - f'(x_-)}.$$

Se  $f'(x)$  è abbastanza piccola ci fermiamo, se no come al solito se  $f'(x) < \epsilon$  sostituiamo  $x$  a  $x_-$  e viceversa se  $f'(x) > \epsilon$  sostituiamo  $x$  a  $x_+$ . Iterando questo procedimento troviamo un punto in cui  $f'(x) \approx 0$ , e quindi un minimo locale.

*Osservazione 2.* Il procedimento descritto equivale all'applicare il metodo delle secanti alla *derivata* di  $f$  (Figura 1.5).

### Metodo di Newton

Un altro modo di trovare un modello del secondo ordine è applicando il Teorema 1.2.1 con  $n = 2$ :

$$p_{x_0}(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)(x - x_0)^2}{2}$$

per qualche  $x_0 \in X$ . Questo nuovo modello ci porta al metodo di Newton (o delle tangenti): Partiamo da  $x_0 \in X$  e per ogni  $i = 0, 1, 2, \dots$

- Se  $|f'(x_i)| < \epsilon$  ci fermiamo

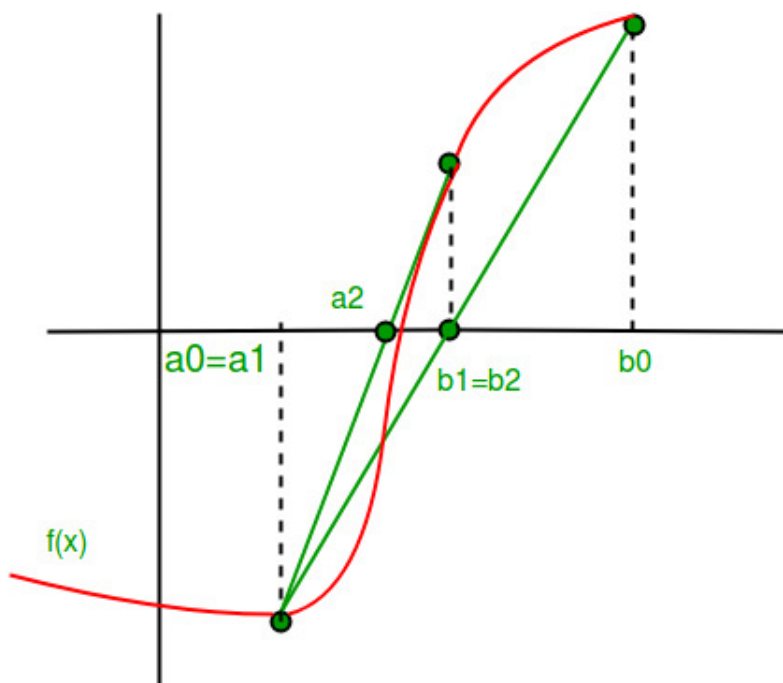


Figura 1.5: Il metodo delle secanti è un semplice metodo per risolvere un'equazione omogenea  $g(x) = 0$ . Nel nostro caso  $g = f'$ .

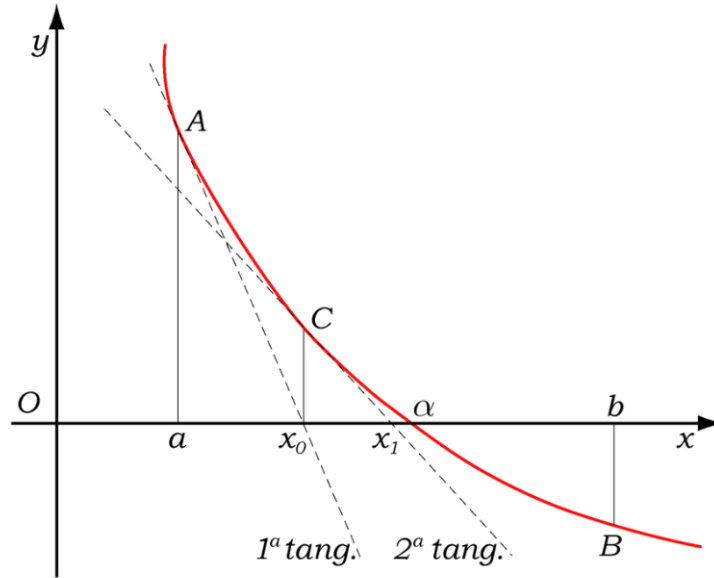
- Altrimenti minimizziamo

$$p_{x_i}(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2$$

Tale minimo sarà assunto quando  $p'_{x_i}(x) = f'(x_i) + f''(x_i)(x - x_i) = 0$  cioè in

$$x_{i+1} = \frac{x_i - f'(x_i)}{f''(x_i)}.$$

Possiamo quindi ripetere il procedimento finché  $|f'(x_i)|$  non è abbastanza piccolo. Questo metodo è detto *metodo di Newton* o *delle tangenti*, poiché stiamo cercando di risolvere l'equazione  $f'(x) = 0$  andando a vedere ad ogni passo dove si annulla la retta tangente alla funzione nel punto  $x_i$  (Figura 1.6). Il metodo di Newton converge se  $x_0$  è abbastanza vicino al punto di minimo  $x_*$ .

Figura 1.6: Metodo delle tangenti applicato ad una funzione  $g$ 

### 1.2.5 Misurare la velocità di convergenza

Iniziamo una breve discussione su come misurare la velocità di convergenza di algoritmi iterativi (tutti gli algoritmi che abbiamo visto fin'ora).

**Definizione. (Tasso di convergenza):** Sia  $f_i = f(x_i)$  la sequenza generata da un algoritmo iterativo e sia  $f_*$  il minimo locale della funzione a cui  $f_i$  converge (cioè  $\lim_{i \rightarrow \infty} f_i = f_*$ ; supponiamo inoltre che esistano delle costanti  $0 < r \leq 1$ ,  $t \geq 1$  tali che

$$\lim_{i \rightarrow \infty} \frac{f_{i+1} - f_*}{(f_i - f_*)^t} = r$$

Possiamo distinguere fondamentalmente 3 casi:

$t = 1, r = 1$ : In questo caso parliamo di tasso di convergenza *sublineare*. In tal caso la sequenza  $f_i - f_*$  converge a 0 molto lentamente: più passi facciamo meno miglioriamo in proporzione la nostra stima.

$t = 1, 0 < r < 1$ : In questo caso parliamo di convergenza *lineare*: il numero di cifre significative cresce in modo lineare. Questo caso può essere buono se  $r \ll 1$ , mentre se  $r \approx 1$  la convergenza può risultare molto lenta.

$t = 1, r = 0$ : In questo caso parliamo di convergenza *superlineare*; in particolare se vale anche che per  $t = 2$  il rapporto converge a un  $r > 0$ , si parla di convergenza quadratica; è il caso in cui il metodo converge più rapidamente.

In figura vediamo come si comportano (in scala semilogaritmica) diverse funzioni che convergono a 0 con tasso sublineare, lineare e superlineare.

Con questa definizione si può vedere che la golden ratio search e la ricerca dicotomica sono lineari con  $r = \frac{\sqrt{5}-1}{2} \approx 0.618$  e  $r = 0.5$  rispettivamente. Il metodo delle secanti è anch'esso lineare al caso pessimo, per esempio quando la funzione è molto inclinata (skewed) verso uno dei due estremi. Vediamo invece che in determinate ipotesi il metodo di Newton è quadratico:

**Teorema 1.2.2.** *Supponiamo che  $f$  sia derivabile 3 volte e che la sua derivata terza sia continua ( $f \in C^3$ ) e che  $x_*$  sia un minimo locale con  $f''(x_*) \neq 0$ . Allora esiste un  $\delta > 0$  tale per cui se scegliamo un qualsiasi  $x_0 \in [x_* - \delta, x_* + \delta]$  il metodo di Newton converge con tasso quadratico partendo da  $x_0$ .*

*Dimostrazione.* Siano  $\{x_i\}$  i passi generati dal metodo di Newton.  $x_{i+1} = x_i - f'(x_i)/f''(x_i)$ , quindi, ricordando che  $f'(x_*) = 0$ ,

$$x_{i+1} - x_* = x_i - x_* + \frac{f'(x_*) - f'(x_i)}{f''(x_i)} = \frac{f'(x_*) - f(x_i) - f''(x_i)(x_* - x_i)}{f''(x_i)}$$

Per la formula di Taylor (con resto di Lagrange) applicata a  $f'$ , esiste un  $y$  tra  $x_i$  e  $x_*$  tale che

$$0 = f'(x_*) = f'(x_i) + f''(x_i)(x_* - x_i) + \frac{f'''(y)(x_* - x_i)^2}{2},$$

dunque  $x_{i+1} - x_* = (f'''(y)/(2f''(x_i)))(x_i - x_*)^2$ . Poiché  $x_*$  è un minimo locale,  $f''(x_*) > 0$ , e dunque esistono un  $\delta > 0$  e due costanti  $k_1$  per cui  $k_1 \leq f''(x)$  e  $|f'''(x)| \leq k_2$  (per continuità di  $f'''$ ) per ogni  $x \in [x_* - \delta, x_* + \delta]$ . Dunque

$$|x_{i+1} - x_*| \leq \frac{k_1}{2k_2}(x_i - x_*)^2.$$

Ciò prova il tasso di convergenza quadratico. □



# Capitolo 2

## Ottimizzazione non vincolata

Passiamo ora al caso in cui la funzione da ottimizzare è una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . Chiameremo  $\mathbf{x}$  in grassetto un vettore di  $\mathbb{R}^n$  e  $(x_1, \dots, x_n)$  le sue componenti (a differenza di prima dove  $x_i$  erano le iterazioni di un metodo iterativo)

### 2.1 Richiami di analisi in più variabili

#### 2.1.1 Prodotti scalari, norme, distanze

**Definizione.** Siano  $\mathbf{x}, \mathbf{y}$  due vettori in  $\mathbb{R}^n$ . Il loro *prodotto scalare* (standard) è

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i.$$

La *norma euclidea* di  $\mathbf{x}$  è

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^n x_i^2}$$

Si può generalizzare il concetto di norma euclidea.

**Definizione.** Una *norma* su  $\mathbb{R}^n$  è una funzione  $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$  con le seguenti proprietà: per ogni  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

- $\|\mathbf{x}\| \geq 0$  e  $\|\mathbf{x}\| = 0$  se e solo se  $\mathbf{x} = \mathbf{0} = (0, \dots, 0)$

- Per ogni  $\alpha \in \mathbb{R}$   $\|\alpha \mathbf{x}\| = |\alpha| \|\mathbf{x}\|$
- (disuguaglianza triangolare)  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

È facile vedere che la norma euclidea è in effetti una norma. Più precisamente fa parte della famiglia di norme  $L_p$ , cioè della forma

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

Queste funzioni sono vere e proprie norme per ogni  $p \geq 1$ , mentre per  $p < 1$  manca la proprietà di disuguaglianza triangolare. La norma euclidea corrisponde alla norma  $L_2$ . Per  $p \rightarrow \infty$  definiamo la norma infinito come il limite di funzioni delle norme  $L_p$ . È facile dimostrare che

$$\|\mathbf{x}\|_\infty = \max_{i=1, \dots, n} |x_i|$$

**Definizione.** Una *distanza* è una qualsiasi funzione  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  con le seguenti proprietà:

- $d(\mathbf{x}, \mathbf{y}) \geq 0$  e  $d(\mathbf{x}, \mathbf{y}) = 0$  se e solo se  $\mathbf{x} = \mathbf{y}$
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$  (Simmetria)
- $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$  (Disuguaglianza triangolare per le distanze)

**Definizione.** Data una qualsiasi norma  $\|\cdot\|$ , la *distanza indotta* da tale norma è la funzione  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  così definita:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$$

Si verifica facilmente che le proprietà della norma rendono  $d$  effettivamente una distanza, cioè  $d$  soddisfa le 3 proprietà della definizione di distanza.

**Definizione.** Data una distanza  $d$  una palla (aperta) di centro  $\mathbf{x}$  e raggio  $R > 0$  è l'insieme  $\mathcal{B}(\mathbf{x}, R) \subseteq \mathbb{R}^n$  definito da

$$\mathcal{B}(\mathbf{x}, R) = \{\mathbf{y} : d(\mathbf{x}, \mathbf{y}) < R\}$$

Nel seguito quando useremo la notazione  $\|\cdot\|$  intenderemo la norma euclidea, e  $d$  sarà l'usuale distanza euclidea

$$d(\mathbf{x}, \mathbf{y}) = \|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

### 2.1.2 Gradiente, Hessiana, Jacobiana

**Definizione. (Tomografia di una funzione)** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  la tomografia di una funzione data un'origine  $\mathbf{x}$  e una direzione  $\mathbf{d}$  è

$$\varphi_{\mathbf{x},\mathbf{d}}(t) = f(\mathbf{x} + t\mathbf{d})$$

Come per funzioni in più variabili possiamo definire il concetto di derivabilità e differenziabilità per funzioni di più variabili:

**Definizione. Derivata direzionale:**

Sia  $\mathbf{d}$  un vettore in  $\mathbb{R}^n$ . La derivata di  $f$  in  $\mathbf{x}$  lungo la direzione  $\mathbf{d}$  è

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{d}} = \lim_{t \rightarrow 0} \frac{f(\mathbf{x} + t\mathbf{d}) - f(\mathbf{x})}{t} = \varphi'_{\mathbf{x},\mathbf{d}}(0)$$

se tale limite esiste ed è finito.

In particolare se  $\mathbf{d} = \mathbf{e}^i$  è un vettore della base canonica di  $\mathbb{R}^n$ , ovvero  $\mathbf{e}_j^i = 0$  per ogni  $j \neq i$  e  $\mathbf{e}_i^i = 1$ , se calcoliamo la derivata direzionale rispetto a  $\mathbf{e}^i$ , facciamo variare  $f$  solo rispetto alla variabile  $x_i$ , perciò denotiamo la derivata direzionale come

$$\frac{\partial f(\mathbf{x})}{\partial x_i}$$

e la chiamiamo *derivata parziale (i-esima)*

Nella pratica per calcolare la i-esima derivata parziale basta calcolare

$$f'(x_1, \dots, x_{i-1}, x, x_{i+1}, \dots, x_n)$$

rispetto a  $x$ , trattando le altre variabili come costanti.

**Definizione.** Il gradiente di  $f$  è il vettore delle derivate parziali:

$$\nabla f(\mathbf{x}) = \left( \frac{\partial f(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right).$$

**Definizione.** Diciamo che una funzione è *differenziabile* in un punto  $\mathbf{x}$  se esiste una funzione lineare  $\phi$  tale che per ogni  $\mathbf{y}$

$$f(\mathbf{y}) = f(\mathbf{x}) + \phi(\mathbf{y} - \mathbf{x}) + R(\mathbf{y}),$$

con  $R(\mathbf{y})$  (il resto) che tende a 0 più velocemente di  $\|\mathbf{y} - \mathbf{x}\|_2$  se  $\mathbf{y} \rightarrow \mathbf{x}$ , cioè  $\frac{R(\mathbf{x})}{\|\mathbf{y} - \mathbf{x}\|} \xrightarrow{\mathbf{y} \rightarrow \mathbf{x}} 0$ .

Se  $f$  è differenziabile, allora  $\phi(\mathbf{y}) = \nabla f(\mathbf{x}) \cdot \mathbf{y}$ . Perciò se  $f$  è differenziabile in  $\mathbf{x}$  allora esistono tutte le derivate parziali  $\frac{\partial f(\mathbf{x})}{\partial x_i}$ , e  $f(\mathbf{y}) + \nabla f(\mathbf{x}) \cdot \mathbf{y}$  è un modello al primo ordine di  $f$ .

**Proposizione 2.1.1.** *Se  $f$  è differenziabile in  $\mathbf{x}$ , per ogni direzione  $\mathbf{d}$  si ha che la derivata direzionale lungo  $\mathbf{d}$  è data dal prodotto scalare tra il gradiente di  $f$  in  $\mathbf{x}$  e  $\mathbf{d}$ :*

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{d}} = \nabla f(\mathbf{x}) \cdot \mathbf{d}$$

*Dimostrazione.* (non necessaria) Si usa la chain rule:

$$\begin{aligned} \frac{\partial f(\mathbf{x})}{\partial \mathbf{d}} &= \lim_{t \rightarrow 0} \frac{f(\mathbf{x} + t\mathbf{d}) - f(\mathbf{x})}{t} = \frac{df(\mathbf{x} + t\mathbf{d})}{dt}(t = 0) \\ &= \nabla f(\mathbf{x}) \frac{d(\mathbf{x} + t\mathbf{d})}{dt} = \nabla f(\mathbf{x}) \cdot \mathbf{d} \end{aligned}$$

□

L'esistenza di tutte le derivate parziali non indica in generale che la funzione sia differenziabile: in  $\mathbb{R}^n$ , con  $n > 2$ , “derivabilità  $\neq$  differenziabilità”. Tuttavia vale il teorema seguente:

**Teorema 2.1.1. (differenziale totale)** *Se esistono le derivate parziali in un punto  $\mathbf{x}$  e esiste un  $\delta > 0$  tale che le derivate parziali sono tutte continue nella palla  $\mathcal{B}(\mathbf{x}, \delta)$ , allora  $f$  è differenziabile in  $\mathbf{x}$*

Cioè se le derivate parziali sono continue, allora  $f$  è differenziabile.

**Definizione.** Se  $F$  è una funzione a valori vettoriali, cioè  $F = (F_1, \dots, F_m) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , possiamo definire la matrice Jacobiana come la matrice  $n \times m$  delle derivate parziali di ciascuna  $f_i$ :

$$J_F(\mathbf{x}) = \begin{pmatrix} \frac{\partial F_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial F_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

Un particolare tipo di matrice jacobiana è la *matrice hessiana* di una funzione in più variabili ovvero la matrice delle derivate seconde:

$$\nabla^2(\mathbf{x}) = J_{\nabla f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{pmatrix}$$

In generale le derivate parziali non commutano:  $\frac{\partial^2 f}{\partial x_i \partial x_j} \neq \frac{\partial^2 f}{\partial x_j \partial x_i}$ . Tuttavia l'uguaglianza vale se  $f \in C^2$ . Perciò se  $f \in C^2$ , la matrice hessiana è *simmetrica*.

### 2.1.3 Condizioni di ottimalità al primo e secondo ordine

Analogamente al caso unidimensionale, se  $\mathbf{x}$  è un minimo locale, allora il gradiente in  $\mathbf{x}$  si annulla. Più precisamente:

**Teorema 2.1.2.** *Se  $f$  è differenziabile in  $\mathbf{x}$  e  $\mathbf{x}$  è un minimo locale, allora  $\nabla f(\mathbf{x}) = 0$ .*

*Dimostrazione.* Supponiamo per assurdo che  $\mathbf{x}$  sia minimo locale e  $\nabla f(\mathbf{x}) \neq 0$ . Allora la derivata direzionale lungo la direzione  $\mathbf{d} = \frac{-\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$  è negativa: infatti per la Proposizione 2.1.1,

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{d}} = \nabla f(\mathbf{x}) \cdot \mathbf{d} = \nabla f(\mathbf{x}) \cdot \frac{-\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} = \frac{\|\nabla f(\mathbf{x})\|^2}{\|\nabla f(\mathbf{x})\|} = -\|\nabla f(\mathbf{x})\|.$$

Dunque per la differenziabilità di  $f$  si ha che per qualsiasi  $\epsilon$  abbastanza piccolo, se  $\mathbf{y} = \mathbf{x} + \epsilon \mathbf{d}$

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x}) \epsilon \mathbf{d} + R(\mathbf{y}) = f(\mathbf{x}) - \epsilon \|\nabla f(\mathbf{x})\| + R(\mathbf{y})$$

Ma abbiamo detto che

$$\frac{R(\mathbf{y})}{\|\mathbf{y} - \mathbf{x}\|} = \frac{R(\mathbf{y})}{\epsilon} \xrightarrow{\epsilon \rightarrow 0} 0$$

perciò esiste  $\epsilon'$  tale che, per ogni  $\epsilon < \epsilon'$ ,  $R(\mathbf{y}) \leq \frac{\nabla f(\mathbf{x})\epsilon}{2}$  e dunque

$$f(\mathbf{y}) = f(\mathbf{x}) - \nabla f(\mathbf{x}) \epsilon \mathbf{d} + R(\mathbf{y}) \leq f(\mathbf{x}) - \nabla f(\mathbf{x}) \frac{\epsilon}{2} < f(\mathbf{x}).$$

Ciò contraddice però che  $\mathbf{x}$  sia un minimo locale. □

*Osservazione 3.*  $\mathbf{d}$  definita prima è la direzione di massima decrescita della funzione  $f$ , e  $\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$  è la direzione di massima crescita: infatti per ogni direzione  $\mathbf{d}'$  con  $\|\mathbf{d}'\| = 1$  si ha che

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{d}'} = \mathbf{d}' \cdot \nabla f(\mathbf{x}) = \|\mathbf{d}'\| \|\nabla f(\mathbf{x})\| \cos \theta = \|\nabla f(\mathbf{x})\| \cos \theta.$$

Il massimo di questa funzione è assunto quando  $\cos \theta = 1$ , cioè quando  $\mathbf{d}'$  è parallelo a  $\nabla f(\mathbf{x})$  e con lo stesso verso, cioè quando  $\mathbf{d}' = \nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\|$ . Viceversa il minimo è assunto quando  $\mathbf{d}' = -\nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\| = \mathbf{d}$ .

Il Teorema 2.1.2 ci dice solo che se  $\mathbf{x}_*$  è un punto di minimo locale allora  $\nabla f(\mathbf{x}_*) = 0$ . Esistono però altri punti stazionari, cioè tali che  $\nabla f(\mathbf{x}) = 0$  che non sono minimi locali: in particolare abbiamo massimi locali e punti di sella. Andando a studiare l'hessiana della funzione si possono tuttavia trovare delle condizioni più precise per assicurarci che il punto stazionario sia effettivamente un minimo:

**Teorema 2.1.3.** *Se  $f \in C^2$*

1. *Se  $x_*$  è un minimo locale di  $f$ , allora  $\nabla^2 f(\mathbf{x}_*)$  è semidefinita positiva.*
2. *Viceversa vale che se  $\mathbf{x}_*$  è un punto stazionario e  $\nabla^2 f(\mathbf{x}_*)$  è definita positiva, allora  $x_*$  è un minimo locale*

.

*Dimostrazione.* Dimostriamo la prima delle due tesi. Abbiamo che vale la formula di Taylor del secondo ordine

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + (\mathbf{y} - \mathbf{x})^T \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + o(\|\mathbf{y} - \mathbf{x}\|^2)$$

Se  $\mathbf{x} = \mathbf{x}_*$  è minimo, in particolare è stazionario e vale quindi che

$$f(\mathbf{y}) = f(\mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T \nabla^2 f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + o(\|\mathbf{y} - \mathbf{x}\|^2) \quad (2.1)$$

Se  $\nabla^2 f(\mathbf{x})$  non fosse semidefinita positiva, esisterebbero uno scalare  $\lambda > 0$  e una direzione  $\mathbf{d}$  tale per cui

$$\mathbf{d}^T \nabla^2 f(\mathbf{x}) \mathbf{d} = -\lambda \|\mathbf{d}\|^2 < 0$$

Prendendo allora  $\mathbf{y} = \mathbf{x} + \epsilon \mathbf{d}$  per ogni  $\epsilon > 0$  abbastanza piccolo, si ha che

$$\begin{aligned} f(\mathbf{y}) &= f(\mathbf{x}) - \frac{1}{2} \epsilon \lambda \|\mathbf{d}\|^2 + o(\epsilon^2) \\ &\leq f(\mathbf{x}) - \frac{1}{4} \epsilon \lambda \|\mathbf{d}\|^2 < f(\mathbf{x}) \end{aligned}$$

Ma ciò è assurdo perchè  $\mathbf{x}$  è un minimo.

Proviamo ora la seconda delle due tesi. Se  $\mathbf{x}$  è un punto stazionario vale ancora l'equazione (2.1). Poiché  $\nabla^2 f(\mathbf{x})$  è definita positiva, esiste una costante  $c \geq 0$  tale per cui per ogni direzione  $\mathbf{d}$

$$\mathbf{d}^T \nabla^2 f(\mathbf{x}) \mathbf{d} \geq c > 0.$$

Prendendo  $\mathbf{y} = \mathbf{x} + \epsilon \mathbf{d}$  per ogni  $\mathbf{d}$  e per ogni  $\epsilon > 0$  abbastanza piccolo, si ha che

$$f(\mathbf{y}) = f(\mathbf{x}) + \frac{1}{2} \epsilon c + o(\epsilon^2) \geq f(\mathbf{x}) + \frac{1}{4} \epsilon c > f(\mathbf{x})$$

Abbiamo perciò dimostrato che esiste un  $\epsilon$  tale per cui  $f(\mathbf{x}) < f(\mathbf{y})$  per ogni  $\mathbf{y} \in \mathcal{B}(\mathbf{x}, \epsilon)$ .  $\square$

### 2.1.4 Funzioni convesse

Se vogliamo garantire che quando troviamo un punto stazionario, questo sia un punto di minimo locale, dobbiamo verificare che l'hessiana in tale punto sia semidefinita positiva. Nel caso delle funzioni convesse questa convinzione è sempre verificata.

**Definizione.** Una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ <sup>1</sup> è *convessa* se per ogni  $\mathbf{x}, \mathbf{y} \in X$  e per ogni  $t \in [0, 1]$

$$f(t\mathbf{x} + (1-t)\mathbf{y}) \leq t f(\mathbf{x}) + (1-t)f(\mathbf{y})$$

Se vale la disuguaglianza opposta allora la funzione si dice *concava*. Enunciamo ora alcune proprietà importanti delle funzioni convesse:

**Proposizione 2.1.2.**  $f$  convessa  $\iff (\nabla f(\mathbf{x}) - \nabla f(\mathbf{z})) \cdot (\mathbf{x} - \mathbf{z}) \geq 0$

**Proposizione 2.1.3.** Se  $f \in C^2$ , allora  $f$  è convessa se e solo se la sua matrice hessiana è semidefinita positiva in ogni punto.

In generale non vale che l'hessiana di  $f$  sia definita positiva in ogni punto, ma si può comunque dimostrare che ogni punto stazionario è un minimo *globale*:

---

<sup>1</sup>In teoria potremmo definire  $f$  anche solo su un insieme  $X \subseteq \mathbb{R}^n$  convesso, ovvero tale per cui, dati due suoi punti qualsiasi, l'intero segmento tra essi è contenuto in  $X$ . La definizione formale di insieme convesso e le relative proprietà sono date nella sezione sull'ottimizzazione vincolata.

**Proposizione 2.1.4.**  $\mathbf{x}$  è un punto stazionario di  $f$  se e solo se per ogni  $\mathbf{z}$   $f(\mathbf{z}) \geq f(\mathbf{x})$

Possiamo anche rinforzare la definizione di convessità richiedendo qualcosa di più

**Definizione.** Una funzione  $f$  è *strettamente convessa* se è convessa e vale per ogni  $t \notin \{0, 1\}$

$$f(t\mathbf{x} + (1-t)\mathbf{y}) < tf(\mathbf{x}) + (1-t)f(\mathbf{y})$$

*Esempio.* Ogni funzione  $f$  lineare è convessa ma non strettamente convessa: vale infatti l'=" nella definizione di convessità per ogni  $t$ .

*Esempio.* Una funzione quadratica  $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x}$  è strettamente convessa se e solo se  $Q$  è definita positiva.

Ovviamente se  $f$  è strettamente convessa allora è anche convessa, ma abbiamo visto con l'esempio delle funzioni lineari che non vale il viceversa. Inoltre se  $f$  è strettamente convessa, allora ha un unico punto stazionario, e tale punto è dunque *l'unico minimo globale*. Una definizione ancora più forte è la seguente.

**Definizione.** Una funzione  $f$  convessa si dice *fortemente convessa* o  $\tau$ -convessa con modulo  $\tau$  se  $f(\mathbf{x}) - \frac{\tau}{2}\|\mathbf{x}\|^2$  è convessa.

*Esempio.* Se  $Q$  è definita positiva,  $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x}$  è  $\tau$ -convessa con

$$\tau = \min_{i=1, \dots, n} \lambda_i,$$

dove i  $\lambda_i$  sono gli autovalori di  $Q$ .

*Esempio.*  $-\log(x)$  è una funzione strettamente convessa, ma non  $\tau$ -convessa.

**Proposizione 2.1.5.**  $f \in C^2$  è  $\tau$ -convessa se e solo se gli autovalori di  $\nabla^2(f\mathbf{x})$  sono sempre  $\geq \tau$  al variare di  $\mathbf{x}$

*Dimostrazione.*  $g(\mathbf{x}) = f(\mathbf{x}) - \frac{\tau}{2}\|\mathbf{x}\|^2$  è convessa e quindi  $\nabla^2 g(\mathbf{x}) = \nabla^2 f(\mathbf{x}) - \tau I$  è semi-definita positiva, perciò detto  $\lambda$  l'autovalore minimo di  $\nabla^2 f(\mathbf{x})$ ,  $\lambda - \tau \geq 0$ , cioè  $\lambda \geq \tau$ .  $\square$



## 2.2 Algoritmo della discesa del gradiente

Abbiamo visto che dato un punto  $\mathbf{x}$  in cui  $f$  è differenziabile, abbiamo un modello del primo ordine di  $f$  dato da

$$f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y} - \mathbf{x})$$

Sottolineiamo che questo modello per la maggior parte delle volte è buono solo se siamo molto vicini a  $\mathbf{x}$  ( in una palla  $\mathcal{B}(\mathbf{x}, \epsilon)$  con  $\epsilon$  molto piccolo).

L'algoritmo della discesa del gradiente si basa su questo modello, ed è il seguente:

- Scegliamo un vettore iniziale  $\mathbf{x}^0 \in \mathbb{R}^n$ , poi per ogni  $i = 0, 1, 2, \dots$ ,
- se  $\|\nabla f(\mathbf{x}^i)\| < \epsilon$  ci fermiamo,
- altrimenti ad ogni iterazione scegliamo un' ampiezza del passo  $\alpha_i$  e aggiorniamo il vettore in questo modo:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \alpha \nabla f(\mathbf{x}^i) = \mathbf{x}^i + \alpha \mathbf{d}^i.$$

Come abbiamo già osservato prima questo metodo ha senso perché  $\mathbf{d} = -\nabla f(\mathbf{x})$  è la direzione in cui la funzione decresce di più in  $\mathbf{x}$ .

---

### Algoritmo 1 Algoritmo della Discesa del gradiente

---

<pre> <b>procedure</b> <math>x = SDQ(f, x, \epsilon)</math>   <b>while</b> (<math>\ \nabla f(x)\  &gt; \epsilon</math>) <b>do</b>     <math>d \leftarrow -\nabla f(x)</math>; <math>\alpha \leftarrow \text{stepsize}(f, x, d)</math>; <math>x \leftarrow x + \alpha d</math>;           </pre>
---

---

La domanda che ci poniamo subito è come scegliere l'ampiezza del passo  $\alpha$ , ovvero come implementare **stepsize**( $f, \mathbf{x}, \mathbf{d}$ ).

Durante l'esecuzione di questo algoritmo infatti, se scegliamo male  $\alpha$  possiamo imbatterci in due problemi opposti <sup>2</sup> :

**Scilla:** Se  $\alpha_i$  è troppo grande, rischiamo che  $f(\mathbf{x}^{i+1}) \gg f(\mathbf{x}^i)$  e l'algoritmo può anche divergere senza trovare il minimo

---

<sup>2</sup>[https://www.elicriso.it/it/mitologia\\_ambiente/scilla\\_cariddi/](https://www.elicriso.it/it/mitologia_ambiente/scilla_cariddi/)

**Cariddi:** Viceversa se  $\alpha^i$  è troppo piccolo, i passi fatti dall'algoritmo saranno molto piccoli e rischiamo di non arrivare al minimo locale a cui stiamo puntando.

In linea di massima  $\alpha_i$  dovrebbero decrescere ma non troppo velocemente:

*Esempio.* Se  $f(\mathbf{x}) = -\mathbf{x}$ ,  $\mathbf{x}^0 = 0$ ,  $\alpha_i = \frac{1}{i^2}$ , per ogni passo si ha  $\mathbf{d}^i = -f'(\mathbf{x}) = 1$  e quindi

$$\mathbf{x}^i = \sum_{j=1}^i \frac{1}{j^2} \xrightarrow{i \rightarrow \infty} \frac{\pi^2}{6} \approx 1.645,$$

mentre evidentemente  $f_* = -\infty$ . Se invece fosse  $\alpha_i = \frac{1}{i}$ , allora

$$\mathbf{x}^i = \sum_{j=1}^i \frac{1}{j} \xrightarrow{i \rightarrow \infty} +\infty$$

e perciò effettivamente si ha che  $f(\mathbf{x}^i) \rightarrow -\infty$

### 2.2.1 Metodo del gradiente con passo costante

spesso nel Machine Learning quando si implementa la discesa del gradiente si sceglie un passo, detto anche *learning rate*,  $\alpha_i = \alpha$ .

Iniziamo con un'osservazione: se vogliamo che la successione  $\mathbf{x}^i$  tenda a un punto  $\mathbf{x}_*$ , dev'essere necessariamente  $\|\mathbf{x}^{i+1} - \mathbf{x}^i\| \rightarrow 0$  (ovvero la successione  $\{\|\mathbf{x}^i\|\}$  è *di Cauchy*). Se prendessimo come direzione  $\mathbf{d}^i = -\frac{\nabla f(\mathbf{x}^i)}{\|\nabla f(\mathbf{x}^i)\|}$ , ovviamente  $\|\mathbf{d}^i\| = 1$  e allora

$$\|\mathbf{x}^{i+1} - \mathbf{x}^i\| = \|\alpha_i \mathbf{d}^i\| = |\alpha_i| \|\mathbf{d}^i\| = |\alpha_i| = |\alpha|, \quad (2.2)$$

cioè la differenza in norma tra i vettori dell'iterazione è costante.

Prendiamo perciò  $\mathbf{d}^i = -\nabla f(\mathbf{x}^i)$  senza normalizzare il vettore. In questo modo cambiando il penultimo passaggio di (2.2) si ha che  $\|\mathbf{x}^{i+1} - \mathbf{x}^i\| = |\alpha_i| \|\mathbf{d}^i\| = |\alpha| \|\mathbf{d}^i\|$ , che tende a 0 se  $\|\mathbf{d}^i\| = \|\nabla f(\mathbf{x}^i)\| \rightarrow 0$ , cioè se ci siamo avvicinando ad un punto stazionario.

Ci chiediamo ora quanto piccolo debba essere  $\alpha$  per garantire che il metodo converga a un punto stazionario. Intuitivamente, se  $f$  varia molto rapidamente,  $\alpha$  dovrà essere molto piccola. Assumiamo perciò come prima cosa che  $f$  sia L-lipchitz.

*Osservazione 4.* Se  $f$  è differenziabile, allora è L-lipschitz se e solo se  $\|\nabla f\| \leq L$ .

In realtà per controllare che  $f$  non vari troppo rapidamente ci serve controllare che  $\nabla f$  sia anch'esso lipschitz:

**Definizione.**  $f$  è  $L$ -smooth se  $\nabla f$  è  $L$ -lipschitz.

In particolare se  $f \in C^2$ , similmente all'osservazione precedente  $f$  è  $L$ -smooth se gli autovalori di  $\nabla^2 f(\mathbf{x})$  al variare di  $\mathbf{x}$  sono compresi in  $[-L, L]$  (ciò corrisponde al fatto che la norma di  $\nabla^2 f(\mathbf{x})$  sia limitata).

Ricordiamo che dati due vettori  $\mathbf{x}$  e  $\mathbf{d}$  la tomografia di  $f$  è la funzione in una variabile  $\varphi_{\mathbf{x},\mathbf{d}}(t) = f(\mathbf{x} + t\mathbf{d})$ . Ci interessa in particolare la tomografia  $\varphi_{\mathbf{x}^i,\mathbf{d}^i}$ , poiché nell'algoritmo del gradiente stiamo proprio variando  $f$  a partire da  $\mathbf{x}^i$  lungo la direzione  $\mathbf{d}^i = -\nabla f(\mathbf{x}^i)$  di un passo  $\alpha$ : ovvero,

$$f(\mathbf{x}^i) = \varphi_{\mathbf{x},\mathbf{d}^i}(0), \quad f(\mathbf{x}^{i+1}) = \varphi_{\mathbf{x},\mathbf{d}^i}(\alpha).$$

Se  $f$  non varia molto velocemente, ci aspettiamo che anche  $\varphi$  non vari molto velocemente. Vale infatti la seguente proposizione:

**Proposizione 2.2.1.** *Se  $f$  è  $L$ -smooth,  $\varphi_{\mathbf{x},\mathbf{d}}(t)$  è  $L\|\mathbf{d}\|^2$ - smooth.*

*In particolare quindi  $\varphi_{\mathbf{x}^i,\mathbf{d}^i}(t)$  è  $L\|\nabla f(\mathbf{x}^i)\|^2$ - smooth.*

Per semplificare la notazione indichiamo con  $\varphi = \varphi_{\mathbf{x}^i,\mathbf{d}^i}$ . Dalla Proposizione 2.1.1 si ha che

$$\varphi'(0) = \frac{\partial f}{\partial \mathbf{d}^i}(\mathbf{x}^i) = \mathbf{d}^i \cdot \nabla f(\mathbf{x}^i) = -\|\nabla f(\mathbf{x}^i)\|^2.$$

Perciò per ogni  $\alpha > 0$  dal fatto che  $\varphi$  è  $L\|\nabla f(\mathbf{x}^i)\|^2$ - smooth, si ha che

$$\varphi'(\alpha) \leq \varphi'(0) + L\|\nabla f(\mathbf{x}^i)\|^2\alpha = \|\nabla f(\mathbf{x}^i)\|^2(L\alpha - 1)$$

Perciò sicuramente  $\varphi'(\alpha) < 0$  per ogni  $\alpha \in [0, 1/L)$ , e dunque nell'intervallo  $[0, 1/L]$   $\varphi$  è decrescente. Prendiamo dunque  $\alpha$  il più grande possibile, ovvero  $\alpha = \frac{1}{L}$ .

Abbiamo dunque che  $\varphi(\alpha) = f(\mathbf{x}^{i+1}) < f(\mathbf{x}^i) = \varphi(0)$ , ma ci chiediamo se possiamo in qualche modo stimare  $f(\mathbf{x}^i) - f(\mathbf{x}^{i+1})$ .

**Teorema 2.2.1.** *Per ogni  $\alpha \in [0, \frac{1}{L})$ ,*

$$\varphi(\alpha) \leq \varphi(0) + \|\nabla f(\mathbf{x})\|^2 \left( \frac{L\alpha^2}{2} - \alpha \right)$$

*Dimostrazione.* (non necessaria) Sia  $g : [0, \frac{1}{L}] \rightarrow \mathbb{R}$  la funzione così definita:

$$g(t) = \varphi'(0) + L\|\nabla f(\mathbf{x}^i)\|^2 t$$

Perciò per ogni  $t$ ,  $\varphi'(t) \leq g(t)$ . Per la monotonia dell'integrale ( e per il Teorema Fondamentale del Calcolo Integrale) si ha quindi che

$$\varphi(\alpha) - \varphi(0) = \int_0^\alpha \varphi'(t) dt \leq \int_0^\alpha g(t) dt. \quad (2.3)$$

Calcoliamo allora il secondo integrale, che è facile perché  $g$  è lineare in  $t$ :

$$\begin{aligned} \int_0^\alpha g(t) dt &= \left[ \varphi'(0)t + \frac{L\|\nabla f(\mathbf{x}^i)\|^2 t^2}{2} \right]_0^\alpha \\ &= \varphi'(0)\alpha + \frac{L\|\nabla f(\mathbf{x}^i)\|^2 \alpha^2}{2} \\ &= -\|\nabla f(\mathbf{x}^i)\|^2 \alpha + \|\nabla f(\mathbf{x}^i)\|^2 \frac{L\alpha^2}{2} \\ &= \|\nabla f(\mathbf{x}^i)\|^2 \left( \frac{L\alpha^2}{2} - \alpha \right). \end{aligned}$$

Mettendo l'ultima equazione e la disequazione (2.3) assieme arriviamo alla tesi.  $\square$

In particolare se  $\alpha = \frac{1}{L}$  si ha che

$$f(\mathbf{x}^{i+1}) = \varphi\left(\frac{1}{L}\right) \leq \varphi(0) - \|\nabla f(\mathbf{x}^i)\|^2 \frac{1}{2L} = f(\mathbf{x}^i) - \|\nabla f(\mathbf{x}^i)\|^2 \frac{1}{2L}$$

In conclusione, il metodo del gradiente con passo costante è sub-lineare su funzioni L-smooth, se “crediamo” che per ogni  $i$ ,  $f(\mathbf{x}^i)$  sia abbastanza vicino alla stima dall'alto appena dimostrata. In particolare si può dimostrare che per ottenere  $f(\mathbf{x}^i) \leq f(\mathbf{x}_*) + \varepsilon$  servono  $O(1/\varepsilon)$  passi. La buona notizia è che la stima non dipende dalla dimensione del problema, ossia dal numero di variabili di  $f$ . In alcuni casi particolari si può tuttavia dimostrare una maggior velocità dell'algoritmo: nel caso in cui la funzione sia anche fortemente convessa con modulo  $\tau$  si dimostra che l'algoritmo converge linearmente, con tasso di convergenza

$$r \leq 1 - \frac{\tau}{\lambda}. \quad (2.4)$$

### 2.2.2 Metodo del gradiente con line-search “esatta”

All’esatto opposto vogliamo vedere le proprietà del gradiente se scegliamo  $\alpha_i$  come il passo che minimizza  $\varphi$ . In altre parole in questo caso per ogni  $i$  vogliamo trovare

$$\mathbf{x}^{i+1} = \arg \min_{\alpha \in \mathbb{R}^+} \varphi(\alpha) = \arg \min_{\alpha \in \mathbb{R}^+} f(\mathbf{x}^i + \alpha \mathbf{d}^i)$$

Ad ogni passo dobbiamo perciò risolvere un problema di line-search (LS) cioè di ottimizzazione in una variabile. Poiché abbiamo visto che la line-search può essere molto difficile, analizziamo il caso particolari di funzioni quadratiche.

Prendiamo dunque  $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{q}^T \mathbf{x}$ , e chiamiamo  $\mathbf{g} = \nabla f(\mathbf{x}) = Q\mathbf{x} + \mathbf{q}$ . Per minimizzare  $\varphi(\alpha) = f(\mathbf{x} - \alpha \mathbf{g})$  dobbiamo trovare dove si annulla  $\varphi'$ :

$$\varphi'(\alpha) = \mathbf{g} \cdot \nabla f(\mathbf{x} - \alpha \mathbf{g}) = \mathbf{g} \cdot (Q\mathbf{x} - \alpha Q\mathbf{g} + \mathbf{q}) = \mathbf{g} \cdot (-\alpha Q\mathbf{g} + \mathbf{g})$$

che si annulla quindi se

$$\alpha = \frac{\|\mathbf{g}\|^2}{\mathbf{g}^T Q \mathbf{g}} \quad (2.5)$$

*Osservazione 5.* Il calcolo di  $\alpha$  richiede una moltiplicazione matrice per vettore, perciò costa  $O(n^2)$  per ogni iterazione. Notiamo che anche il calcolo di  $\nabla f(\mathbf{x})$  costa  $O(n^2)$  operazioni.

#### Velocità di convergenza del metodo

Vogliamo analizzare il tasso di convergenza del metodo per funzioni quadratiche: siano  $\lambda_1 \geq \dots \geq \lambda_n$  gli autovalori di  $Q$ . Notiamo che possiamo assumere che  $Q$  sia semi-definita positiva, poiché se esiste un autovalore negativo, il problema è illimitato inferiormente. Per dimostrare il risultato di convergenza ci serve in realtà che  $Q$  sia strettamente definita positiva, cioè  $\lambda_n > 0$ . In questo caso  $Q$  è anche invertibile, e gli autovalori di  $Q^{-1}$  sono  $1/\lambda_n \geq \dots \geq 1/\lambda_1$ .

**Lemma 2.2.1.** *Chiamiamo  $E(\mathbf{x}^i) = f(\mathbf{x}^i) - f(\mathbf{x}_*)$ . Allora la sequenza  $\{\mathbf{x}^i\}_i$  generata dal metodo del gradiente con  $\alpha_i$  scelto come in (2.5) soddisfa*

$$E(\mathbf{x}^{i+1}) = \left( 1 - \frac{\|\mathbf{d}^i\|^4}{((\mathbf{d}^i)^T Q \mathbf{d}^i) ((\mathbf{d}^i)^T Q^{-1} \mathbf{d}^i)} \right) E(\mathbf{x}^i) \quad (2.6)$$

Perciò, visto che per ogni  $\mathbf{x}$  vale  $\mathbf{x}^T Q \mathbf{x} \leq \lambda_1 \|\mathbf{x}\|^2$  e  $\mathbf{x}^T Q^{-1} \mathbf{x} \leq \|\mathbf{x}\|^2 / \lambda_n$  vale

$$\frac{\|\mathbf{x}\|^2}{\mathbf{x}^T Q \mathbf{x}} \geq \frac{1}{\lambda_1}, \quad \frac{\|\mathbf{x}\|^2}{\mathbf{x}^T Q^{-1} \mathbf{x}} \geq \lambda_n,$$

da cui prendendo  $\mathbf{x} = \mathbf{d}^i$  e sostituendo in (2.6), si ottiene

$$E(\mathbf{x}^{i+1}) \leq \left(1 - \frac{\lambda_n}{\lambda_1}\right) E(\mathbf{x}^i).$$

Il tasso  $r$  soddisfa perciò lo stesso bound che si trova nel caso di passo costante, ma in realtà usando la disuguaglianza di Kantorovich si può dimostrare una limitazione più forte:

$$r \leq \left(\frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}\right)^2.$$

Il metodo dunque converge con velocità lineare, ma tasso migliore di quello che abbiamo visto scegliendo un passo costante.

### Metodo nel caso di funzioni generali

Nel caso generale, abbiamo già visto nel capitolo 1 che una line-search esatta è spesso troppo costosa o addirittura impossibile, perciò in realtà possiamo usare solo una ricerca approssimativamente esatta.

### 2.2.3 Condizioni di Armijo e Wolfe

Il metodo del gradiente funziona persino con un passo costante, perciò a maggior ragione dovrebbe funzionare con una line search, anche se non esatta. Per farlo diamo delle condizioni su questa ricerca: vogliamo in innanzitutto che  $f^i = f(\mathbf{x}^i)$  decresca abbastanza:

**Definizione. (Condizione di Armijo)** Dato un valore  $0 < m_1 \ll 1$ , vogliamo che

$$\varphi(\alpha) \leq \varphi(0) + \alpha m_1 \varphi'(0) \tag{2.7}$$

Il problema è che se  $\alpha$  è molto vicino a 0, la condizione è soddisfatta ( $\varphi(\alpha) \approx \varphi(0) + \alpha \varphi'(0) > \varphi(0) + \alpha m_1 \varphi'(0)$ , vedasi la Figura (2.1)). Tuttavia se vale la condizione di Armijo e evitiamo che i passi siano troppo piccoli, allora abbiamo convergenza:

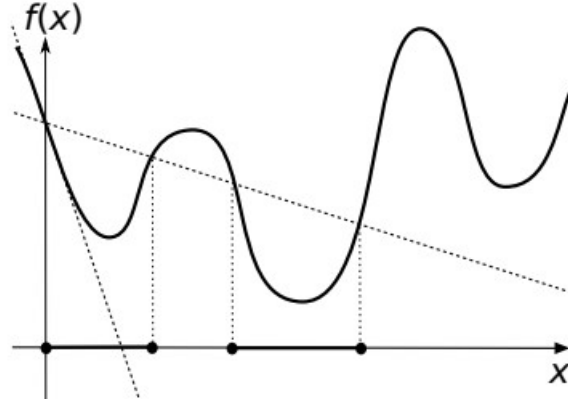


Figura 2.1: I segmenti più spessi sono le aree in cui vale la condizione di Armijo

**Proposizione 2.2.2.** *Se esiste un  $\bar{\alpha} > 0$  tale che per ogni  $i$   $\alpha_i < \bar{\alpha}$ , allora*

- *$f$  è inferiormente illimitata ( $f_* = -\infty$ )*
- *oppure  $\|\nabla f(\mathbf{x}^i)\| \rightarrow 0$ , ovvero  $\{\mathbf{x}^i\}$  converge ad un punto stazionario.*

*Dimostrazione.* Se vale la condizione (2.7) per ogni  $i$  e  $\|\nabla f(\mathbf{x}^i)\| > \epsilon > 0$  per ogni  $i$ <sup>3</sup>, vale allora

$$f^{i+1} = f^i + m_1 \alpha_i \varphi'_i(0) \leq f^i - m_1 \bar{\alpha} \epsilon$$

e per induzione si ha che

$$f^i \leq f^0 - \bar{\alpha} i \epsilon \rightarrow -\infty$$

□

Un modo per evitare di scegliere  $\alpha$  troppo piccolo è imporre anche la cosiddetta condizione di Goldstein: dato  $m_2 > m_1$ , imponiamo

$$\varphi(\alpha) \geq \varphi(0) + \alpha m_2 \varphi'(0)$$

Il problema di questa condizione è che insieme alla condizione di Armijo può escludere molti/ tutti i minimi locali (Figura 2.2)

<sup>3</sup>In realtà sarebbe più corretto dire per infiniti  $i$ , visto che stiamo negando che  $\nabla f(\mathbf{x}^i) \rightarrow 0$ , ma la dimostrazione funziona ugualmente a meno di ridursi ad una sottosuccessione  $\{i_j\}_{j=0}^\infty$

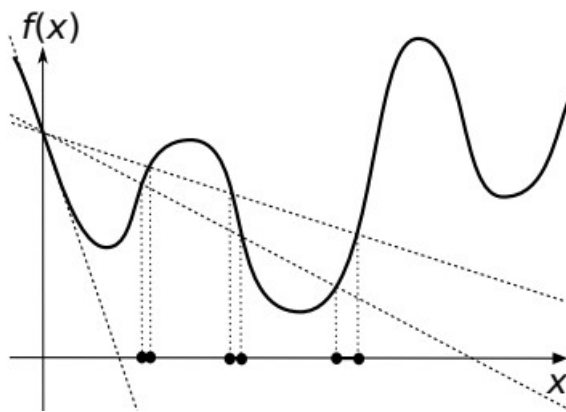


Figura 2.2: I brevissimi segmenti tra le coppie di punti sono le aree in cui valgono le condizioni di Armijo e di Goldstein. Si può notare come in questo caso tutti i minimi siano esclusi.

Introduciamo perciò una terza condizione che non esclude di per sé nessun punto stazionario:

**Definizione. (Condizione di Wolfe)** Dato  $m_1 < m_3 < 1$ , vogliamo che sia soddisfatta la condizione

$$\varphi'(\alpha) \geq m_3 \varphi'(0) \quad (2.8)$$

Vogliamo cioè che la derivata lungo la direzione di ricerca di  $f$  sia più vicina allo 0 dopo ogni passo, oppure sia positiva (anche molto più grande di 0). Per evitare che  $\varphi'(\alpha) \gg 0$  possiamo rafforzare la condizione in

$$|\varphi'(\alpha)| \geq m_3 |\varphi'(0)|. \quad (2.9)$$

Quest'ultima condizione è detta di Wolfe forte.

### Line Search di Armijo e Wolfe

Le condizioni di Armijo e Wolfe combinate forniscono un modo ragionevole di scegliere il passo per ogni iterazione, senza rischiare di tagliare troppi minimi locali: i minimi locali potrebbero essere tagliati fuori solo dalla condizione di Armijo, ma se  $m_1$  non è troppo vicino a 1 ciò non succede. Nella pratica si sceglie infatti  $m_1$  pari a circa  $10^{-4}$ .



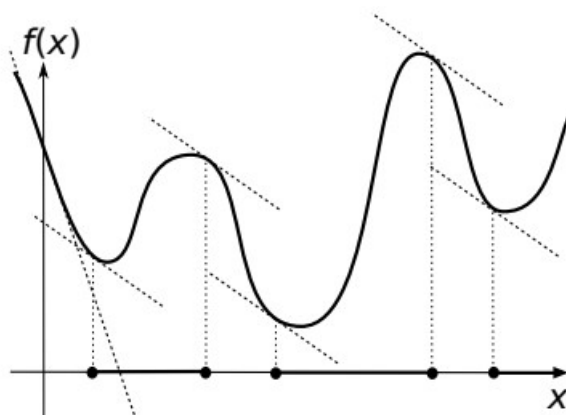


Figura 2.3: I segmenti in grassetto sono le aree in cui vale la condizione di Wolfe

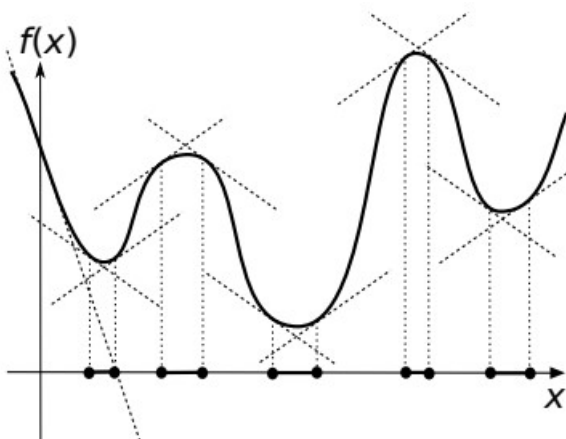


Figura 2.4: I segmenti in grassetto sono le aree in cui vale la condizione di Wolfe forte

**Proposizione 2.2.3.** *Se  $\varphi$  è derivabile con continuità e  $\varphi(\alpha)$  è inferiormente limitato (non va a  $-\infty$  se  $\alpha \rightarrow +\infty$ ), allora esiste un  $\alpha$  che soddisfa le condizioni di Armijo e di Wolfe.*

*Dimostrazione.* Definiamo la funzione  $g(\alpha) = \varphi(0) + m_1\alpha\varphi'(0)$ .

Definiamo  $d(\alpha) = g(\alpha) - \varphi(\alpha)$ . Questa funzione è la distanza (con segno) tra  $g$  e  $\varphi$  in  $\alpha$ . Osserviamo che  $d(0) = 0$  e

$$d'(\alpha) = m_1\varphi'(0) - \varphi'(\alpha),$$

perciò in particolare

$$d'(0) = (m_1 - 1)\varphi'(0) > 0$$

Se non esistesse  $\bar{\alpha} > 0$  tale che  $d(\bar{\alpha}) = 0$ ,  $\varphi$  sarebbe illimitata interiormente: sarebbe infatti per ogni  $\alpha$ ,  $d(\alpha) > 0$  e quindi  $g(\alpha) > \varphi(\alpha)$ . Ma  $g(\alpha) \xrightarrow{\alpha \rightarrow \infty} -\infty$ , perciò anche  $\varphi(\alpha) \xrightarrow{\alpha \rightarrow \infty} -\infty$ .

Esiste perciò  $\bar{\alpha}$  tale che  $d(\bar{\alpha}) = 0$ . Prendiamo in particolare il più piccolo.

Vediamo ora che vale la condizione di Armijo per ogni  $\alpha < \bar{\alpha}$ : infatti  $\bar{\alpha}$  è il più piccolo  $\alpha > 0$  per cui  $d(\alpha) = 0$ , per cui per ogni  $\alpha < \bar{\alpha}$ , abbiamo che

$$d(\alpha) > 0,$$

ovvero vale la condizione di Armijo.

Per il teorema di Rolle abbiamo inoltre che esiste  $0 < \alpha' < \bar{\alpha}$  tale per cui  $d'(\alpha') = 0$ , ovvero

$$m_1\varphi'(0) = \varphi'(\alpha').$$

Poiché  $m_3 < m_1$ ,

$$\varphi'(\alpha') > m_3\varphi'(0),$$

cioè  $\alpha'$  soddisfa anche la condizione di Wolfe. □

Dato che  $\alpha$  esiste, tutto sta nel trovarlo: per farlo basta usare un qualsiasi metodo di ottimizzazione in una variabile, usando come nuova condizione di stop che valgano (2.7) e (2.8), oppure (2.7) e (2.9) se vogliamo la versione forte.

## Backtracking Line Search

Una versione ancora più semplice, che controlla solo se è soddisfatta la condizione di Armijo, è la Backtracking Line Search (BLS). Per ogni iterazione della discesa del gradiente ripetiamo la seguente procedura:

1. Prendiamo un passo  $\alpha$  iniziale e un numero reale  $0 < t < 1$
2. Se vale  $\varphi(\alpha) \leq \varphi(0) + m_1\alpha\varphi'(\alpha)$  finiamo l'iterazione, altrimenti
3. Ripetiamo con un nuovo  $\alpha' = t\alpha$

Ripetendo 2. e 3. finché non è soddisfatta la (2.7), garantiamo di trovare un  $\alpha$  che soddisfi la condizione: infatti la successione  $\{t^k\alpha\}$  tende a 0, e per  $\alpha$  che tende a 0 la condizione di Armijo è sempre soddisfatta. Evitiamo inoltre “Cariddi” cioè di muoverci troppo poco, perché nei primi passi di ogni iterazione possiamo provare un  $\alpha$  anche molto grande

---

**Algoritmo 2** Backtracking Line Search

---

<b>procedure</b> $\alpha = \text{BLS}(\varphi, \alpha, m_1, \tau)$ $// \tau < 1$ <b>while</b> $(\varphi(\alpha) > \varphi(0) + m_1\alpha\varphi'(0))$ <b>do</b> $\alpha \leftarrow \tau\alpha;$
--

---

In che condizioni possiamo garantire che la BLS funzioni? Ricordiamo che per ogni  $i$  esiste un  $\alpha_i$  tale che per ogni  $\alpha < \alpha_i$ , la condizione di Armijo è soddisfatta all'iterazione  $i$  della discesa del gradiente.

Assumiamo per semplicità che per ogni iterazione  $\alpha = 1$  (intendiamo l' $\alpha$  iniziale dato in input). Allora sicuramente il passo all'iterazione  $i$  sarà  $t^{k_i} < 1$ , con  $k_i > \min\{k \mid t^k < \alpha_i\}$ . Se garantiamo che esista un  $\bar{\alpha}$  tale per cui  $\alpha_i > \bar{\alpha}$  per ogni  $i$ , allora si ha anche che  $k_i > \bar{k} = \min\{k \mid t^k < \bar{\alpha}\}$ . Dunque per la Proposizione 2.2.2, si ha o la convergenza del metodo ad un punto stazionario o che  $f$  è illimitata inferiormente.

Per garantire che un tale  $\bar{\alpha}$  esista, possiamo assumere che  $f$  sia L-smooth: in questo caso ricordiamo che  $\varphi$  è  $(L\|\nabla f(\mathbf{x})\|^2)$ -smooth. Ricordiamo anche che esiste un  $\alpha' < \bar{\alpha}$  che soddisfa anche la condizione di Wolfe (visto nella dimostrazione della Proposizione 2.2.2). Dunque vale

$$L\|\nabla f(\mathbf{x})\|^2\alpha' \geq \varphi'(\alpha') - \varphi'(0) > (1 - m_3)(-\varphi'(0)) = (1 - m_3)\|\nabla f(\mathbf{x})\|^2$$

Semplificando si ottiene allora che

$$\alpha' > \frac{1 - m_3}{L}.$$

Perciò  $\bar{\alpha} > \alpha'$  esiste ed è maggiore di 0.

Se  $f$  è anche  $\tau$ -convessa abbiamo inoltre assicurato un ordine di convergenza lineare con  $r \approx \frac{1 - \tau}{L}$  dipendente da  $m_1$  e  $m_3$ .

## 2.3 Metodo di Newton in più variabili

Abbiamo finora assunto che la direzione  $\mathbf{d}^i$  sia l'antigradiente in  $\mathbf{x}^i$ . Tuttavia esistono infinite direzioni di decrescita in  $\mathbf{x}^i$ . Abbiamo usato l'antigradiente per due motivi:

1. Lontano da un punto stazionario  $\varphi'_i(0) = -\|\nabla f(\mathbf{x}^i)\|^2$  è molto negativa  $\Rightarrow$  la decrescita è veloce.
2. Se i passi  $\alpha_i$  non tendono a 0 troppo velocemente, abbiamo una decrescita significativa ad ogni passo, a meno che  $\|\nabla f(\mathbf{x}^i)\| \rightarrow 0$  (che è comunque ciò che vogliamo, stiamo cercando un punto stazionario).
2. Non dipende sostanzialmente dalla direzione scelta, e ci sono molte altre direzioni in cui vale 1. a meno di un fattore moltiplicativo.

*Esempio.* Se  $\mathbf{d}^i = R(-\nabla f(\mathbf{x}^i))$  con  $R$  una matrice di rotazione di  $\frac{\pi}{4}$ ,

$$\varphi'_i(0) = -\|\nabla f(\mathbf{x}^i)\|^2 \cos\left(\frac{\pi}{4}\right) = -\|\nabla f(\mathbf{x}^i)\|^2 \frac{\sqrt{2}}{2}$$

e la dimostrazione di convergenza funziona allo stesso modo.

In effetti ogni direzione tale per cui  $\nabla f(\mathbf{x}^i) \cdot \mathbf{d}^i < 0$  è di decrescita. abbiamo cioè che esiste un intero semispazio di direzioni di decrescita. Dobbiamo accertarci tuttavia che  $\nabla f(\mathbf{x}^i) \cdot \mathbf{d}^i$  non converga a 0 troppo velocemente:

**Teorema 2.3.1** (Zoutendijk). *Sia  $f \in C^1$ ,  $L$ -smooth e con  $f_* > -\infty$ , se  $\{\mathbf{d}^i\}$  sono le direzioni di decrescita con  $\varphi'_i(0) = -\|\nabla f(\mathbf{x}^i)\|^2 \cos \theta^i$ , allora se valgono le condizioni di Armijo e Wolfe vale anche*

$$\sum_{i=0}^{\infty} \cos^2 \theta^i \|\nabla f(\mathbf{x}^i)\|^2 < \infty$$

Come conseguenza abbiamo che se  $\sum_{i=0}^{\infty} \cos^2 \theta^i \rightarrow \infty$ , allora per forza  $\|\nabla f(\mathbf{x}^i)\| \rightarrow 0$ , quindi se  $\mathbf{d}^i$  non diventa ortogonale a  $\nabla f(\mathbf{x}^i)$  troppo velocemente, c'è sempre convergenza.

Ci chiediamo perciò come scegliere delle direzioni  $\mathbf{d}^i$  migliori dell'antigradiente. Per farlo dobbiamo guardare al modello del secondo ordine.

### 2.3.1 Newton per funzioni convesse

Usiamo il modello del secondo ordine

$$T_{\mathbf{x}^0}^{(2)}(\mathbf{x}) = f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^0)^T \nabla^2 f(\mathbf{x}^0)(\mathbf{x} - \mathbf{x}^0)$$

Nel caso localmente convesso il modello è definito positivo, perciò esiste un minimo: tale minimo è in  $\bar{\mathbf{x}} = \mathbf{x}^0 + \mathbf{d}$  con

$$\mathbf{d} = -(\nabla^2 f(\mathbf{x}^0))^{-1} \nabla f(\mathbf{x}^0)$$

Il metodo di Newton non è globalmente convergente, esattamente come in una variabile, perciò va globalizzato: per farlo cominciamo con una LS Backtracking o con le condizioni di Armijo-Wolfe con  $\alpha = 1$  iniziale. Per dimostrare che un algoritmo del genere funziona abbiamo 3 Teoremi da dimostrare:

**Teorema 2.3.2.** *Sia  $f \in C^2$  e  $L$ -smooth e  $\tau$ -convessa. Chiamiamo  $\theta_i$  l'angolo tra  $\mathbf{d}^i = -(\nabla^2 f(\mathbf{x}^i))^{-1} \nabla f(\mathbf{x}^i)$  e  $-\nabla f(\mathbf{x}^i)$ . Si ha allora che*

$$\cos \theta^i \leq -\frac{\tau}{L}$$

**Corollario 2.3.1.** *Grazie al Teorema di Zoutendijk, il teorema 2.3.2 ci garantisce che il metodo di Newton con le condizioni di Armijo-Wolfe o con BLS converga.*

*Dimostrazione.* Si ha che  $\nabla f(\mathbf{x}^i)^T \mathbf{d}^i = -\nabla f(\mathbf{x}^i)^T \nabla f(\mathbf{x}^i)$ , perciò

$$\nabla f(\mathbf{x}^i)^T \mathbf{d}^i = (\mathbf{d}^i)^T (\nabla^2 f(\mathbf{x}^i)) \mathbf{d}^i \leq -\tau \|\mathbf{d}^i\|^2$$

e

$$\|\nabla f(\mathbf{x}^i)\| = \|\nabla f(\mathbf{x}^i)^T \mathbf{d}^i\| \leq \|\nabla f(\mathbf{x}^i)^2\| \|\mathbf{d}^i\| \leq L \|\mathbf{d}^i\|.$$

Dunque

$$\cos \theta^i = \frac{\|\nabla f(\mathbf{x}^i)^T \mathbf{d}^i\|}{\|\nabla f(\mathbf{x}^i)\| \|\mathbf{d}^i\|} \leq -\frac{\tau}{L}$$

□

**Teorema 2.3.3.** *Se  $f \in C^3$ , e esiste un punto  $\mathbf{x}_*$  tale che  $\nabla f(\mathbf{x}_*) = 0$  e  $\nabla^2 f(\mathbf{x}_*)$  è definita positiva, allora esiste un  $\delta > 0$  tale che iniziando il metodo di Newton da  $\mathbf{x}^0 \in \mathcal{B}(\mathbf{x}_*, \delta)$ , il metodo converge con tasso di convergenza quadratico*

*Dimostrazione.* Lasciata al lettore, è del tutto simile al caso unidimensionale (Teorema 1.2.2)  $\square$

**Teorema 2.3.4.** *Sia  $f$  come nel Teorema 2.3.2. Se  $\mathbf{x}^i \rightarrow \mathbf{x}_*$ , allora esiste un  $h \in \mathbb{N}$  tale che, per ogni  $i \geq h$ , la condizione di Armijo (2.7) è sempre soddisfatta per  $\alpha^i = 1$  (richiedendo  $m_1 \leq \frac{1}{2}$ :  $m_1 > \frac{1}{2}$  taglia via il minimo se  $f$  è quadratica)*

*Dimostrazione.*  $\mathbf{x}^i \rightarrow \mathbf{x}_*$  implica che  $\|\nabla f(\mathbf{x}^i)\| \rightarrow 0$  e quindi anche  $\|\mathbf{d}^i\| \rightarrow 0$ . Perciò per la formula di Taylor al secondo ordine

$$f(\mathbf{x}^{i+1}) = f(\mathbf{x}^i + \mathbf{d}^i) = f(\mathbf{x}^i) + \nabla f(\mathbf{x}^i)^T \mathbf{d}^i + \frac{1}{2} (\mathbf{d}^i)^T \nabla^2 f(\mathbf{x}^i) \mathbf{d}^i + R(\mathbf{d}^i)$$

$\mathbf{d}^i = -\nabla^2 f(\mathbf{x}^i)^{-1} \nabla f(\mathbf{x}^i)$  perciò

$$\begin{aligned} f(\mathbf{x}^{i+1}) &= f(\mathbf{x}^i) - \nabla f(\mathbf{x}^i)^T \nabla^2 f(\mathbf{x}^i)^{-1} \nabla f(\mathbf{x}^i) \\ &\quad + \frac{1}{2} (-\nabla^2 f(\mathbf{x}^i)^{-1} \nabla f(\mathbf{x}^i))^T \nabla^2 f(\mathbf{x}^i) (-\nabla^2 f(\mathbf{x}^i)^{-1} \nabla f(\mathbf{x}^i)) \\ &\quad + R(\mathbf{d}^i) \\ &= f(\mathbf{x}^i) - \nabla f(\mathbf{x}^i)^T \nabla^2 f(\mathbf{x}^i)^{-1} \nabla f(\mathbf{x}^i) \\ &\quad + \frac{1}{2} \nabla f(\mathbf{x}^i)^T \nabla^2 f(\mathbf{x}^i)^{-1} \nabla f(\mathbf{x}^i) + R(\mathbf{d}^i) \\ &= f(\mathbf{x}^i) + \frac{1}{2} \nabla f(\mathbf{x}^i)^T \mathbf{d}^i + R(\mathbf{d}^i). \end{aligned}$$

Abbiamo inoltre assunto che  $f$  sia  $\tau$ -convessa, perciò

$$-\nabla f(\mathbf{x}^i)^T \mathbf{d}^i = (\mathbf{d}^i)^T \nabla^2 f(\mathbf{x}^i) \mathbf{d}^i \geq \tau \|\mathbf{d}^i\|^2,$$

Inoltre  $R(\mathbf{d}^i) = o(\|\mathbf{d}^i\|)$ , dunque dato  $m_1 < \frac{1}{2}$  esiste un  $\bar{i}$  tale per cui, per ogni  $i > \bar{i}$ ,

$$f(\mathbf{x}^{i+1}) \leq f(\mathbf{x}^i) + m_1 \nabla f(\mathbf{x}^i)^T \mathbf{d}^i,$$

cioè vale la condizione di Armijo con  $\alpha^i = 1$ .  $\square$

### Metodo di Newton come metodo del gradiente in uno spazio modificato

Una possibile interpretazione del metodo di Newton è quella che vede l'algoritmo come un metodo del gradiente in uno spazio modificato. Prendiamo

per esempio una funzione quadratica  $f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} + \mathbf{q} \mathbf{x}$ . In questo caso il modello del secondo ordine è  $f$  stesso, perciò compiendo un passo dell'algoritmo di Newton ci troviamo subito in  $\mathbf{x}_* = -Q^{-1} \mathbf{q}$ , con  $\nabla f(\mathbf{x}_*) = 0$ . Quindi  $\mathbf{x}_*$  è un punto stazionario e l'algoritmo impiega un solo passo prima di terminare. Abbiamo visto che l'algoritmo del gradiente è molto efficiente per funzioni quadratiche in cui l'autovalore più grande e quello più piccolo di  $Q$ , rispettivamente  $\lambda_1(Q)$ , e  $\lambda_n(Q)$ , sono molto vicini ( $Q$  è ben condizionata).

Il metodo di Newton applicato ad una funzione quadratica corrisponde ad un metodo del gradiente su un'altra funzione quadratica

$$h(\mathbf{z}) = \mathbf{z}^t Q' \mathbf{z} + q^T \mathbf{z}$$

in cui la matrice  $Q'$  è molto ben condizionata:

**Definizione.** Definiamo la “radice quadrata” di  $Q$  come la matrice  $R$  tale che  $R^2 = Q$ . In generale tale matrice può non esistere (come matrice reale) o non essere unica.

Se  $Q$  è semidefinita positiva possiamo calcolare la decomposizione spettrale  $Q = H \Lambda H^T$  e prendere  $R = H \sqrt{\Lambda} H^T$ .

Se inoltre  $Q$  è non singolare, anche  $R$  è non singolare perciò esiste un'inversa  $R^{-1}$  e le funzioni

$$\mathbf{x} \rightarrow \mathbf{z} = R \mathbf{x} \quad \mathbf{z} \rightarrow \mathbf{x} = R^{-1} \mathbf{z}$$

sono bigezioni (più precisamente isomorfismi lineari) dello spazio  $\mathbb{R}^n$ . In questo modo possiamo definire

$$h(\mathbf{z}) = f(R^{-1} \mathbf{z}) = \frac{1}{2} \mathbf{z}^T I \mathbf{z} + \mathbf{q} R^{-1} \mathbf{z}.$$

notiamo che  $\nabla^2 h(\mathbf{z}) = I$ , dunque “nello spazio  $\mathbf{z}$  l'hessiana di  $f$  è l'identità”. In questo spazio il metodo del gradiente per minimizzare  $h$  è molto efficiente: detto  $\mathbf{g} = \nabla h(\mathbf{z}) = -\mathbf{z} - R^{-1} \mathbf{q}$ , si ha

$$\nabla h(\mathbf{z} + \mathbf{g}) = \nabla h(-R^{-1} \mathbf{q}) = -R^{-1} \mathbf{q} + R^{-1} \mathbf{q} = 0.$$

Lo svantaggio di questa trattazione è che calcolare  $R$  è molto costoso, tuttavia  $\mathbf{z} = R \mathbf{x}$  non è l'unica scelta, soprattutto se  $Q$  non è definita positiva.

### 2.3.2 Newton per funzioni non convesse

#### Modificare l'hessiana

Abbandoniamo ora l'ipotesi che ci ha accompagnato nelle sezioni precedenti, la  $(\tau-)$ convessità di  $f$ , e vediamo come garantire comunque la convergenza del metodo di Newton. Abbiamo visto che il metodo di Newton corrisponde ad una deformazione dello spazio che rende  $\nabla^2 f(\mathbf{x})$  meglio condizionata, e tale dilatazione è appunto prodotta dalla moltiplicazione per  $-\nabla^2(\mathbf{x})^{-1}$ . Specialmente se  $\nabla^2 f(\mathbf{x})$  non è definita positiva, tale scelta può non essere la migliore. In generale si può pensare di prendere per ogni  $i$ ,  $\mathbf{d}^i = -H_i \nabla f(\mathbf{x}^i)$ , per qualche matrice  $H_i$ . I metodi in cui l'iterazione è eseguita in questo modo vengono detti metodi *quasi-Newton*.

**Teorema 2.3.5.** *Se scegliamo ad ogni passo una matrice  $H_i$  tale che gli autovalori di  $H_i$  sono contenuti in  $[\tau, L]$  per due valori  $\tau < L$  (cioè  $\tau I \preceq H_i \preceq L I$ ), e prendiamo  $\mathbf{d}^i = H_i^{-1} \nabla f(\mathbf{x}^i)$ , se valgono le condizioni di Armijo-Wolfe si ha convergenza globale.*

*Dimostrazione.* È uguale al Teorema 2.3.2 con  $H_i$  al posto di  $\nabla^2 f(\mathbf{x}^i)$ . Avevamo infatti usato le ipotesi di  $L$ -smoothness e  $\tau$ -convessità per garantire che  $\tau \preceq \nabla^2 f(\mathbf{x}^i) \preceq L$ .  $\square$

Come possiamo trovare  $H_i$  come nel teorema precedente? Se  $\nabla^2 f(\mathbf{x}^i)$  non è semi-definita positiva, esistono degli autovalori minori di 0 di  $\nabla^2 f(\mathbf{x}^i)$ .

In teoria se  $\lambda_1 > \dots > \lambda_n$ , e  $\lambda_n < 0$  è il più piccolo di essi, basta prendere  $H_i = \nabla^2 f(\mathbf{x}^i) + \epsilon_i I$  con  $\epsilon_i > -\lambda_n$  per garantire che  $H_i$  sia definita positiva. Tuttavia incontriamo due problemi:

**Problema numerico:** In precisione di macchina valori troppo piccoli ( $\approx 10^{-16}$  ma anche più grandi) potrebbero essere considerati 0.

**Problema algoritmico:** Se  $\lambda_n + \epsilon_i$  è troppo piccolo, si ha come nella discussione sull'efficienza del metodo del gradiente, linee di livello molto allungate (nel nuovo spazio) e il metodo è poco efficiente.

**Soluzione:** Prendiamo  $\epsilon_i = \max\{0, \delta - \lambda_n\}$  con  $\delta > 0$  un parametro piccolo fissato. In questo modo se già  $\nabla^2 f(\mathbf{x}^i)$  ha tutti autovalori maggiori di  $\delta$ , si fa un passo normale del metodo di Newton, in caso contrario modifichiamo l'hessiana come sopra.



Si dimostra che questa scelta di  $H_i$  risolve il problema di ottimizzazione

$$\min \{ \|H_i - \nabla^2 f(\mathbf{x}^i)\|_2 : H_i \succeq \delta \}.$$

Possiamo anche puntare a minimizzare una norma di matrice diversa da  $\|\cdot\|_2$ : ad esempio usando la norma di Frobenius e risolvendo

$$\min \{ \|H_i - \nabla^2 f(\mathbf{x}^i)\|_F : H_i \succeq \delta \}$$

si ottiene che  $H_i = H\bar{\Lambda}H^T$  con

$$\bar{\Lambda} = \begin{pmatrix} \bar{\lambda}_1 & & \\ & \ddots & \\ & & \bar{\lambda}_n \end{pmatrix}$$

e  $\bar{\lambda}_i = \max\{\lambda_i, \delta\}$ . In ogni caso per trovare  $H_i$  appropriato abbiamo bisogno di  $O(n^3)$  operazioni, che sono troppe per ottimizzazione per larga scala, ad esempio  $n \geq 10^4$ .

## Regioni di fiducia

Round balls are better than  
kinky balls

---

Frangioni ti fa cock-rating

Un'altra tecnica per far funzionare il metodo di Newton senza che  $\nabla^2 f(\mathbf{x}^i)$  sia definita positiva è cercare delle regioni di fiducia. Se l'Hessiana non è semi-definita positiva, il modello del secondo ordine non ha minimo in  $\mathbb{R}^n$ , perchè ci sono delle direzioni di curvatura negativa. Invece che escludere tali regioni come abbiamo fatto nella sezione precedente, vogliamo sfruttare tali direzioni di decrescita. Minimizziamo allora il modello del secondo ordine attorno a  $\mathbf{x}^i$ ,  $T_{\mathbf{x}^i}^{(2)}(\mathbf{y})$ , in un insieme *compatto*  $\mathcal{T}_i$  detto *regione di fiducia* attorno a  $\mathbf{x}^i$ . Cerchiamo cioè

$$\mathbf{x}^{i+1} \in \arg \min_{\mathbf{y} \in \mathcal{T}_i} T_{\mathbf{x}^i}^{(2)}(\mathbf{y})$$

Il problema diventa un problema di ottimizzazione vincolato che è NP-hard per molte scelte di  $\mathcal{T}_i$ , ad esempio prendendo le palle (chiuse) in norma 1 o in norma  $\infty$ . Se però  $\mathcal{T}_i = \bar{\mathcal{B}}_2(\mathbf{x}^i, r)$  è la palla chiusa in norma 2, tale problema è risolvibile. Abbiamo incontrato uno dei primi casi di ottimizzazione vincolata, che studieremo nel prossimo capitolo.

## 2.4 Metodi quasi-Newton

Vediamo come ricavare dei metodi in qualche modo simili a quello di Newton senza calcolare però l'Hessiana. Tale calcolo è infatti molto dispendioso se  $n$  è molto grande. Prendiamo perciò un metodo del tipo

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i \mathbf{d}^i, \quad \mathbf{d}^i = -H^i \nabla f(\mathbf{x}^i). \quad (2.10)$$

Lo spazio delle  $H_i$  per cui si ha convergenza è grande:

**Teorema 2.4.1.** *Se  $f \in C^2$  e*

$$\lim_{i \rightarrow \infty} \frac{\| (H_i - \nabla^2 f(\mathbf{x}^i)) \mathbf{d}^i \|}{\| \mathbf{d}^i \|} = 0,$$

*si ha convergenza superlineare del metodo*

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{d}^i, \quad \mathbf{d}^i = -H_i^{-1} \nabla f(\mathbf{x}^i).$$

Con un metodo quasi-Newton, per ogni  $i$  stiamo implicitamente costruendo un modello

$$m_i(\mathbf{x}) = \nabla f(\mathbf{x}^i)^T (\mathbf{x} - \mathbf{x}^i) + (\mathbf{x} - \mathbf{x}^i)^T H_i (\mathbf{x} - \mathbf{x}^i)$$

Alcune proprietà di  $m_i$  e  $H_i$  che vorremmo sono:

1. Per ogni  $i$ ,  $H_i$  è definita positiva (i modelli sono fortemente convessi);
2.  $\nabla m_{i+1}(\mathbf{x}^i) = \nabla f(\mathbf{x}^i)$  (il nuovo modello conviene con i precedenti sulle vecchie informazioni: infatti anche  $\nabla m_i(\mathbf{x}^i) = \nabla f(\mathbf{x}^i)$ );
3.  $\|H_{i+1} - H_i\|$  è piccolo (il nuovo modello è simile a quello precedente).

La proprietà 2. è equivalente a

$$2'. \quad H_{i+1}(\mathbf{x}^{i+1} - \mathbf{x}^i) = \nabla f(\mathbf{x}^{i+1}) - \nabla f(\mathbf{x}^i) \quad (\text{equazione della secante})$$

**Notazione:** denotiamo con  $\mathbf{s}^i = \mathbf{x}^{i+1} - \mathbf{x}^i = \alpha_i \mathbf{d}^i$ ,  $\mathbf{y}^i = \nabla f(\mathbf{x}^{i+1}) - \nabla f(\mathbf{x}^i)$ . Si nota immediatamente che con questa nuova notazione, l'equazione delle secanti 2'. si può riscrivere come

$$H_{i+1} \mathbf{s}^i = \mathbf{y}^i \quad (2.11)$$

L'equazione 2.11 implica che  $\mathbf{s}^i \cdot \mathbf{y}^i = (\mathbf{s}^i)^T H_{i+1} \mathbf{s}^i$ . Se vale anche la proprietà 1. vale dunque  $\mathbf{s}^i \cdot \mathbf{y}^i > 0$ . Tale proprietà è detta *condizione di curvatura* ed è spesso scritta come

$$\rho_i = \frac{1}{\mathbf{s}^i \cdot \mathbf{y}^i} > 0 \quad (2.12)$$

Vogliamo perciò garantire che la condizione sulla curvatura (2.12) sia soddisfatta.

**Proposizione 2.4.1.** *La condizione di Wolfe (2.8) implica la condizione sulla curvatura.*

*Dimostrazione.*

$$\varphi'(\alpha_i) = \nabla f(\mathbf{x}^{i+1}) \cdot \mathbf{d}^i \geq m_3 \varphi'(0) = m_3 (\nabla f(\mathbf{x}^i) \cdot \mathbf{d}^i)$$

, dunque

$$(\nabla f(\mathbf{x}^{i+1}) - \nabla f(\mathbf{x}^i)) \cdot \mathbf{d}^i = \mathbf{y}^i \cdot \mathbf{d}^i = (m_3 - 1) \varphi'(0) > 0.$$

Ma  $\mathbf{s}^i = \alpha_i \mathbf{d}^i$  perciò

$$\mathbf{y}^i \cdot \mathbf{s}^i = \alpha (m_3 - 1) \varphi'(0) > 0$$

□

Dunque con una Armijo-Wolfe Line Search la condizione (2.12) è sempre soddisfatta.

### 2.4.1 DFP

Come detto nella proprietà 3. , vorremmo che  $H_{i+1} - H_i$  sia piccolo. Una soluzione è prendere

$$H_{i+1} = \arg \min \{ \|H - H_i\| : H \succeq 0, H \text{ soddisfa 2'.} \} \quad (2.13)$$

per una qualche norma a nostra scelta.

**Definizione.** <sup>4</sup> Data una matrice  $W$ , la norma di Frobenius pesata è  $\|A\|_W = \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|$ .

---

<sup>4</sup>Questa definizione e le seguenti sono solo per dare un'idea da dove vengano fuori le formule DFP e BFGS. Non sono necessarie ai fini dell'implementazione dell'algoritmo

Possiamo scegliere qualsiasi  $W$  tale che vale  $W\mathbf{y}^i = \mathbf{s}^i$ , e in particolare possiamo prendere, per ogni  $i$ ,  $W = \bar{G}_i^{-1}$ , dove  $\bar{G}_i$  è l'Hessiana media, definita da

$$\bar{G}_i = \int_0^1 \nabla^2 f(\mathbf{x}^i + t\alpha_i \mathbf{d}^i) dt$$

Scegliendo questa norma e questa matrice,  $H_{i+1}$  è facilmente calcolabile attraverso la formula di Davidon-Fletcher-Powell:

**Proposizione 2.4.2.** *La soluzione del problema (2.13) con la norma di Frobenius indotta da  $\bar{G}^{-1}$  è*

$$(DFP-H) \quad H_{i+1} = (I - \rho_i \mathbf{y}^i (\mathbf{s}^i)^T) H_i (I - \rho_i \mathbf{s}^i (\mathbf{y}^i)^T) - \rho_i \mathbf{y}^i (\mathbf{y}^i)^T \quad (2.14)$$

Per applicare il metodo non ci interessa in realtà  $H_i$ , ma la sua inversa  $B_i = H_i^{-1}$ .  $B_i$  si può calcolare altrettanto facilmente come segue:

$$(DFP-B) \quad B_{i+1} = B_i - \rho_i \mathbf{s}^i (\mathbf{s}^i)^T - \frac{(B_i \mathbf{y}^i (\mathbf{y}^i)^T B_i)}{(\mathbf{y}^i)^T B_i \mathbf{y}^i} \quad (2.15)$$

Ad ogni passo quindi correggiamo  $B_i$  sommandogli due matrici di rango 1. Calcolare tali matrici è poco costoso: il costo è dominato dallo svolgimento di 3 prodotti matrice-vettore, perciò il metodo DFP richiede  $O(n^2)$  operazioni per iterazione.

## 2.4.2 BFGS

Invece che risolvere il problema (2.13), possiamo risolvere il problema equivalente per  $B_i$ , che si ottiene scambiando i ruoli di  $\mathbf{s}^i$  e  $\mathbf{y}^i$ :

$$B'_{i+1} = \arg \min \{ \|B - B_i\| : B \succeq 0, B\mathbf{y}^i = \mathbf{s}^i \} \quad (2.16)$$

Usando la norma definita in precedenza con  $\bar{G}_i$  (senza l'inversa), si ottengono in modo simmetrico scambiando  $H \leftrightarrow B$ ,  $\mathbf{y} \leftrightarrow \mathbf{s}$ , le formule di Broyden-Fletcher-Goldfarb-Shanno:

$$(BFGS-H) \quad H'_{i+1} = H'_i - \rho_i \mathbf{y}^i (\mathbf{y}^i)^T - \frac{H_i \mathbf{s}^i (\mathbf{s}^i)^T H'_i}{(\mathbf{s}^i)^T H_i \mathbf{s}^i} \quad (2.17)$$

$$(BFGS-B) \quad B'_{i+1} = (I - \rho_i \mathbf{s}^i (\mathbf{y}^i)^T) B'_i (I - \rho_i \mathbf{y}^i (\mathbf{s}^i)^T) - \rho_i \mathbf{s}^i (\mathbf{s}^i)^T \quad (2.18)$$

In che modo scegliere  $B^0$ ? una prima idea è  $\gamma I$  per qualche  $\gamma$ . Un'alternativa è calcolare un'approssimazione di  $\nabla^2 f(\mathbf{x}^0)$  attraverso il metodo delle differenze finite. Si può in qualche modo pensare ai metodi DFP e BFGS come metodi per imparare l'hessiana  $\nabla^2 f(\mathbf{x})$  tramite esempi dati da  $\nabla f(\mathbf{x})$ . Lo studio della convergenza del metodo è complicata ma nella pratica non è molto distante alla velocità del metodo di Newton, ed è a volte anche meno incline a fermarsi in minimi locali: all'inizio il metodo è  $\approx$  gradiente, quindi riesce a sfuggire meglio a minimi locali, mentre nelle ultime iterazioni, avendo imparato meglio  $\nabla^2 f(\mathbf{x})$ , si possono vedere le proprietà di convergenza veloce del metodo di Newton.

### 2.4.3 Limited-memory BFGS

Possiamo essere ancora più efficienti ad ogni iterazione: notiamo che sviluppando (BFGS-B) e chiamando  $V^i = I - \rho_i \mathbf{y}^i (\mathbf{s}^i)^T$ , otteniamo

$$B^{i+1} = (V^i)^T B^i V^i + \rho^i \mathbf{s}^i (\mathbf{s}^i)^T \quad (2.19)$$

Possiamo allora provare a sviluppare la ricorsione, ottenendo

$$\begin{aligned} B^{i+1} = & (V^0 \cdot V^1 \cdot \dots \cdot V^i)^T B^0 (V^0 \cdot V^1 \cdot \dots \cdot V^i) \\ & + \rho_0 (V^1 \cdot \dots \cdot V^i)^T \mathbf{s}^0 \cdot (\mathbf{s}^0)^T (V^1 \cdot \dots \cdot V^i) \\ & + \rho_1 (V^2 \cdot \dots \cdot V^i)^T \mathbf{s}^1 \cdot (\mathbf{s}^1)^T (V^2 \cdot \dots \cdot V^i) \\ & + \dots \\ & + \rho_i \mathbf{s}^i (\mathbf{s}^i)^T \end{aligned} \quad (2.20)$$

In questo modo è facile vedere che  $B^{i+1}$  può essere calcolato solo tramite prodotti scalari tra vettori, che richiedono  $O(n)$  operazioni, se svolgiamo i prodotti in un ordine “furbo”. Se lasciamo la formula così com'è ci servono comunque  $O(n^2)$  operazioni totali. Possiamo perciò troncare la formula dopo  $k$  passi:

$$\begin{aligned} B^{i+1} = & (V^{i-k} \cdot V^{i-k+1} \cdot \dots \cdot V^i)^T B^{i-k} (V^{i-k} \cdot V^{i-k+1} \cdot \dots \cdot V^i) \\ & + \rho_{i-k} (V^{i-k+1} \cdot \dots \cdot V^i)^T \mathbf{s}^{i-k} \cdot (\mathbf{s}^{i-k})^T (V^{i-k+1} \cdot \dots \cdot V^i) \\ & + \rho_{i-k+1} (V^{i-k+2} \cdot \dots \cdot V^i)^T \mathbf{s}^{i-k+1} \cdot (\mathbf{s}^{i-k+1})^T (V^{i-k+2} \cdot \dots \cdot V^i) \\ & + \dots \\ & + \rho_i \mathbf{s}^i (\mathbf{s}^i)^T \end{aligned} \quad (2.21)$$

*Osservazione 6.* In pratica, non serve che in (2.21)  $B^{i-k}$  sia davvero la  $B$  calcolata all'iterazione  $i-k$ , ma la matrice può essere scelta in modo arbitrario per ogni  $i$ , in modo sparso: ad esempio  $B^{i-k} = \gamma_i I$  con  $\gamma_i = \mathbf{s}^i \cdot \mathbf{y}^{i-1} / \|\mathbf{y}^{i-1}\|^2$ . In questo modo si può dimostrare che servono  $O(kn)$  operazioni totali.

Se  $n$  è molto grande, scegliendo un  $k \ll n$  si può ottenere un vantaggio computazionale significativo, ma l'efficienza del metodo in termini di operazioni se  $k$  è molto piccolo è paragonabile al metodo del gradiente. Se invece  $k$  cresce l'efficienza comincia ad essere simile a quella della BFGS, e quindi si avvicina al metodo di Newton.

## 2.5 Gradiente coniugato e Heavy Ball

Se vogliamo dei metodi ancora meno costosi, invece che modificare la direzione di decrescita  $\mathbf{d}_i$  con una matrice  $H$ , possiamo deviarla sommando un vettore  $\mathbf{v}^i = \mathbf{v}^i$  è  $\mathbf{v}^i = \beta_i \mathbf{d}^{i-1}$ , con  $\beta_i > 0$ : in questo modo, assumendo  $\mathbf{v}^0 = 0$ , otteniamo che

$$\mathbf{d}^i = -\sum_{j=0}^i \gamma_j \nabla f(\mathbf{x}^j), \quad \gamma_j = \sum_{k=j+1}^i \beta_k$$

cioè ad ogni passo stiamo tenendo conto di tutta la storia precedente.

Tuttavia il problema di scegliere i  $\beta_i$  tali per cui le direzioni sono di discesa è non banale: se per esempio  $\beta_i \rightarrow 0$ , la direzione sarà di discesa per  $i$  abbastanza grande, tuttavia  $\mathbf{d}^i \rightarrow \nabla f(\mathbf{x}^i)$ , dunque ci possiamo aspettare una convergenza lenta come quella della discesa del gradiente.

### 2.5.1 Metodo del gradiente coniugato

Se applichiamo il metodo di Newton con Line Search esatta ad una funzione quadratica, abbiamo visto che  $\mathbf{d}^{i+1} \perp \mathbf{d}^i$ : questo accade perché  $\mathbf{x}^{i+1}$  è il minimo di  $f$  lungo  $\mathbf{d}^i$ , ovvero nel sottospazio vettoriale (unidimensionale) generato da  $\mathbf{d}^i$  che denotiamo con  $\text{span}\{\mathbf{d}^i\}$ . Tale proprietà si perde al passo successivo: niente ci garantisce che  $\mathbf{x}^{i+2}$  sia ancora il minimo di  $f$  lungo  $\mathbf{d}^i$ . L'idea è quindi richiedere che  $\mathbf{x}^{i+1}$  sia il minimo in tutto il sottospazio  $\text{span}\{\mathbf{d}^1, \dots, \mathbf{d}^i\}$ . Ciò è possibile prendendo  $\mathbf{d}^i = -\nabla f(\mathbf{x}^i) + \beta_i \mathbf{d}^{i-1}$ , con

$$\beta_i = \frac{\nabla f(\mathbf{x}^i)^T Q \mathbf{d}^i}{(\mathbf{d}^{i-1})^T Q \mathbf{d}^{i-1}}. \quad (2.22)$$

La formula (2.22) si può riscrivere più semplicemente come

$$\beta_i = \frac{\|\nabla f(\mathbf{x}^i)\|}{\|\nabla f(\mathbf{x}^{i-1})\|} \quad (2.23)$$

Questa formula è detta di *Fletcher-Reeves*.

Queste considerazioni portano al seguente algoritmo:

---

**Algoritmo 3** Algoritmo del gradiente coniugato per funzioni quadratiche

---

```

procedure  $x = NCG(f, x, \varepsilon)$ 
   $\nabla f^- = 0$ ;
  while ( $\|\nabla f(x)\| > \varepsilon$ ) do
    if ( $\nabla f^- = 0$ ) then  $d \leftarrow -\nabla f(x)$ ;
    else  $\{ \beta = \|\nabla f(x)\|^2 / \|\nabla f^-\|^2; d \leftarrow -\nabla f(x) + \beta d^-; \}$ 
     $\alpha \leftarrow LS(f, x, d); x \leftarrow x + \alpha d; d^- \leftarrow d; \nabla f^- \leftarrow \nabla f(x)$ ;

```

---

Per  $f$  non quadratica la Line Search esatta non è possibile, o comunque non conveniente, quindi possiamo escludere che  $\mathbf{x}^{i+1}$  sia il minimo nella direzione  $\mathbf{d}^i$ , e a maggior ragione nel sottospazio generato da tutte le direzioni precedenti. Ci sono tuttavia alcuni modi di scegliere  $\beta_i$  in modo che  $\mathbf{x}^i$  si avvicini a tale minimo. Tali formule coincidono con la (2.23) se  $f$  è quadratica:

**Polak-Ribière:**

$$\beta_i^{PR} = \frac{(\nabla f(\mathbf{x}^i) - \nabla f(\mathbf{x}^{i-1})) \cdot \nabla f(\mathbf{x}^i)}{\|\nabla f(\mathbf{x}^{i-1})\|^2}$$

**Hestenes-Stiefel:**

$$\beta_i^{HS} = \frac{(\nabla f(\mathbf{x}^i) - \nabla f(\mathbf{x}^{i-1})) \cdot \nabla f(\mathbf{x}^i)}{(\nabla f(\mathbf{x}^i) - \nabla f(\mathbf{x}^{i-1})) \cdot \mathbf{d}^{i-1}}$$

**Day-Yuan:**

$$\beta_i^{DY} = \frac{\|\nabla f(\mathbf{x}^i)\|^2}{(\nabla f(\mathbf{x}^i) - \nabla f(\mathbf{x}^{i-1})) \cdot \mathbf{d}^{i-1}}$$

Tale modo di scegliere  $\mathbf{d}^i$  viene applicato assieme ad una Armijo-Wolfe Line Search.

In generale la convergenza dipende dal tipo di  $\beta_i$  scelto. Assumendo ragionevoli ipotesi su  $f$ , si può dimostrare che scegliendo  $\beta_i$  come nella formula di Fletcher-Reeves (2.23) il metodo converge scegliendo  $m_1 < m_3 < \frac{1}{2}$  con una line search che soddisfa la condizione di Wolfe forte.

Usando  $\beta_i^{PR}$  o  $\beta_i^{HS}$ , una buona idea è il *restart*, ovvero scegliere

$$\beta_i = \max\{\beta_i^*, 0\}$$

dove  $*$  =  $PR, HS$ , ovvero prendere di tanto in tanto  $\mathbf{d}^i = -\nabla f(\mathbf{x}^i)$ .

Per quanto riguarda la convergenza, si rimanda alla parte di Analisi Numerica (coming soon) per l'analisi

### 2.5.2 Heavy Ball

Un'alternativa ancora meno costosa rispetto al metodo del gradiente coniugato è il metodo Heavy Ball, in cui prendiamo  $\beta$  costante:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \alpha_i \nabla f(\mathbf{x}^i) + \beta(\mathbf{x}^i - \mathbf{x}^{i-1}).$$

- $\beta(\mathbf{x}^i - \mathbf{x}^{i-1})$  è un termine che dà il momento della discesa: grazie a questo termine  $\mathbf{x}^{i+1}$  si mantiene circa nella stessa direzione;
- l'antigradiente è la forza che devia la traiettoria verso un punto di discesa.

Un momento  $\beta_i$  molto grande ci permette di avere meno zig-zag nella discesa e di ottenere perciò una convergenza migliore. Si può dimostrare che con la giusta scelta di  $\alpha_i$  e  $\beta_i$  si ha convergenza lineare con tasso migliore di quello della discesa del gradiente classica: si può dimostrare che se  $f$  è  $L$ -smooth e  $\tau$ -convessa,

$$\|\mathbf{x}^{i+1} - \mathbf{x}_*\| \leq r \|\mathbf{x}^i - \mathbf{x}_*\|$$

con  $r = \frac{\sqrt{k} - 1}{\sqrt{k} + 1}$  e  $k = \frac{L}{\tau}$ . La convergenza è perciò almeno lineare.

## 2.6 Ottimizzazione di funzioni non differenziabili

In questa sezione vediamo come si possono ottimizzare funzioni continue ma non  $C^1$ . La motivazione di questa trattazione è per esempio la minimizza-



zione della regolarizzazione L1: la norma  $\|\cdot\|_1$  non è differenziabile in punti  $\mathbf{x}$  tale che esiste un indice  $i$  con  $x_i = 0$ .

### 2.6.1 Subgradiente e subdifferenziale

Per tutta la trattazione assumeremo la funzione  $f$  almeno convessa. Dunque anche se  $f$  non è differenziabile, esiste un unico minimo locale.

**Definizione.** Data una funzione  $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  convessa, diciamo che un vettore  $\mathbf{s}$  è un suo subgradiente in  $\mathbf{x}$  se per ogni  $\mathbf{z} \in X$

$$f(\mathbf{z}) \geq f(\mathbf{x}) + \mathbf{s} \cdot (\mathbf{z} - \mathbf{x})$$

L'utilità di questa definizione è che se  $0$  è un subgradiente di  $f$  in  $\mathbf{x}$ , allora  $\mathbf{x}$  è chiaramente il minimo globale: per ogni  $\mathbf{z}$ ,  $f(\mathbf{z}) \geq f(\mathbf{x})$ . Il problema è che  $0$  potrebbe essere solo uno dei tanti subgradienti in  $\mathbf{x}$ :

**Definizione.** Il *subdifferenziale* di  $f$  in  $\mathbf{x}$  è l'insieme

$$\partial f(\mathbf{x}) = \{\mathbf{s} \mid \mathbf{s} \text{ è subgradiente di } f \text{ in } \mathbf{x}\}$$

In particolare, se  $f$  è differenziabile in  $\mathbf{x}$ , allora  $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ , cioè l'unico subgradiente è il gradiente stesso. Notiamo che per ogni direzione  $\mathbf{d}$  e per ogni  $\mathbf{s} \in \partial f(\mathbf{x})$ ,

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{d}} \geq \mathbf{s} \cdot \mathbf{d}$$

Ciò implica che  $\mathbf{d}$  è una direzione di decrescita se e solo se  $\mathbf{s} \cdot \mathbf{d} \leq 0$  per ogni subgradiente  $\mathbf{s}$ , quindi la direzione

$$\mathbf{s}^* = - \arg \min_{\mathbf{s} \in \partial f(\mathbf{x})} \|\mathbf{s}\|$$

è la direzione di massima decrescita.

Notiamo inoltre che  $\mathbf{x}$  è minimo locale se e solo se  $0 \in \partial f(\mathbf{x})$ .

### 2.6.2 Metodi subgradiente

Chiamiamo metodo del subgradiente un metodo del tipo  $\mathbf{x}^{i+1} = \mathbf{x}^i + \alpha_i \mathbf{d}^i$ , con  $\mathbf{d}^i \in -\partial f(\mathbf{x}^i)$ . A differenza del metodo del gradiente, ci sono più direzioni possibili se la funzione non è differenziabile in  $\mathbf{x}^i$ . In particolare quindi se

$f(\mathbf{x}^i)$  è vicino ad un minimo locale  $f_*$ , non è detto che la direzione  $\mathbf{d}^i$  sia vicina a 0, perciò la condizione  $\|\mathbf{d}^i\| < \varepsilon$  non è una condizione di stop efficace come nel caso del metodo del gradiente.

Non possiamo poi usare un passo di grandezza fissa  $\alpha_i = \alpha$ : prendiamo per esempio come funzione da ottimizzare  $f(x) = L|x|$ . Scelto un qualsiasi passo  $\alpha > 0$ , se partiamo da  $x^0 = -\frac{\alpha L}{2}$ , essendo  $f'(x^0) = -L$ ,

$$x^1 = x^0 + \alpha L = -\frac{\alpha L}{2} + \alpha L = \frac{\alpha L}{2}.$$

Continuando,  $f'(x^1) = L$ , dunque

$$x^2 = x^1 - \alpha L = -\frac{\alpha L}{2} = x^0.$$

L'algoritmo va dunque in loop. Osserviamo che entrambi questi problemi derivano appunto dal fatto che il subgradiente, anche se siamo molto vicini al minimo, potrebbe non essere piccolo: nell'esempio precedente il gradiente è  $\pm L$  in ogni punto diverso da 0.

## DSS

Una soluzione che funziona è prendere  $\alpha_i$  corti ma non troppo, cioè tali che

$$\sum_{i=1}^{\infty} \alpha_i = \infty,$$

ma

$$\sum_{i=1}^{\infty} \alpha_i^2 < \infty.$$

Questo tipo di passi si dice *Diminishing-Square Summable (DSS)*. Un esempio è  $\alpha_i = \frac{1}{i}$ . Questa scelta funziona perché se denotiamo con  $\mathbf{g}^i$  il subgradiente scelto al passo  $i$ , possiamo notare che

$$\begin{aligned} \|\mathbf{x}^{i+1} - \mathbf{x}_*\|^2 &= \|\mathbf{x}^i - \alpha_i \mathbf{g}^i - \mathbf{x}_*\|^2 \\ &= \|\mathbf{x}^i - \mathbf{x}_*\|^2 + 2\alpha_i \mathbf{g}^i \cdot (\mathbf{x}_* - \mathbf{x}^i) + \alpha_i^2 \|\mathbf{g}^i\|^2 \\ &\leq \|\mathbf{x}^i - \mathbf{x}_*\|^2 + 2\alpha_i (f(\mathbf{x}_*) - f(\mathbf{x}^i)) + \alpha_i^2 \|\mathbf{g}^i\|^2, \end{aligned}$$

dove la disuguaglianza segue dal fatto che  $\mathbf{g}^i$  è un subgradiente. Assumiamo ora inoltre che esista un  $L$  per cui  $\|\mathbf{g}^i\| \leq L$ ,

$$\begin{aligned}\|\mathbf{x}^{i+1} - \mathbf{x}_*\| &\leq \|\mathbf{x}^i - \mathbf{x}_*\|^2 + 2\alpha^i(f(\mathbf{x}_*) - f(\mathbf{x}^i)) + \alpha_i^2 L^2 \\ &\leq \|\mathbf{x}^i - \mathbf{x}_*\|^2 + 2\alpha^i(f(\mathbf{x}_*) - f(\mathbf{x}^i)) + \alpha_i^2 L^2\end{aligned}$$

e per induzione otteniamo che

$$\|\mathbf{x}^{i+1} - \mathbf{x}_*\| \leq \|\mathbf{x}^1 - \mathbf{x}_*\|^2 + \sum_{k=1}^i \alpha_k(f_* - f(\mathbf{x}^k)) + \sum_{k=i}^i \alpha_k^2 L^2. \quad (2.24)$$

Sia  $\bar{x}$  un punto limite di  $\mathbf{x}^i$  (la successione è limitata poichè in (2.24) il secondo termine è negativo e il terzo è  $< +\infty$  per (DSS), quindi esiste un punto limite). Supponiamo allora per assurdo che  $\mathbf{x}^i \not\rightarrow \mathbf{x}_*$ , cioè esiste un  $\delta > 0$  per cui  $f(\mathbf{x}^i) - f_* \geq \delta$ . Prendendo il limite per  $i \rightarrow \infty$  nella disequazione (2.24), otteniamo

$$\|\bar{\mathbf{x}} - \mathbf{x}_*\| \leq \|\mathbf{x}^1 - \mathbf{x}_*\|^2 + \delta \sum_{k=1}^{\infty} \alpha_k + \sum_{k=i}^{\infty} \alpha_k^2 L^2 = -\infty,$$

che è assurdo.

Dal punto di vista pratico però la velocità di convergenza del metodo è molto lenta, perciò ci serve un metodo più veloce per scegliere i passi.

### Passi di Polyak

Abbiamo precedentemente dimostrato che  $\|\mathbf{x}^{i+1} - \mathbf{x}_*\|^2 \leq \|\mathbf{x}^i - \mathbf{x}_*\|^2 + 2\alpha^i(f(\mathbf{x}_*) - f(\mathbf{x}^i)) + \alpha_i^2 \|\mathbf{g}^i\|^2$ . Notiamo allora che

$$\min 2\alpha^i(f(\mathbf{x}_*) - f(\mathbf{x}^i)) + \alpha_i^2 \|\mathbf{g}^i\|^2 = \frac{f(\mathbf{x}^i) - f_*}{\|\mathbf{g}^i\|^2}$$

Un passo della forma  $\alpha^i = \beta^i \frac{f(\mathbf{x}^i) - f_*}{\|\mathbf{g}^i\|^2}$  è detto di Polyak. Abbiamo appena visto che  $\beta^i = 1$  è ottimale, ma anche  $\beta^i \in (0, 2)$  ci garantisce che  $\|\mathbf{x}^{i+1} - \mathbf{x}_*\|^2 \leq \|\mathbf{x}^i - \mathbf{x}_*\|^2$ .

Con questa scelta possiamo dimostrare che la velocità di convergenza è  $O(1/\varepsilon^2)$ , perciò il metodo è comunque molto lento: ad esempio per passare da  $\varepsilon = 10^{-3}$  a  $\varepsilon = 10^{-4}$  servono circa 100 volte il numero di iterazioni usate per arrivare a  $10^{-3}$ . In ogni caso la scelta del passo è ottimale se conosciamo  $f_*$ ,

**Target Level Stepsize**

TODO

**Deflected Subgradient**

Se vogliamo una velocità di convergenza migliore possiamo provare a cambiare la direzione di decrescita, deviando il gradiente. Prendiamo  $\mathbf{x}^{i+1} = \mathbf{x}^i - \alpha^i \mathbf{d}^i$  dove

$$\mathbf{d}^i = \gamma^i \mathbf{g}^i + (1 - \gamma^i) \mathbf{d}^{i-1}$$

con  $0 < \gamma^i < 1$  e  $\mathbf{g}^i$  un subgradiente in  $\mathbf{x}^i$ . Se vogliamo qualche risultato di convergenza teorica, dobbiamo usare qualche accorgimento.

**2.6.3 Smoothed Gradient**

Supponiamo che  $f$  sia della forma  $f(x) = \max\{\mathbf{x}^T A \mathbf{z} : \mathbf{z} \in Z\}$ , cioè per ogni  $\mathbf{x}$ ,  $f(\mathbf{x})$  è un problema di massimizzazione con funzione obbiettivo lineare. In generale una funzione del genere è continua ma non differenziabile: si può dimostrare che se  $\bar{\mathbf{z}}$  è ottimale per  $\mathbf{x}$  (nel problema di massimizzazione), allora  $\bar{\mathbf{z}} \in \partial f(\mathbf{x})$ , dunque se per un qualche  $\mathbf{x}$  esiste più di un  $\mathbf{z}$  ottimale,  $|\nabla f(\mathbf{x})| > 1$ , e quindi  $f$  non è differenziabile in  $\mathbf{x}$ . Prendiamo allora una modificazione di  $f$  “smoothed”, cioè

$$f_\mu(\mathbf{x}) = \max\{\mathbf{x}^T A \mathbf{z} + -\mu \|\mathbf{z}\|^2 : \mathbf{z} \in Z\}. \quad (2.25)$$

$f_\mu$  così definito è differenziabile, e possiamo perciò applicare uno dei metodi che abbiamo già descritto per funzioni differenziabili. Scegliendo  $\mu$  piccolo ( $\mu = O(\varepsilon)$ ), la velocità di convergenza risulta essere  $O(1/\varepsilon)$ , che è molto migliore di quella del metodo subgradiente, ma comunque piuttosto lenta.

**2.6.4 Bundle Methods**

L'idea fondamentale dei bundle methods è spostarsi solo quando è davvero conveniente, e altrimenti raccogliere informazioni per provare una nuova direzione: TO DO

# Capitolo 3

## Ottimizzazione vincolata

### 3.1 Introduzione all'ottimizzazione vincolata

Torniamo al problema generale descritto nell'introduzione, cioè risolvere

$$(P) \quad \min f(x) \quad x \in X \subseteq \mathbb{R}^n \quad (3.1)$$

Dove  $X$  è la regione ammissibile per le soluzioni. Se  $x \in X$  diciamo che è una soluzione ammissibile, altrimenti diciamo che  $x$  non è ammissibile.  $x_*$  che soddisfa il problema di minimo viene spesso detto *ottimo*, e indicheremo il valore ottimo  $f_* = f(x_*)$  con  $v(P)$ . Notiamo che in teoria potremmo nascondere i vincoli nella funzione obiettivo, e passare così ad un problema non vincolato, in questo modo:

$$(P) \quad \min f(x) + \mathbb{1}_X(x)$$

con

$$\mathbb{1}_X(x) = \begin{cases} 0 & x \in X \\ +\infty & x \notin X \end{cases}.$$

Questa idea non è quasi mai buona, perchè la funzione indicatrice  $\mathbb{1}_X$  è una funzione difficile da ottimizzare, in quanto non continua.

Ci sono alcuni problemi che possono nascere quando imponiamo dei vincoli sulle soluzioni:

- Se  $X = \emptyset$ ,  $v(P) = +\infty = \inf\{\emptyset\}$ . In questo caso non possiamo risolvere (P) trovando una soluzione (perché non ce ne sono), ma dobbiamo dimostrare che  $X$  è vuoto.

- Anche se  $X$  non è vuoto, il minimo potrebbe comunque non appartenere a  $X$ : prendiamo per esempio  $\min x$  con  $x \in (0, 1)$ . Chiaramente vorremo prendere come soluzione  $x = 0$ , che tuttavia non appartiene alla regione ammissibile.

In ogni caso questi problemi non sono molto importanti nel Machine learning, perciò non daremo loro molto peso nella trattazione.

Come nel caso non vincolato, l'ottimizzazione globale è difficile, perciò ci restringiamo a trovare un minimo locale cioè  $x^* = \arg \min \{f(x) : x \in B(x^*, \varepsilon) \cap X\}$  per qualche  $\varepsilon > 0$ .

### 3.1.1 Cenni di topologia

**Definizione.** Diciamo che un insieme  $X \subseteq \mathbb{R}^n$  è *aperto* se per ogni punto  $x \in X$  esiste un  $\varepsilon > 0$  tale che la palla  $B(x, \varepsilon)$  è contenuta in  $X$ . Diciamo che  $X$  è *chiuso* se il suo complementare  $Y = \mathbb{R}^n \setminus X$  è aperto.

Una caratterizzazione equivalente delle due definizioni è la seguente: Chiamiamo  $\overset{\circ}{X}$  la parte interna di  $X$ , ovvero tutti i punti  $x \in X$  per cui vale che esiste un  $\varepsilon > 0$  tale che la palla  $B(x, \varepsilon)$  è contenuta in  $X$ . È evidente che  $X$  è aperto se e solo se  $X = \overset{\circ}{X}$ .

Chiamiamo poi  $\partial X$  la frontiera di  $X$  ovvero i punti  $x \in \mathbb{R}^n$  per cui vale che per ogni  $\varepsilon > 0$  esistono  $y_1, y_2 \in B(x, \varepsilon)$  tali che  $y_1 \in X$  e  $y_2 \notin X$ . La chiusura di  $X$ , denotata con  $\bar{X}$ , è  $\overset{\circ}{X} \cup \partial X$ . Si può dimostrare facilmente che  $X$  è chiuso se e solo se  $X = \bar{X}$ .

*Esempio.* Alcuni esempi di insiemi aperti e chiusi sono:

- Dati  $a < b \in \mathbb{R}$ , l'intervallo  $(a, b)$  è aperto, mentre l'intervallo  $[a, b]$  è chiuso.
- Gli intervalli  $[a, b)$  e  $(a, b]$  non sono né aperti né chiusi.
- La frontiera di tutti questi insiemi è  $\{a, b\}$ , mentre la parte interna è  $(a, b)$ .
- La palla aperta  $B(x, \varepsilon) = \{y \in \mathbb{R}^n : \|x - y\| < \varepsilon\}$  è chiaramente aperta, mentre la palla chiusa  $\bar{B}(x, \varepsilon) = \{y \in \mathbb{R}^n : \|x - y\| \leq \varepsilon\}$  è chiusa.

Un'altra caratterizzazione utile degli insiemi chiusi è la seguente:

$X$  è chiuso se e solo se, per ogni successione  $(x_i)_{i=1}^{\infty}$  di punti appartenenti a  $X$ , se  $x_i \rightarrow x$ , allora  $x \in X$ . Nel seguito assumeremo sempre  $X$  chiuso. Così facendo, ci assicuriamo che se  $x \in \partial X$ , allora  $x$  è ammissibile. Nella prossima sezione infatti daremo delle condizioni per cui possiamo assicurarci che  $x \in \partial X$  sia ottimale.

## 3.2 Condizioni di ottimalità

### 3.2.1 Cono Tangente

Se  $x_* \in \overset{\circ}{X}$  è un ottimo locale, allora è ovvio che  $\nabla f(x_*) = 0$ . Tuttavia così non è se  $x \in \partial X$ : un esempio banale è  $\min\{x : x \in [0, 1]\}$ : in questo caso 0 è un minimo, ma  $f'(x) = (x)' = 1$  per ogni  $x$ , dunque  $f'(0) = 0$ . Ci servono perciò delle diverse condizioni di ottimalità per punti appartenenti alla frontiera. Introduciamo per questo motivo il cono tangente a  $X$  in  $x$ :

**Definizione.** Il cono tangente è l'insieme

$$T_X(x) = \left\{ d : \exists (z_i) \subseteq X, z_i \rightarrow x, \text{ e } (t_i) > 0, t_i \rightarrow 0 \text{ t.c. } \lim_{i \rightarrow +\infty} \frac{z_i - x}{t_i} = d \right\}$$

**Definizione.** Un insieme  $\mathcal{C}$  è un *cono* se vale  $x \in \mathcal{C} \Rightarrow tx \in \mathcal{C}$  per ogni  $t \geq 0$ .

*Osservazione 7.* Si verifica facilmente che  $T_X(x)$  è effettivamente un cono.

**Proposizione 3.2.1.** Se  $x \in X$  è un minimo locale, vale

$$(TCC) \quad \nabla f(x) \cdot d \leq 0 \quad \forall d \in T_X(x)$$

*Dimostrazione.* Supponiamo per assurdo che  $x$  sia un ottimo locale, ma esista  $d \in T_X(x)$  tale che  $\nabla f(x) \cdot d < 0$ .  $T_X(x)$  è un cono e  $d \neq 0$ , quindi possiamo assumere che  $\|d\| = 1$ . Dalla definizione di  $d \in T_X(x)$  si vede che esiste una successione  $(z_i)$  e una successione  $(t_i) > 0$  tale per cui

$$\lim_{i \rightarrow \infty} \frac{\|z_i - x\|}{t_i} = 1,$$

cioè  $\|z_i - x\|$  e  $t_i$  convergono a 0 con la stessa velocità. Per la formula di Taylor al primo ordine,

$$f(z_i) - f(x) = \nabla f(x) \cdot (z_i - x) + o(\|z_i - x\|)$$

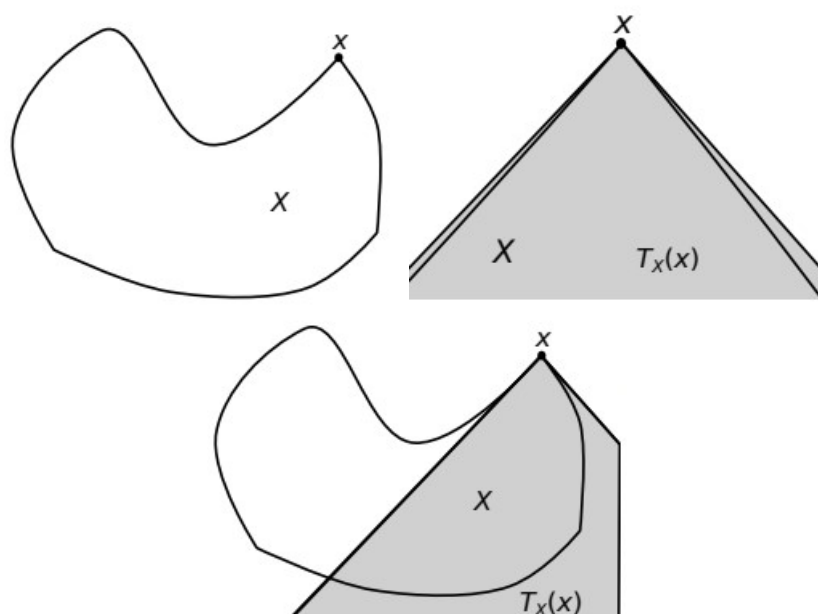


Figura 3.1: Se zoomiamo la regione  $X$  in  $x$ , localmente  $X$  assomiglia ad un cono. Tale cono approssima  $X$  molto vicino a  $x$ . Nella terza figura vediamo  $T_X(x)$  e  $X$  nella scala originaria.



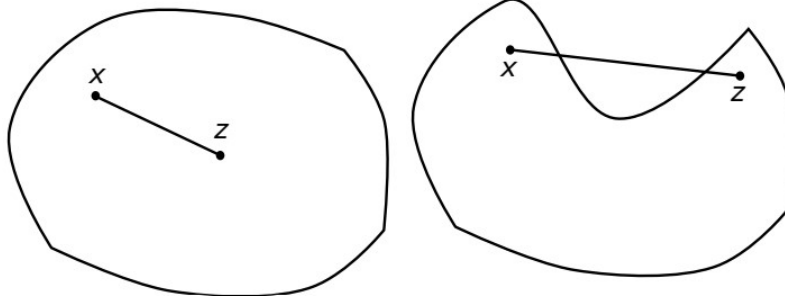


Figura 3.2: A sinistra un insieme convesso, a destra un insieme non convesso.

Per ipotesi  $\nabla f(x) \cdot d < 0$ , dunque poiché  $t_i > 0$  e  $\frac{z_i - x}{t_i} \rightarrow d$ , da un certo  $i$  in poi  $\nabla f(x_i) \cdot (z_i - x) < 0$ . Inoltre

$$\lim_i \frac{\nabla f(x) \cdot (z_i - x)}{t_i} = \nabla f(x) \cdot d = \text{costante},$$

Perciò  $\nabla f(x) \cdot (z_i - x)$  va a 0 velocemente quanto  $t_i$ , e quindi quanto  $\|z_i - x\|$ . Dunque da un certo  $i$  in poi,  $f(z_i) - f(x) < 0$ . Poiché  $z_i \rightarrow x$  si ottiene che per ogni  $\epsilon$  esiste  $z_i \in B(x, \epsilon)$  con  $f(z_i) < f(x)$ , contraddicendo che  $x$  sia un minimo locale.  $\square$

### 3.2.2 Insiemi convessi

Abbiamo appena dimostrato che  $x$  minimo locale  $\Rightarrow$  (TCC). L'implicazione inversa non vale in generale ma si può vedere che vale se  $T_X(x) + x \supset X$ . Ciò è equivalente a dire che  $X$  è convesso:

**Definizione.** Diciamo che un insieme  $C$  è *convesso* se, per ogni  $x, z \in X$ , il segmento tra  $x$  e  $z$ ,  $\text{conv}(x, z) = \{y : y = tx + (1 - t)z, t \in [0, 1]\}$ , è tutto contenuto in  $C$ .

Dato un qualsiasi insieme  $X \subseteq \mathbb{R}^n$ , possiamo definire l'involuppo convesso di  $X$  come

$$\text{conv}(X) = \bigcap \{C : C \text{ è convesso}, X \subseteq C\} = \bigcup_{x, y \in X} \text{conv}(x, y)$$

cioè il più piccolo insieme convesso contenente  $X$  o equivalentemente l'unione di tutti i segmenti con entrambi gli estremi in  $X$ .

**Proposizione 3.2.2.** *Se  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  è una funzione,  $f$  è convessa se e solo se il suo epigrafico  $\text{epi}(f) = \{(x, z) \in \mathbb{R}^{n+1} : z \geq f(x)\}$  è convesso.*

**Proposizione 3.2.3.** *Se  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  è convessa, i suoi insiemi di sottolivello  $S(f, z) = \{x \in \mathbb{R}^n : f(x) \leq z\}$  è convesso. Il viceversa non è vero,  $S(f, z)$  convesso  $\Leftrightarrow f$  quasi-convessa.*

Un buon numero di insiemi che già abbiamo visto sono convessi:

- l'involuppo convesso di un numero finito di punti

$$\text{conv}(\{x_1, \dots, x_k\}) = \{x = \sum_{i=1}^k \alpha_i x_i : \sum \alpha_i = 1, \alpha_i \geq 0\}.$$

- gli iperpiani affini  $\mathcal{H} = \{x \in \mathbb{R}^n : ax = b\}$ , che sono insiemi di livello di funzioni lineari affini.
- i semispazi affini  $\mathcal{S} = \{x \in \mathbb{R}^n : ax \leq b\}$ , che sono insiemi di sottolivello di funzioni lineari affini.
- Gli ellissoidi  $\mathcal{E}(Q, x, r) = \{z \in \mathbb{R}^n : (z - x)^T Q (z - x) \leq r\}$ , che sono insiemi di sottolivello di funzioni quadratiche.
- Le palle in norma  $p$  per ogni  $p \geq 1$ .

Come abbiamo già accennato, se  $X$  è convesso allora se in  $x$  vale (TCC), ciò è sufficiente perché  $x$  sia ottimale.

**Definizione.** Il cono delle direzioni ammissibili in  $x$  è

$$F_X(x) = \{d : \exists \bar{\varepsilon} > 0 \text{ t.c. } x + \bar{\varepsilon}d \in X\}.$$

Si può dimostrare che  $F_X(x)$  è effettivamente un cono, e che dato un insieme  $X$  qualsiasi (anche non convesso),  $X \subseteq F_X(x) + x$  per ogni  $x$ . Inoltre se  $X$  è convesso,  $F_X(x) \subseteq T_X(x)$  (in particolare  $T_X$  è la chiusura di  $F_X$ ). Dunque se  $X$  è convesso,  $T_X(x) + x \supseteq X$ , e (TCC) implica che nessuna direzione ammissibile sia anche di decrescita. Inoltre se  $f$  è convessa esiste un unico minimo globale in  $X$ . Abbiamo dimostrato perciò

**Proposizione 3.2.4.** *Se  $(P)$  è convesso (ovvero  $f$  e  $X$  sono convessi), allora  $x_*$  soddisfa (TCC)  $\Rightarrow x_*$  è un minimo globale.*

Osserviamo che se  $x \in \overset{\circ}{X}$ ,  $T_X(x) = F_X(x) = \mathbb{R}^n$ , perciò la (TCC) si traduce in  $\nabla f(x) \cdot d \geq 0$  per ogni  $d$ , e quindi  $\nabla f(x) = 0$ . Perciò in generale possiamo vedere (TCC) come una generalizzazione del fatto che  $x$  sia un punto stazionario: è solo necessaria e non sufficiente nel caso generale, ma se (P) è convesso siamo fortunati, perché esiste un unico punto stazionario, ed è un minimo globale.

Non è ovvio come assicurarci che un punto  $x$  soddisfi (TCC): ci sarebbero infiniti controlli da fare, uno per ogni direzione. Nella prossima sezione perciò vogliamo trasformare questa condizione in un'altra più utilizzabile in pratica.

### 3.2.3 Lemma di Farkas

Fino ad ora abbiamo assunto  $X$  come un generico sottoinsieme (chiuso) di  $\mathbb{R}^n$ . Nella pratica però ci serve una descrizione algebrica di  $X$ : in particolare lo pensiamo come descritto da un insieme di disequazioni e di equazioni. Una forma standard che useremo è la seguente, dati due insiemi di indici  $\mathcal{I}$  e  $\mathcal{J}$ , e delle funzioni  $g_i$  per ogni  $i \in \mathcal{I}$  e  $h_j$  per ogni  $j \in \mathcal{J}$ :

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \forall i \in \mathcal{I}, h_j(x) = 0 \forall j \in \mathcal{J}\} \quad (3.2)$$

equivalentemente possiamo scrivere  $X$  con vincoli in forma vettoriale: dette  $G(x) = [g_i(x)]_{i \in \mathcal{I}} : \mathbb{R}^n \rightarrow \mathbb{R}^{|\mathcal{I}|}$ ,  $H(x) = [h_j(x)]_{j \in \mathcal{J}} : \mathbb{R}^n \rightarrow \mathbb{R}^{|\mathcal{J}|}$ ,

$$X = \{x : G(x) \leq 0, H(x) = 0\} \quad (3.3)$$

Osserviamo che, almeno dal punto di vista della trattazione teorica (dal punto di vista algoritmico spesso non conviene) possiamo assumere che non ci siano vincoli di uguaglianza: infatti  $h_j = 0$  corrisponde a  $h_j \leq 0, -h_j \leq 0$ . Notiamo che  $X$  definito come in 3.2 è convesso se e solo se le  $g_i$  sono convesse e sia  $h_j$  che  $-h_j$  sono convesse, cioè le  $h_j$  sono lineari. Molto spesso ci restringeremo a questo caso o ad altri ancora più semplici.

*Esempio.* Cominciamo dal caso più semplice di tutti, quello di vincoli di uguaglianza lineari:

$$(P) \quad \min\{f(x) : Ax = b\},$$

dove  $A \in \mathbb{R}^{m \times n}$  ( $m$  è il numero di vincoli,  $n$  la dimensione del problema). In questo caso notiamo che le direzioni ammissibili e quelle del cono tangente

coincidono, e corrispondono alle  $d$  tali per cui  $Ad = 0$ : infatti se  $Ax = b$ ,  $A(x + d) = Ax + Ad = Ax = b$ . In particolare se  $d \in F_X$ , anche  $-d \in F_X$ .

Quindi la condizione (TCC) diventa  $\nabla f(x) \cdot d = 0$  per ogni  $d \in F_X(x)$ . Poiché  $F_X(x) = \ker A$ , si ha che  $\nabla f(x) \in (\ker A)^\perp = \text{im}(A^T)$ , cioè esiste un  $\mu \in \mathbb{R}^m$  tale per cui  $\nabla f(x) = \mu A$ . Le condizioni di ottimalità diventano perciò:

$$Ax = b, \text{ e esiste } \mu \in \mathbb{R}^m \text{ t.c. } \nabla f(x) = \mu A \quad (3.4)$$

$\mu$  è il primo esempio di *variabile duale*: siamo passati da un "per ogni" nel (TCC) originale, ad un "esiste": per verificare che  $x$  sia ottimale basta trovare  $\mu$ .

Passiamo ora al caso di generiche equazioni non lineari  $g_i(x) \leq 0$ ,  $i \in \mathcal{I}$ . Chiamiamo *vincoli attivi* di  $x \in X$  i vincoli per cui  $g_i(x) = 0$ , cioè

$$\mathcal{A}(x) = \{i \in \mathcal{I} : g_i(x) = 0\}$$

Questi vincoli sono gli unici che davvero ci interessano quando ci troviamo in  $x$ : il cono tangente è un concetto locale,  $T_X(x) = T_{X \cap B(x,r)}(x)$ , e se per qualche  $i$   $g_i(x) < 0$ , tale vincolo non ha impatto su  $T_X(x)$ . Si veda la Figura 3.1: il cono dipende solo dai vincoli che definiscono il punto angoloso in cui si trova  $x$ . Chiamiamo inoltre cono delle direzioni ammissibili al primo ordine

$$D_X(x) = \{d : \nabla g_i(x) \cdot d \leq 0 \text{ } i \in \mathcal{A}(x), \nabla h_j(x) \cdot d = 0 \text{ } j \in \mathcal{J}\}$$

Si può dimostrare che  $D_X \supseteq T_X$  per ogni  $x$ . In generale non vale l'uguaglianza, ma in molti casi sì: si parla di *qualifica dei vincoli*, cioè di trovare condizioni per cui vale  $D_X = T_X$ . Alcuni esempi tipici sono:

**Vincoli affini:** se le  $g_i$  e le  $h_j$  sono affini,  $D_X(x) = T_X(x)$  per ogni  $x \in X$

**Slater:** se le  $g_i$  sono convesse, le  $h_j$  sono affini e vale che esiste  $\bar{x}$  tale che  $g_i(\bar{x}) < 0$ , allora  $D_X(x) = T_X(x)$  per ogni  $x \in X$

**Lineare indipendenza dei gradienti:** se per un  $\bar{x} \in X$  vale che i vettori  $\{(\nabla g_i(\bar{x}))_{i \in \mathcal{A}(\bar{x})}, (\nabla h_j(\bar{x}))_{j \in \mathcal{J}}\}$  sono linearmente indipendenti, allora  $D_X(\bar{x}) = T_X(\bar{x})$

Nel caso in cui valga  $D_X = T_X$  possiamo quindi controllare (TCC) usando la definizione di  $D_X$ . Per fare ciò useremo il lemma di Farkas. Prima di enunciarlo diamo una definizione:

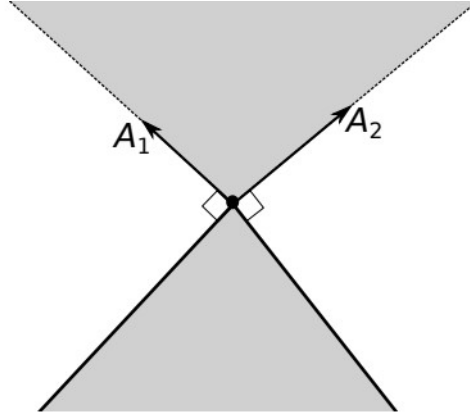


Figura 3.3: Il cono  $\mathcal{C}$  (in basso) e il suo duale  $\mathcal{C}^*$  (in alto).  $\mathcal{C}^*$  è la combinazione lineare con coefficienti non negativi di  $A_1$  e  $A_2$ , che sono i vettori che definiscono  $\mathcal{C}$

**Definizione.**  $\mathcal{C}$  è un cono poliedrico se si può scrivere come

$$\mathcal{C} = \{d : Ad \leq 0\}$$

per qualche  $A \in \mathbb{R}^{k \times n}$

Si vede facilmente che  $D_X(x)$  è un cono poliedrico: basta prendere la matrice

$$A = \begin{pmatrix} (\nabla g_i(x))_{i \in \mathcal{A}(x)} \\ (\nabla h_j(x))_{j \in \mathcal{J}} \\ (-\nabla h_j(x))_{j \in \mathcal{J}} \end{pmatrix}. \quad (3.5)$$

Dato un qualunque cono poliedrico  $\mathcal{C}$  possiamo definire il cono duale  $\mathcal{C}^*$  come

$$\mathcal{C}^* = \{c = \sum_{i=1}^k \lambda_i A_i : \lambda \geq 0\}$$

Possiamo enunciare il lemma di Farkas:

**Lemma. (di Farkas):** Per ogni  $c \in \mathbb{R}^n$  vale sempre una e una sola delle seguenti condizioni:

- o esiste  $\lambda \in \mathbb{R}^k$ ,  $\lambda \geq 0$ , tale che  $c = \sum_{i=1}^k \lambda_i A_i$
- oppure esiste  $d$  tale che  $Ad \leq 0$  e  $c \cdot d > 0$

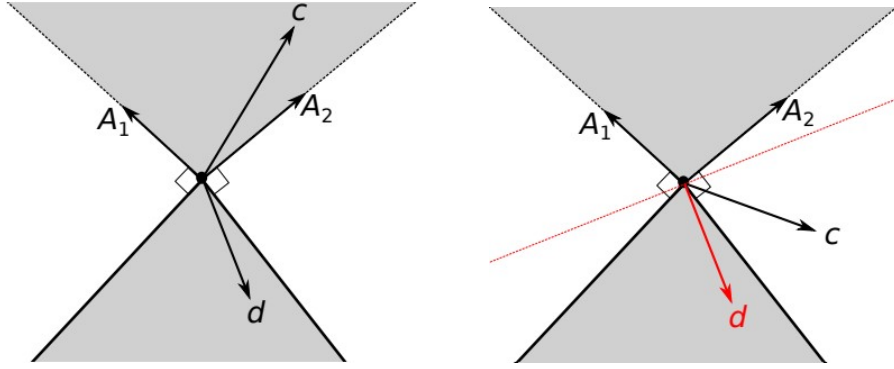


Figura 3.4: Il cono  $\mathcal{C}^*$  è composto da tutte e sole le direzioni che formano un angolo maggiore di 90 gradi con ogni vettore del cono  $\mathcal{C}$ ; a sinistra siamo nel primo caso dell'enunciato del Lemma di Farkas, cioè  $c \in \mathcal{C}^*$ ; a destra nel secondo caso, cioè  $c \notin \mathcal{C}^*$  e  $c \cdot d > 0$ .

*In altre parole dato un cono poliedrico  $\mathcal{C}$ , o  $c \in \mathcal{C}^*$ , oppure il prodotto scalare con uno degli elementi di  $\mathcal{C}$  è positivo, e le due condizioni sono mutualmente esclusive.*

Nella figura 3.4 si dà un'idea della dimostrazione: possiamo definire il cono polare

$$\mathcal{C}^\circ = \{c : \forall d \in \mathcal{C} \ c \cdot d \leq 0\},$$

e vedere che  $\mathcal{C}^\circ = \mathcal{C}^*$ . Il lemma di Farkas dice quindi “ $c \in \mathcal{C}^*$  oppure  $c \notin \mathcal{C}^*$ ”

### 3.2.4 Condizioni di Karush–Kuhn–Tucker

Dal Lemma di Farkas è facile arrivare a delle condizioni che ci permettano di verificare facilmente l'ottimalità. Supponiamo che (P) sia convesso e che  $D_X(x) = T_X(x)$ . In queste ipotesi per verificare l'ottimalità globale di  $x$ , basta verificare (TCC) su  $D_X$ , ovvero

$$\nabla f(x) \cdot d \geq 0 \text{ per ogni } d \text{ t.c. } \nabla g_i(x) \cdot d \leq 0 \ i \in \mathcal{A}(x), \nabla h_j(x) \cdot d = 0 \ j \in \mathcal{J} \quad (3.6)$$

Per verificare (3.6) usiamo il Lemma di Farkas con  $\mathcal{C} = D_X$ ,  $A$  come definita in (3.5) e  $c = -\nabla f(x)$ : per il lemma, (3.6) vale se e solo se esistono  $\lambda \in \mathbb{R}_+^{|\mathcal{A}(x)|}$

e  $\mu', \mu'' \in \mathbb{R}_+^{|\mathcal{J}|}$  tali che

$$\nabla f(x) + \sum_{i \in \mathcal{A}(x)} \lambda_i \nabla g_i(x) + \sum_{j \in \mathcal{J}} \mu'_j \nabla h_j(x) - \sum_{j \in \mathcal{J}} \mu''_j \nabla h_j(x) = 0$$

chiamando  $\mu = \mu' - \mu''$ , otteniamo che  $x$  è ottimale se esistono  $\lambda \in \mathbb{R}_+^{|\mathcal{A}(x)|}$  e  $\mu \in \mathbb{R}^{|\mathcal{J}|}$  ( $\mu$  non necessariamente positivo) tali che

$$\nabla f(x) + \sum_{i \in \mathcal{A}(x)} \lambda_i \nabla g_i(x) + \sum_{j \in \mathcal{J}} \mu_j \nabla h_j(x) = 0 \quad (3.7)$$

Le equazioni di Karush–Kuhn–Tucker sono molto simili all'equazione (3.7) e sono ad essa equivalenti:

**Condizioni di Karush-Kuhn-Tucker (KKT):** Esistono  $\lambda \in \mathbb{R}^{|\mathcal{I}|}$  e  $\mu \in \mathbb{R}^{|\mathcal{J}|}$  tali che

$$g_i(x) \leq 0 \quad i \in \mathcal{I}, \quad h_j(x) = 0 \quad j \in \mathcal{J} \quad (3.8)$$

$$\nabla f(x) + \sum_{i \in \mathcal{I}} \lambda_i \nabla g_i(x) + \sum_{j \in \mathcal{J}} \mu_j \nabla h_j(x) = 0 \quad (3.9)$$

$$\sum_{i \in \mathcal{I}} \lambda_i (-g_i(x)) = 0 \quad (3.10)$$

Analizziamo le 3 equazioni:

- La (3.8) ci dice semplicemente che  $x$  è una soluzione ammissibile.
- La (3.9) corrisponde alla (3.7), ma con tutto  $\mathcal{I}$  al posto dell'insieme dei vincoli attivi  $\mathcal{A}(x)$ . Perciò per rendere equivalenti le due equazioni ci serve la terza equazione.
- la (3.10), detta *complementary slackness*, ci garantisce che se  $i \notin \mathcal{A}(x)$ , allora  $\lambda_i = 0$ : infatti la sommatoria è composta di termini tutti maggiori o uguali a 0, perciò, se vogliamo che sia nulla, ogni termine dev'essere esattamente 0. Se  $i \notin \mathcal{A}(x)$ , allora  $-g_i(x) > 0$ , e dunque per forza  $\lambda_i = 0$ .

*Osservazione 8.* La riscrittura dell'equazione (3.7) nelle equazioni KKT è utile dal punto di vista pratico perché nella KKT il vettore  $\lambda$  ha sempre la stessa dimensione, mentre nella (3.7) dipende da quanti vincoli sono attivi e quindi da  $x$ .

Esattamente come (TCC), le (KKT) sono solo necessarie perché  $x$  sia ottimo, ma in generale non sufficienti (non riescono a distinguere massimi da minimi da punti di sella).

Tuttavia, se (P) è convesso, allora sono necessarie e sufficienti per l'ottimalità: se  $x_*$  soddisfa (KKT),  $\nabla f(x) \cdot d \geq 0$  per ogni  $d \in D_X(x_*)$ , e  $D_X \supseteq T_X \supseteq F_X$ , quindi  $\nabla f(x) \cdot d \geq 0$  per ogni  $d \in F_X(x_*)$ , cioè  $x_*$  è ottimo globale.

### 3.3 Dualità Lagrangiana

#### 3.3.1 Funzione Lagrangiana e dualità

Riprendiamo la condizione (KKT):

$$\nabla f(x) + \sum_{i \in \mathcal{A}(x)} \lambda_i \nabla g_i(x) + \sum_{j \in \mathcal{J}} \mu_j \nabla h_j(x) = 0$$

Notiamo che l'equazione è equivalente a  $\nabla_x L(x, \lambda, \mu) = 0$ , dove

$$L(x, \lambda, \mu) = f(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x) + \sum_{j \in \mathcal{J}} \mu_j h_j(x) \quad (3.11)$$

e  $\nabla_x$  indica il gradiente ottenuto derivando solo nella variabile  $x$ .

**Definizione.**  $L(\cdot)$  è detta *funzione lagrangiana* del problema (P).

Cioè  $x_*$  è ottimo se è un punto stazionario di  $L$  per i giusti  $\lambda$  e  $\mu$  fissati. Ci chiediamo perciò come trovare  $x_*$  conoscendo i giusti  $\lambda$  e  $\mu$ : l'idea è di risolvere il *rilassamento lagrangiano*

$$(R_{\lambda, \mu}) \quad \psi(\lambda, \mu) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu)$$

In generale chiamiamo un rilassamento ( $\underline{P}$ ) di (P) è un altro problema di ottimizzazione

$$(\underline{P}) \quad \min \{ \underline{f}(x), x \in \underline{X} \}$$

con

i)  $\underline{f}(x) \leq f(x)$  per ogni  $x \in X$

ii)  $\underline{X} \supseteq X$ .



In questo modo è ovvio che  $v(\underline{P}) \leq v(P)$ , cioè  $(\underline{P})$  ci fornisce un lower bound al valore ottimo di  $(P)$ .

**Lemma 3.3.1.** *Per ogni  $\lambda \in \mathbb{R}_+^{|\mathcal{I}|}$ ,  $\mu \in \mathbb{R}^{|\mathcal{J}|}$ ,  $(R_{\lambda,\mu})$  è un rilassamento di  $(P)$ .*

*Dimostrazione.* La regione ammissibile di  $(R_{\lambda,\mu})$  è tutto  $\mathbb{R}^n$ , dunque vale ii).

Vediamo che vale i): se  $x$  è ammissibile per  $(P)$ ,  $h_j(x) = 0$  per ogni  $j \in \mathcal{J}$  e  $g_i(x) \leq 0$  per ogni  $i \in \mathcal{I}$ . Inoltre  $\lambda \geq 0$ , perciò  $\sum_j \mu_j h_j(x) = 0$  e  $\sum_i \lambda_i g_i(x) \leq 0$ , perciò  $L(x, \lambda, \mu) \leq f(x)$  qualsiasi siano  $\lambda \geq 0$  e  $\mu$ .  $\square$

Il Lemma 3.3.1 ci porta al concetto di dualità debole:

**Dualità debole:** Per ogni  $\lambda \geq 0, \mu$  e  $x$  vale  $\psi(\lambda, \mu) \leq v(P) \leq f(x)$ .

In altre parole, possiamo definire il *problema duale*

$$(D) \quad \max \psi(\lambda, \mu) \quad \lambda \in \mathbb{R}_+^{|\mathcal{I}|}, \mu \in \mathbb{R}^{|\mathcal{J}|} \quad (3.12)$$

La dualità debole ci dice che vale sempre  $v(D) \leq v(P)$ .

$(D)$  è spesso un problema non troppo difficile da risolvere, anche se  $(P)$  fosse difficile:

- Per calcolare  $\psi(\lambda, \mu)$  bisogna risolvere  $(R_{\lambda,\mu})$  che è un problema di ottimizzazione non vincolato.
- $\psi$  è concava: è il massimo di (infinite) funzioni lineari in  $\lambda$  e  $\mu$ . (la “non linearità” di  $L$  è solo in  $x$ ), perciò trovarne il massimo è facile (equivale a trovar e il minimo di una funzione convessa). Tuttavia in alcuni casi  $\psi = -\infty$
- In generale  $\psi \notin C^1$ , anche se  $f, g_i, h_j \in C^1$ , ma vale che se  $\bar{x}$  è una soluzione ottima di  $R_{\lambda,\mu}$ , allora il vettore  $(G(\bar{x}), H(\bar{x})) \in \partial\psi(\lambda, \mu)$
- In particolare se  $\bar{x}$  nel punto sopra è l’unica soluzione ottima di  $(R_{\lambda,\mu})$ , si dimostra che  $\psi$  è differenziabile in  $(\lambda, \mu)$ , e  $\nabla\psi(\lambda, \mu) = (G(\bar{x}), H(\bar{x}))$

Queste proprietà ci portano a concludere che  $\psi$  è spesso facile da massimizzare, dunque abbiamo trasformato un problema  $(P)$  in un altro problema  $(D)$  quasi non vincolato (il vincolo  $\lambda \geq 0$  è facile) che ci dà un valore  $v(D) \leq v(P)$ . Allo stesso tempo, per calcolare  $\psi(\lambda, \mu)$  bisogna risolvere un problema di minimizzazione non necessariamente convesso, che può essere in alcuni casi difficile.

Idealmente vorremmo che valga la **dualità forte**, ovvero

$$v(D) = v(P)$$

In generale questa proprietà non vale, ma vale se  $(P)$  è convesso.

### 3.3.2 Duali speciali

#### Problemi lineari

Nel caso di problemi lineari

$$(P) \quad \min c^T x \quad Ax \geq b,$$

la funzione lagrangiana diventa

$$L(x, \mu) = c^T x + \lambda^T (b - Ax) = \lambda^T b + (c^T - \lambda^T A)x.$$

Perciò il rilassamento lagrangiano è

$$(R_\lambda) \quad \lambda^T b + \min_{x \in \mathbb{R}^n} (c^T - \lambda^T A)x$$

che può essere risolto in formula chiusa:

$$\psi(\lambda) = \begin{cases} -\infty & \text{se } c^T - \lambda^T A \neq 0 \\ \lambda^T b & \text{altrimenti} \end{cases}$$

Il duale è quindi

$$(D) \quad \max \psi \quad \lambda^T A = c^T, \lambda \geq 0.$$

Vale la dualità forte, a meno che entrambe le regioni ammissibili non siano vuote (in questo caso  $v(D) = -\infty < v(P) = +\infty$ ).

#### Problemi quadratici

Iniziamo da un problema quadratico molto semplice:

$$(P) \quad \frac{1}{2} \min \|x\|^2 \quad Ax = b.$$

In questo caso

$$L(w, \mu) = \left[ \frac{1}{2} \|x\|^2 + \mu^T Ax \right] - \mu^T b. \quad (3.13)$$

Il minimo in  $x$  della lagrangiana si ha quando  $\nabla_x L(x, \mu) = x + (\mu^T A)^T = 0$ . Se sostituiamo  $x = -(\mu^T A)^T = -A^T \mu$  in (3.13) otteniamo che

$$\psi(\mu) = -\frac{1}{2} \mu^T (AA^T) \mu - \mu^T b,$$

dunque il problema duale diventa

$$(D) \quad \max -\frac{1}{2} \mu^T (AA^T) \mu - \mu^T b \quad \mu \in \mathbb{R}^m$$

Che è un problema quadratico non vincolato.

Analizziamo ora un caso più generale:

$$(P) \quad \min \frac{1}{2} x^T Q x + q^T x \quad Ax \geq b$$

con  $Q$  definita positiva. La lagrangiana è

$$L(x, \lambda) = \frac{1}{2} x^T Q x + q^T x + \lambda^T (b - Ax) = \lambda^T b + \left[ \frac{1}{2} x^T Q x + (q^T - \lambda^T A) x \right],$$

il cui minimo è quando  $\nabla_x L = Qx + q - (\lambda^T A)^T = 0$  e quindi quando  $x = Q^{-1}((\lambda^T A)^T - q)$ . Definiamo  $v = (\lambda^T A)^T - q$ , e quindi  $x = Q^{-1}v$ , e otteniamo che

$$\psi(\lambda) = \frac{1}{2} v^T Q^{-1} v - v^T Q^{-1} v + \lambda^T b = -\frac{1}{2} v^T Q^{-1} v + \lambda^T b.$$

Dunque

$$(D) \quad \max -\frac{1}{2} v^T Q^{-1} v + \lambda^T b \quad \lambda^T A - v = q, \lambda \geq 0$$

### 3.4 Algoritmi per l'ottimizzazione vincolata

Per la trattazione di questi algoritmi, ci restringeremo al caso di problemi quadratici, ovvero dove  $f(x) = \frac{1}{2} x^T Q x + q^T x$ , e l'insieme delle regioni ammissibili è  $X = \{x : Ax \leq b\}$ .

Un'importante notazione che useremo è la seguente: dati  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ , e un sottoinsieme  $B \subseteq 1, \dots, m$ , denotiamo con  $A_B$  la matrice  $(A_{ij})_{i \in B}$ , e con  $b_B$  il vettore  $(b_i)_{i \in B}$ . Con questa notazione il cono tangente in un punto  $x$  è

$$T_X(x) = F_X(x) = \{d \in \mathbb{R}^n : A_{\mathcal{A}(x)}d \leq 0\}$$

Prima di passare al primo algoritmo vero e proprio, trattiamo un caso più semplice.

### Problemi quadratici con vincoli di uguaglianza

$$(P) \quad \min \left\{ \frac{1}{2} x^T Q x + q x : A x = b \right\}$$

In questo caso, se  $Q \succeq 0$ , il problema è convesso, e per risolverlo basta risolvere il sistema dato dalle (KKT)

$$Qx + A^T \mu + q = 0 \tag{3.14}$$

$$Ax = b \tag{3.15}$$

che in forma matriciale è

$$\begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ \mu \end{pmatrix} = \begin{pmatrix} -q \\ b \end{pmatrix}.$$

Questo sistema appare in molti contesti diversi e va risolto efficientemente (vedasi Poloni).

#### 3.4.1 Metodo Active Set per problemi quadratici

Come visto nella sezione precedente, se sapessimo l'insieme di vincoli attivi della soluzione ottimale  $x_*$ , basterebbe risolvere un problema lineare. Ovviamente non sapendo qual è  $x_*$ , non conosciamo neanche  $\mathcal{A}(x_*)$ , perciò un'idea è quella di stimarlo:

- Dato  $x$ , prendiamo  $B = \mathcal{A}(x)$ , e risolviamo il problema quadratico ristretto a  $B$

$$\min \left\{ \frac{1}{2} x^T Q x : A_B x = b_B \right\}$$

Così facendo otteniamo una soluzione  $(\bar{x}, \bar{\mu}_B)$

- Non sappiamo se  $\bar{x}$  soddisfi tutti gli altri vincoli, cioè se dato il complementare  $B^C$  di  $B$ , valga  $A_{B^C} \bar{x} \leq b_{B^C}$ :
  - Se vale  $A_{B^C} \bar{x} \leq b_{B^C}$ 
    - \* se vale anche  $\mu_B \geq 0$ , allora  $\bar{x}$  è ottimale, perché soddisfa le condizioni KKT;
    - \* se invece  $\mu_B \not\geq 0$ , prendiamo il minimo indice  $h$  tale per cui  $(\bar{\mu}_B)_h < 0$ , e ricominciamo con  $B \setminus \{h\}$
  - Se invece  $\bar{x}$  non soddisfa tutti gli altri vincoli, comunque  $d = \bar{x} - x$  è una direzione di decrescita: prendiamo perciò

$$\alpha = \min \left\{ \alpha_i = \frac{b_i - A_i x}{A_i d} : A_i d > 0, i \notin B \right\}$$

E ricominciamo con  $x + \alpha d$  come nuovo punto, aggiornando  $B$  di conseguenza.

**Lemma 3.4.1.**  $\alpha$  appena definito è il massimo passo che possiamo fare per cui  $x + \alpha d$  è ancora una soluzione ammissibile. Inoltre  $\alpha < +\infty$ .

*Dimostrazione.* Ricordiamo che  $x + \alpha d$  è ammissibile se e solo se, per ogni  $i$ ,  $A_i(x + \alpha d) \leq b_i$ .

Se  $i \in B$ ,  $A_i(x + \alpha d) = A_i((1 - \alpha)x + \alpha \bar{x}) = (1 - \alpha)b_i + \alpha b_i = b_i$ .

Se  $A_i d \leq 0$ , chiaramente non c'è alcun problema.

Se invece  $A_i d > 0$  ha che se  $x + \alpha' d$  è ammissibile, dev'essere  $b_i - A_i x - \alpha' A_i d \geq 0$ , ovvero  $\alpha' \leq \alpha_i = \frac{b_i - A_i x}{A_i d}$ . Ciò deve valere per ogni  $i$  per cui vale  $A_i d > 0$ , perciò  $\alpha'$  dev'essere minore o uguale del minimo tra gli  $\alpha_i$ .

Si ha inoltre che  $\alpha < 1$ , perché  $x + d = \bar{x}$  non è ammissibile, perciò viola un qualche vincolo  $\bar{i}$ , dunque in particolare  $\alpha_{\bar{i}} < 1$ .  $\square$

Si dimostra che il metodo Active Set finisce in un numero finito di passi: si può vedere infatti che la funzione obiettivo diminuisce strettamente per ogni iterazione, quindi non viene mai ripetuto l'insieme dei vincoli attivi  $B$ . Il problema è che nel caso sfortunato potremmo analizzare ogni  $B$  possibile, e i sottoinsiemi possibili di  $\{1, \dots, m\}$  sono  $2^m$ .

### Metodo nel caso di box constraint

. Un caso particolare in cui il calcolo di  $\alpha$  è particolarmente semplice è quello di *box constraint* ovvero vincoli della forma  $\underline{x} \leq x \leq \bar{x}$ . Chiamiamo

$N = \{1, \dots, n\}$ , dove  $n$  è la dimensione di  $x$ . In questo caso l'insieme dei vincoli attivi  $B$  si può dividere in una partizione  $U \cup L$ , dove  $l \in L$  se  $x_l = \underline{x}_l$ , e  $u \in U$  se  $x_u = \underline{x}_u$ . Chiamiamo  $F = N \setminus (L \cup U)$ , l'insieme dei vincoli liberi. La condizione  $A_B = b_B$  ci dice che  $x = (x_L, x_F, x_U) = (\underline{x}_L, x_F, \bar{x}_U)$ , cioè il solo vettore  $x_F$  è libero, mentre gli altri due sono vincolati a essere uguali a  $\underline{x}_L$  e  $\bar{x}_U$ .

Se assumiamo  $\underline{x} = 0$  (possiamo prendendo  $x \leftarrow x + \underline{x}$ ),  $x = (0, x_F, \bar{x}_U)$  e quindi il problema da risolvere nell'iterazione dell'algoritmo diventa

$$\min \left\{ \frac{1}{2} x_F^T Q_{FF} x_F + (q_F + \bar{x}_U^T Q_{UF}) x_F \right\},$$

che è un problema non vincolato in dimensione minore.  $Q_{UF}$  denota la sottomatrice di  $Q$  ottenuta prendendo le righe e con indici in  $U$  e le colonne con indici in  $F$ . Analogamente  $Q_{FF}$ .

### 3.4.2 Gradiente proiettato

Mettiamoci ora nel caso di un problema dove la funzione obiettivo  $f \in C^1$  è generica e i vincoli sono lineari di uguaglianza. Dato  $x \in X$ , una direzione che sappiamo essere di decrescita per  $f$  è  $-\nabla f(x)$ . Se però  $-\nabla f(x)$  non è una direzione ammissibile, vogliamo prendere  $d$  ammissibile “simile” a  $-\nabla f(x)$ : la soluzione è prendere la *proiezione* dell'antigradiente sul cono delle soluzioni ammissibili  $F(x)$ . Cerchiamo dunque

$$p_{F(x)}(-\nabla f(x)) = \min \{ h(d) = \frac{1}{2} \|d + \nabla f(x)\|^2 : A_{\mathcal{A}(x)} d \leq 0 \}. \quad (3.16)$$

Per ogni iterazione dobbiamo dunque risolvere un problema quadratico su un constraint di cono poliedrico, ad esempio usando il metodo active set che abbiamo visto nella sezione precedente.

**Lemma 3.4.2.** *Se  $p_{F(x)}(-\nabla f(x)) = 0$ , allora  $\lambda A x + \nabla f(x) = 0$ , cioè valgono le (KKT) e quindi  $x$  è ottimale*

*Dimostrazione.*  $\nabla h(d) = d + \nabla f(x)$ , e quindi in particolare  $\nabla h(0) = \nabla f(x)$ . Dunque, se  $p_{F(x)}(-\nabla f(x)) = \nabla f(x)$ , usando la (KKT) (3.9) applicate al problema (3.16), si ottiene che esiste  $\lambda' \in \mathbb{R}^{|\mathcal{A}(x)|}$  tale per cui

$$\nabla h(0) + \lambda' A_{\mathcal{A}(x)} = \nabla f(x) = 0.$$

Prendendo  $\lambda = \lambda'$  negli indici di  $\mathcal{A}(x)$  e  $\lambda = 0$  negli altri indici, otteniamo che  $\nabla f(x) + \lambda A = 0$ , cioè vale la condizione (3.9) per il problema originario. Per costruzione valgono ovviamente anche le altre due condizioni (KKT), perciò  $x$  è ottimale.  $\square$

Grazie al lemma 3.4.2, siamo sicuri che la condizione  $\|d\| < \epsilon$  sia un buon criterio di stop per il metodo iterativo

### Metodo nel caso di box constraint

Il problema (3.16) può essere costoso da risolvere, ma nel caso di box constraint  $\underline{x} \leq x \leq \bar{x}$ , è molto semplice. Possiamo decomporre  $X = X_1 \times \dots \times X_n$ , con  $X_i = [\underline{x}, \bar{x}]$ , e di conseguenza decomporre anche  $F(x) = F(x)_1 \times \dots \times F(x)_n$ , con

$$F(x)_i = \begin{cases} \mathbb{R}_{\geq 0} & x_i = \underline{x}_i \\ \mathbb{R}_{\leq 0} & x_i = \bar{x}_i \\ \mathbb{R} & x_i \in (\underline{x}_i, \bar{x}_i) \end{cases}$$

Cioè se  $x_i = \underline{x}_i$ ,  $x_i$  può solo aumentare, e analogamente se  $x_i = \bar{x}_i$ , può solo diminuire, mentre se  $x_i$  è “libero” possiamo farlo variare in entrambe le direzioni. Possiamo perciò minimizzare  $\|d_i + \nabla f(x)_i\|$  con  $d_i \in F(x)_i$  uno a uno ottenendo  $n$  problemi molto più semplici e parallelizzabili.

### Gradiente proiettato di Goldstein

Una variante del gradiente proiettato è quella in cui “prima muoviamo, poi proiettiamo”. Se al passo  $i$  ci troviamo in  $x^i$ , calcoliamo

$$\begin{aligned} y^i &= x^i + \alpha \nabla f(x^i) \\ x^{i+1} &= p_X(y^i) \end{aligned}$$

Questo metodo non ci garantisce che  $f(x^{i+1}) < f(x^i)$ , e in generale per la convergenza serve un passo appropriato, ad esempio, se  $f$  è L-lipschitz,  $\alpha = \frac{1}{L}$ . Tuttavia in alcuni casi si può garantire un risultato di convergenza simile a quello del metodo del gradiente nel caso non vincolato.

### Gradiente proiettato di Rosen

Se la proiezione su  $X$  è troppo costosa, possiamo proiettare sul bordo del cono ammissibile  $\partial F_X = \{d \in \mathbb{R}^n : A_{\mathcal{A}(x)}d = 0\}$ . La proiezione su questo insieme

è molto semplice: è un problema quadratico con vincoli di uguaglianza, e se  $\bar{A} = A_{\mathcal{A}(x)}$  ha rango massimo, abbiamo una formula chiusa per  $d$  è  $\mu$ :

$$\mu = -[\bar{A}\bar{A}^T]^{-1}\bar{A}\nabla f(x), \quad (3.17)$$

$$d = (I - \bar{A}^T[\bar{A}\bar{A}^T]^{-1}\bar{A})(-\nabla f(x)) \quad (3.18)$$

### 3.4.3 Metodo di Frank-Wolfe

Nel gradiente proiettato dobbiamo ad ogni passo risolvere un problema quadratico con vincoli rappresentati da  $F_X$ . Nel metodo di Frank-Wolfe o del gradiente condizionale, ad ogni passo risolviamo un sotto-problema in cui la funzione obbiettivo è lineare:

$$\bar{x}_i = \arg \min \{ \nabla f(x_i) \cdot x : Ax \leq b \}$$

Ad ogni passo minimizziamo quindi il prodotto scalare tra  $x$  e  $\nabla f(x_i)$ : in questo modo minimizziamo il modello del primo ordine in  $x_i$  all'interno di  $X$ . Detta  $d_i = \bar{x}_i - x_i$ , prendiamo  $x_{i+1} \leftarrow x_i + \alpha_i d_i$  (con  $\alpha_i \in [0, 1]$  risultato di una line-search lungo  $d_i$ ). Notiamo che per convessità di  $X$ ,  $x_{i+1} \in X$ .

**Lemma 3.4.3.** <sup>1</sup> Se  $\nabla f(x) \cdot d = 0$ , allora  $x$  è un ottimo locale.

**Lemma 3.4.4.** Se  $\nabla f(x) \cdot d \neq 0$ , allora  $\nabla f(x) \cdot d < 0$ , cioè  $d$  è una direzione di discesa.

*Dimostrazione.*  $\bar{x}$  è il minimo di  $L_x(y) = \nabla f(x) \cdot y$ , perciò in particolare  $\nabla f(x) \cdot \bar{x} \leq \nabla f(x) \cdot x$ , e dunque  $\nabla f(x) \cdot (\bar{x} - x) = \nabla f(x) \cdot d \leq 0$ . Poiché non vale l'uguaglianza, otteniamo la tesi.  $\square$

Abbiamo quindi il classico criterio di stop  $\nabla f(x) \cdot d \geq -\varepsilon$ . Possiamo però dire di più sul metodo nel caso di funzioni convesse: chiamiamo

$$v^* = f(x) + \nabla f(x) \cdot \bar{x} - x = L_x(\bar{x}).$$

Se  $f$  è convessa, il modello del primo ordine approssima dal basso  $f$ , cioè  $L_x(y) \leq f(y)$  per ogni  $x, y$ . Ciò vale in particolare se  $y = x^*$  è l'ottimo: per ogni  $x$  vale dunque

$$\begin{aligned} v(P) &= \min \{ f(y) : y \in X \} \geq \min \{ L_x(y) : y \in X \} \\ &= f(x) + \min \{ \nabla f(x) \cdot y : y \in X \} = f(x) + \nabla f(x) \cdot \bar{x} = v^*. \end{aligned}$$

---

<sup>1</sup>Da qui in poi ometto gli indici  $i$  per semplicità di notazione.



Ciò ci dice che  $v^*$  fornisce un *lower bound* del valore ottimo della funzione. Possiamo perciò per la prima volta stimare il gap tra  $f(x)$  e  $v(P)$ :

$$f(x) - v^* \geq f(x) - v(P).$$

In particolare, se teniamo in memoria il miglior lower bound  $l$  (quindi il *massimo*  $v^*$  per ogni iterazione) abbiamo un nuovo criterio di stop:  $f(x_i) - l \geq \varepsilon$

### Metodo nel caso di box-constraint

Nel caso di Box constraint  $X = \{x : \underline{x} \leq x \leq \bar{x}\}$ , il problema lineare è risolvibile in modo immediato: basta prendere per ogni componente  $k$

$$\bar{x}_k = \begin{cases} \underline{x}_k & \text{se } \nabla f(x)_k \geq 0 \\ \bar{x}_k & \text{se } \nabla f(x)_k < 0 \end{cases}$$

### Stabilizzazione del metodo di Frank Wolfe

La convergenza del metodo di Frank Wolfe risulta essere piuttosto lenta: ad ogni passo cerchiamo  $\bar{x}$  che minimizza il modello del primo ordine  $L_x$  dentro tutto l'insieme ammissibile  $X$ . In altre parole ci fidiamo del modello anche molto lontano da  $x$ , dove non abbiamo garanzie che il modello stesso sia buono. Per ovviare a questo problema possiamo provare a regolarizzare il problema lineare, aggiungendo un termine  $\gamma\|x - y\|_2^2$  a  $L_x(y)$ . In questo modo però trasformiamo il sottoproblema in un altro problema quadratico.

Un'altra soluzione è quella della stabilizzazione tramite regioni di fiducia: possiamo aggiungere un termine  $\|x - y\|_\infty \leq \tau$  ai vincoli del problema, o in altre parole dei box constraint  $x_i - \tau \leq y_i \leq x_i + \tau$ . Il problema rimane così lineare, e i vincoli che aggiungiamo non rendono quasi mai il problema più difficile.

### 3.4.4 Metodo duale

Fino ad ora abbiamo analizzato metodi che si basavano sul problema primale, ottenendo ad ogni passo soluzioni primali  $x^k$  ammissibili, e trovando solo una volta raggiunto l'ottimo variabili duali  $\lambda \geq 0$  ammissibili. Un approccio alternativo è invece considerare direttamente il duale: prendiamo

$$\psi(\lambda) = \min\left\{\frac{1}{2}x^T Qx + q^T x + \lambda(b - Ax)\right\},$$

Abbiamo visto che se  $Q$  è definita positiva, l'ottimo si ha in  $x(\lambda) = Q^{-1}(\lambda A - q)$ . Possiamo allora risolvere il problema duale

$$(D) \quad \max\{\psi(\lambda) : \lambda \geq 0\}.$$

In generale,  $\psi \in C^1$ , ma  $\psi \notin C^2$ , perciò  $(D)$  può essere risolto con qualsiasi metodo adatto a funzioni  $C^1$ . Il problema di questo metodo è che, dato  $\lambda_i$ ,  $x(\lambda_i)$  è ammissibile per  $(P)$  se e solo se sia  $x(\lambda_i)$  che  $\lambda_i$  sono ottimali per i rispettivi problemi, quindi non abbiamo a disposizione nessuna soluzione ammissibile per  $(P)$ , finchè non troviamo l'ottimo. Tuttavia se i vincoli di  $(P)$  sono abbastanza semplici, possiamo calcolare *l'euristica lagrangiana*, ovvero la proiezione

$$x_i = p_X(x(\lambda_i)) = \min\{\|x - x(\lambda_i)\| : x \in X\}$$

### 3.4.5 Barrier Methods

TO DO