Ottimizzazione Sparsa

Matteo Lapucci
matteo.lapucci@unifi.it

Global Optimization Laboratory, Università degli Studi di Firenze

1 Introduzione

In molti contesti applicativi capita che, tra le varie proprietà desiderate per la soluzione di un problema di ottimizzazione, venga richiesto che questa abbia una struttura *sparsa*.

Il termine "sparso", in italiano, indica qualcosa di sparpagliato, distribuito un po' ovunque e non ordinato organicamente. Per questo, il termine tecnico di "sparsità", mutuato dall'inglese, può alle volte risultare poco chiaro. *Sparsità* viene infatti inteso come contrario di densità: intuitivamente, un insieme di oggetti è sparso se tra un elemento e l'altro c'è dello spazio vuoto; un vettore è sparso se ha poche componenti diverse da zero immerse tra molti elementi nulli.

Una volta chiarita la terminologia, possiamo passare alla domanda che a questo punto dovrebbe essere spontanea: perché si dovrebbe essere interessati a soluzioni sparse nei problemi di ottimizzazione? In realtà, ci sono diverse motivazioni, che si ritrovano in applicazioni abbastanza disparate. Di seguito, riportiamo alcuni tra i principali casi in cui la sparsità di una soluzione rappresenta un valore importante:

- In primo luogo, una soluzione sparsa ha il vantaggio di essere più facilmente *interpretabile* da un essere umano. Questo è utile per esempio in statistica, dove si può avere interesse a definire modelli che dipendano solo da poche variabili veramente rilevanti per descrivere il fenomeno studiato. Analogamente, nelle applicazioni in ambito medico è vantaggioso poter segnalare quali siano gli elementi realmente significativi per il decorso di una malattia.

- Vettori con una prevalenza di elementi nulli possono essere rappresentati in *memoria* in modo efficiente. Applicazioni importanti in cui è vitale risparmiare in modo ottimale unità informative sono quelle che riguardano la compressione e ricostruzione di immagini o la trasmissione di segnali (si veda la tecnica del *compressed sensing* in teoria dei segnali). Inoltre, un problema particolarmente attuale è quello che

riguarda l'addestramento di reti neurali profonde che utilizzino un numero di parametri relativamente contenuto, in modo da poter essere poi utilizzate in dispositivi con hardware limitato.

- Un beneficio duale a quello precedente riguarda l'*efficienza* con cui le soluzioni sparse possono essere utilizzate una volta messe "in funzione". Di fatto, utilizzare soluzioni sparse consente di effettuare calcoli in spazi a dimensionalità molto inferiore di quella di partenza, abbattendo i tempi di calcolo nel momento in cui il modello ottenuto viene utilizzato operativamente.

- In alcuni contesti, la sparsità è un *vincolo imposto dal mondo reale*. Può infatti essere possibile effettuare un numero limitato di operazioni, oppure apportare modifiche ad un numero limitato di componenti di un sistema. Si pensi ad esempio al problema della selezione del protafoglio: un investitore può effettuare, per legge, un numero limitato di operazioni finanziarie in un certo intervallo di tempo.

- In statistica e nell'apprendimento automatico, modelli che soddisfano opportuni requisiti di sparsità sono dimostrati avere importanti proprietà di *generalizzazione* ed aiutano a combattere il problema dell'overfitting.

Il concetto di sparsità può essere modellato in matematica tramite la "norma" zero, ℓ_0.

Definizione 1. Sia $x \in \mathbb{R}^n$. La *pseudo-norma zero* (o pseudo-norma ℓ_0) di x è definita da

$$\|x\|_0 = |\{i \mid x_i \neq 0,\ i = 1, \ldots, n\}|.$$

La pseudo-norma zero di un vettore indica il numero di componenti non nulle del vettore stesso. Vedremo in modo più rigoroso nella prossima sezione le proprietà di $\|x\|_0$. Prima però osserviamo come questo strumento consenta di modellare in modo opportuno il concetto di sparsità nei problemi di ottimizzazione.

Si possono definire tre formulazioni per problemi di ottimizzazione sparsa. Queste tre formulazioni sono, come specificato in seguito, teoricamente equivalenti, ma ognuna ha una sfumatura che la rende più o meno appropriata per determinate applicazioni.

Sia $f : \mathbb{R}^n \to \mathbb{R}$ una funzione continuamente differenziabile limitata inferiormente. Possiamo definire i tre seguenti problemi di ottimizzazione sparsa.

Formulazione I Problema con vincolo di cardinalità:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &\text{s.t. } \|x\|_0 \leq s, \end{aligned} \tag{1}$$

dove s è un intero tale che $s \leq n$.

Formulazione II Problema con regolarizzazione ℓ_0:

$$\min_{x \in \mathbb{R}^n} f(x) + \lambda \|x\|_0 \tag{2}$$

dove $\lambda \in \mathbb{R}_+$.

Formulazione III Problema di minimizzazione della cardinalità:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \|x\|_0 \\ &\text{s.t. } f(x) \leq \tau, \end{aligned} \tag{3}$$

dove $\tau \in \mathbb{R}_+$.

Si può dimostrare che le tre formulazioni precedenti sono equivalenti, nel senso specificato qui di seguito. Osserviamo che ciascuna delle tre formulazioni, dipende da un parametro (rispettivamente s, λ e τ). Consideriamo una qualsiasi coppia di formulazioni: si può dimostrare che per qualsiasi scelta del parametro per la prima formulazione esiste una soluzione ottimale $\bar{x}$ che è soluzione ottimale pure per la seconda formulazione per una opportuna scelta del parametro di quest'ultima.

Sebbene sappiamo che una opportuna scelta dei parametri renda le formulazioni equivalenti, passare da una formulazione all'altra è complicato perché la relazione esatta tra s, λ e τ che rende i problemi equivalenti non è nota.

In sostanza quindi, i parametri s, λ e τ definiscono, in modo diverso, il trade-off tra valore di f e sparsità. Pur avendo lo stesso scopo e pur essendo in un certo modo equivalenti, le tre formulazioni hanno sfumature diverse dal punto di vista applicativo, ed a seconda dei contesti una può essere più utile delle altre.

La prima formulazione è utile nei casi in cui è noto un limite al numero di variabili che possono essere considerate (es., selezione del portafoglio); la seconda formulazione modella casi in cui si vogliono bilanciare qualità e complessità (es., modelli di regressione interpretabili); la terza formulazione si ritrova quando abbiamo requisiti di qualità minima (es., compressione di immagini).

1.1 La Pseudo-Norma Zero

In questa sezione vediamo, in modo formale, le proprietà della norma zero. Per iniziare, richiamiamo le proprietà che definiscono una norma.

Definizione 2. Una norma $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_+$ è una funzione che verifica le seguenti condizioni:

(i) $\|x\| \geq 0$ per ogni $x \in \mathbb{R}^n$;

(ii) $\|x\| = 0$ se e solo se $x = 0$;

(iii) $\|kx\| = |k|\|x\|$ per ogni x e per ogni $k \in \mathbb{R}$ (omogeneità);

(iv) $\|x + y\| \leq \|x\| + \|y\|$ per ogni $x, y \in \mathbb{R}^n$ (disuguaglianza triangolare).

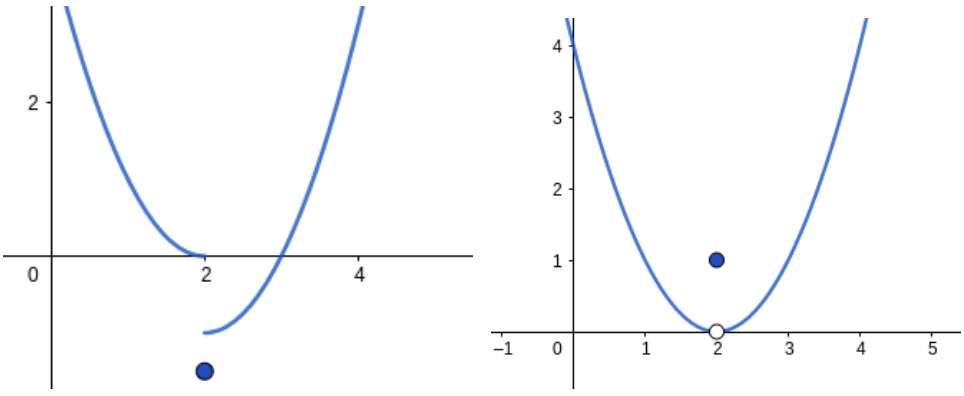
Alcuni esempi di norme sono:

- la norma ℓ_2, definita come $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$;
- la norma ℓ_1, definita come $\|x\|_1 = \sum_{i=1}^n |x_i|$;
- la norma ℓ_∞, definita come $\|x\|_\infty = \max\{|x_i| \mid i = 1, \ldots, n\}$;
- più in generale, la norma ℓ_p, per $p > 0$, è definita da $\|x\|_p = \left(\sum_{i=1}^n x_i^p\right)^{1/p}$.

Ogni norma ℓ_p, per $p \geq 1$ è una funzione convessa. È inoltre continuamente differenziabile per ogni $1 < p < \infty$. Infine, ogni norma ℓ_p è una funzione continua per ogni $p > 0$.

È possibile dimostrare che, per $p \to 0$, si ottiene la norma ℓ_0 definita in Definizione 1. Notiamo che la norma ℓ_0 non verifica la proprietà (iii), per questo è formalmente corretto parlare di *pseudo-norma*. È inoltre una funzione discontinua e fortemente nonconvessa.

L'unica proprietà di regolarità che si può attribuire alla pseudo norma zero è la *semicontinuità inferiore*, cioè, per ogni $x \in \mathbb{R}^n$ e per ogni $\varepsilon > 0$ esiste un intorno $U(x)$ tale che $f(x) < f(z) + \varepsilon$ per ogni $z \in U(x)$. La semicontinuità inferiore implica che, data una sequenze $\{x^k\} \subset \mathbb{R}^n$, vale $\|\lim_{k\to\infty} x^k\|_0 \leq \lim_{k\to\infty} \|x^k\|_0$. Questa proprietà è fondamentale in ottimizzazione, come si può intuire dalla Figura 1.



(a) funzione semicontinua inferiormente (b) funzione non semicontinua inferiormente

Figura 1: Semicontinuità inferiore.

Infatti, una funzione non semicontinua inferiormente potrebbe "saltare" nel punto limite di una sequenza di punti prodotta da un ottimizzatore. Questo punto limite potrebbe quindi essere peggiore dei punti nel suo intorno, e quindi subottimo, o inammissibile, a seconda che la funzione sia l'obiettivo o un vincolo.

Nonostante questa fondamentale proprietà, la discontinuità e la nonconvessità di $\|\cdot\|_0$ rendono comunque molto complicato gestire termini di ℓ_0. In effetti, le tre formulazioni dei problemi di ottimizzazione sparsa nascondono una natura combinatoria che si rispecchia nel fatto che i problemi siano $\mathcal{NP}$-hard.

È per aggirare queste complessità che molti approcci esistenti per gestire i requisiti di sparsità nei problemi di ottimizzazione sono basati su approssimazioni della pseudo-norma ℓ_0. Inizieremo a studiare queste metodologie nella prossima sezione.

2 Formulazioni Approssimate

In questa sezione descriviamo alcuni approcci per problemi di ottimizzazione sparsa basati su approssimazioni della pseudo-norma ℓ_0.

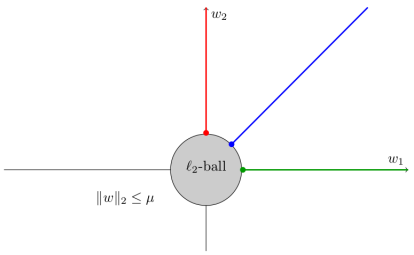
2.1 La Norma 1

La pseudo-norma ℓ_0, per definizione, modella in modo esatto la densità di un vettore e quindi consente di introdurre in un problema di ottimizzazione un elemento che induca esplicitamente la sparsità della soluzione. Tuttavia, abbiamo visto nella sezione precedente che ℓ_0 ha scarse proprietà di regolarità.

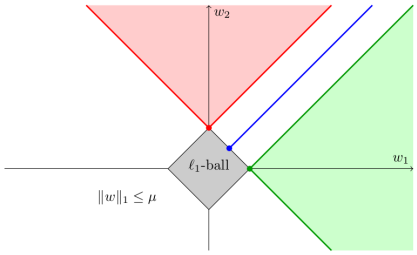
Abbiamo anche visto che le norme, di cui ℓ_0 è un caso limite, hanno buone proprietà di regolarità e sono quindi più facilmente trattabili. È quindi lecito chiedersi se altre norme abbiano effetti "sparsity-inducing" simili a quelli di ℓ_0 e possano quindi essere impiegate come surrogati della pseudo-norma zero nei problemi di ottimizzazione sparsa.

In effetti, alcune norme hanno questa proprietà. Non tutte però. L'intuizione grafica di questo fatto è riportata in Figura 2.

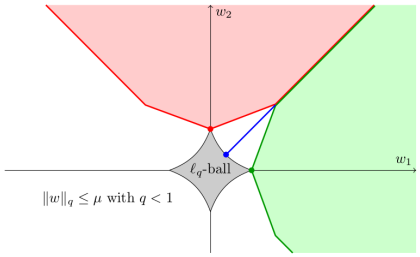
Possiamo infatti osservare che l'operatore di proiezione su una ℓ_p-sfera è in qualche modo sbilanciato a favore delle singolarità. Se si considera la ℓ_1-sfera unitaria, i punti sugli assi coordinati hanno un ampio bacino di attrazione. Questo effetto non è per esempio presente con la ℓ_2-sfera, che è *isotropica*. La capacità attrattiva delle singolarità sugli assi si accentua sempre di più al diminuire di p (ciò è in linea con il fatto che la ℓ_0 sia il caso limite). Viceversa, per valori superiori a 2 si ottiene l'effetto opposto, fino ad arrivare alla norma ℓ_∞ che penalizza solo la componente massima in valore assoluto e che quindi non ha alcun "vantaggio" ad azzerare componenti.



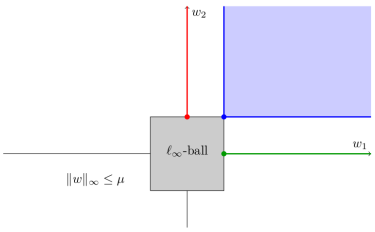
(a) La ℓ_2-sfera è isotropica.



(b) La ℓ_1-sfera favorisce le soluzioni singolari.



(c) La generica ℓ_q-sfera, con $q < 1$ ha un effetto sparsity inducing ancora più aggressivo.



(d) La ℓ_∞-sfera incoraggia soluzioni in cui $|w_1| = |w_2|$.

Figura 2: Bacini di attrazione delle operazioni di proiezione su ℓ_p-sfere.

Quale norma scegliere allora come surrogato della pseudo-norma zero? Le norme ℓ_p, con $p \in (0,1)$, sono state e sono ancora utilizzate, ma sono nonconvesse e nondifferenziabili e quindi non molto più trattabili della pseudo-norma zero stessa.

Per questo, l'approccio che storicamente è sempre parso più naturale, e che probabilmente è ancora oggi quello più ampiamente diffuso, consiste nell'utilizzare la norma ℓ_1 come proxy per indurre sparsità. Questa norma infatti è una funzione contiua, lineare a tratti e, soprattutto convessa.

In molti campi applicativi è quindi comune imbattersi in problemi di ottimizzazione del tipo:

Formulazione I Problema vincolato in norma 1:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &\text{s.t. } \|x\|_1 \leq s, \end{aligned} \tag{4}$$

dove s è un intero tale che $s \leq n$.

Formulazione II Problema con regolarizzazione ℓ_1:

$$\min_{x \in \mathbb{R}^n} f(x) + \lambda \|x\|_1 \tag{5}$$

dove $\lambda \in \mathbb{R}_+$.

Formulazione III Problema di minimizzazione della norma ℓ_1:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \|x\|_1 \\ &\text{s.t. } f(x) \leq \tau, \end{aligned}$$

dove $\tau \in \mathbb{R}_+$.

Per queste tre formulazioni valgono proprietà di equivalenza analoghe a quelle del caso con $\|\cdot\|_0$.

Pur presentando nondifferenziabilità, ognuno di questi problemi può essere risolto con metodologie abbastanza consolidate, che verranno brevemente descritte di seguito.

2.1.1 Metodi di tipo proximal gradient

Gli algoritmi di tipo *proximal gradient* rappresentano una generalizzazione del metodo del gradiente proiettato. Sono infatti basati su un opportuno operatore di prossimità che consente di gestire complessità specifiche di vari problemi, tra i quali la nondifferenziabilità dell'insieme ammissibile o di una parte dell'obiettivo.

In generale, il metodo di proximal gradient è utilizzato per risolvere problemi della forma

$$\min_x \ f(x) + \Omega(x),$$

dove Ω è una funzione nondifferenziabile.

Ad ogni iterazione la nuova iterata x^{k+1} è ottenuta risolvendo il seguente problema di ottimizzazione

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \|x - (x^k - \nabla f(x^k))\|_2^2 + \Omega(x).$$

È abbastanza immediato accorgersi che:

- Se Ω non è presente, si ricava il classico metodo del gradiente:

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

  in cui la lunghezza del passo può eventualmente essere aggiustata in modo opportuno con una ricerca di linea.

- Se Ω è la funzione indicatrice di un insieme convesso C

  $$\Omega(x) = \begin{cases} 0 & \text{se } x \in C, \\ +\infty & \text{altrimenti,} \end{cases}$$

  allora si recupera lo schema di aggiornamento del metodo del gradiente proiettato:

  $$x^{k+1} = \text{proj}_C(x^k - \nabla f(x^k)) = x^k + (\text{proj}_C(x^k - \nabla f(x^k)) - x^k),$$

  dove di nuovo il passo può essere aggiustato con una ricerca di linea.

In generale, il passo chiave dei metodi proximal gradient consiste nel calcolo dell'*operatore di prossimità*

$$\text{Prox}_\Omega(x) = \arg\min_y \|y - x\|_2^2 + \Omega(y),$$

che è univocamente definito essendo l'obiettivo dell'operatore di argmin fortemente convesso.

In conclusione, il metodo di proximal gradient effettua passi della forma

$$x^{k+1} = \text{Prox}_\Omega(x^k - \nabla f(x^k)).$$

Torniamo ora ai casi particolari dei problemi in norma 1. Nel caso del problema (4), abbiamo

$$\Omega(x) = \begin{cases} 0 & \text{se } \|x\|_1 \leq s, \\ +\infty & \text{altrimenti.} \end{cases}$$

L'operatore Prox_Ω consiste in questo caso nella proiezione sulla ℓ_1-sfera di raggio s, che può essere calcolata risolvendo un problema di programmazione quadratica continua attraverso uno degli efficientissimi solver specifici per questa classe di problemi.

Nel caso del problema (5), abbiamo $\Omega(x) = \lambda\|x\|_1$; possiamo osservare che

$$\begin{aligned}\text{Prox}_\Omega(x) &= \arg\min_y \|y - x\|_2^2 + \lambda\|y\|_1 \\ &= \arg\min_y \sum_{i=1}^{n} (y_i - x_i)^2 + \lambda|y_i|,\end{aligned}$$

il problema è quindi separabile:

$$y_i^\star = \arg\min_{y_i} (y_i - x_i)^2 + \lambda|y_i|.$$

Abbiamo inoltre che

$$\min_{y_i} (y_i - x_i)^2 + \lambda|y_i| = \min\left\{ \min_{y_i \geq 0} (y_i - x_i)^2 + \lambda y_i,\ \min_{y_i \leq 0} (y_i - x_i)^2 - \lambda y_i \right\}.$$

Consideriamo i due casi (Figura 3):

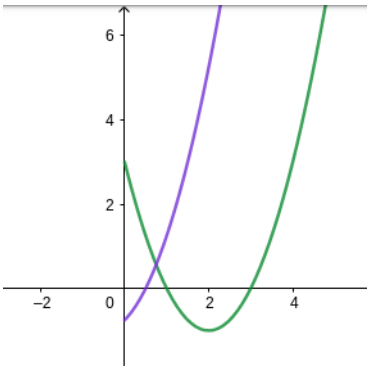
(i) l'obiettivo di
$$\min_{y_i \geq 0} (y_i - x_i)^2 + \lambda y_i$$
è una parabola con minimo in $x_i - \frac{\lambda}{2}$. Quindi la soluzione del sotto-problema è $y_i^\star = \max\{0, x_i - \frac{\lambda}{2}\}$.

(ii) l'obiettivo di
$$\min_{y_i \leq 0} (y_i - x_i)^2 - \lambda y_i$$
è una parabola con minimo in $x_i + \frac{\lambda}{2}$. Quindi la soluzione del sotto-problema è $y_i^\star = \min\{0, x_i + \frac{\lambda}{2}\}$.

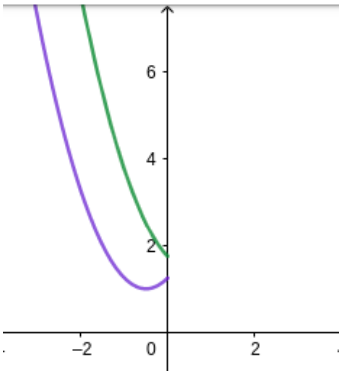
Pertanto, possiamo distinguere tre casi nel calcolo di $y_i^\star$:

$$y_i^\star = \begin{cases} x_i - \frac{\lambda}{2} & \text{se } x_i > \frac{\lambda}{2}, \\ 0 & \text{se } -\frac{\lambda}{2} \leq x_i \leq \frac{\lambda}{2}, \\ x_i + \frac{\lambda}{2} & \text{se } x_i < -\frac{\lambda}{2}, \end{cases}$$

ed ottenere il risultato dell'operatore di prossimità componente per componente in forma chiusa.



(a) Punti di minimo di parabole con vincolo $x \geq 0$



(b) Punti di minimo di parabole con vincolo $x \leq 0$

Figura 3:

2.1.2 Metodi derivative-free

Metodi senza derivate possono essere utilizzati per risolvere problemi che presentano non differenziabilità.

A titolo di esempio, il metodo di *coordinate search* può essere impiegato per la formulazione (5). In breve, il metodo prevede di effettuare un passo di prova lungo ogni coordinata e anti-coordinata, con un passo fissato; se almeno uno di questi passi ha prodotto un miglioramento dell'obiettivo, si sceglie il migliore tra questi. Dopodiché, la lunghezza del passo viene aggiornata, incrementandola se l'iterazione ha prodotto un sufficiente decremento o, al contrario, riducendola se non è stato trovato un punto migliore dell'iterata corrente.

In formule:

$$x^{k+1} = \arg\min_{z} F(z) \quad \text{s.t. } z \in \{x^k \pm \alpha^k e_i \mid i = 1, \ldots, n\} \cup \{x^k\}$$

$$\alpha^{k+1} \leftarrow \begin{cases} > \alpha^k & \text{se } F(x^{k+1}) < F(x^k) - \eta, \\ < \alpha^k & \text{se } F(x^{k+1}) = F(x^k), \\ = \alpha^k & \text{altrimenti.} \end{cases}$$

Metodi derivative-free come il coordinate search hanno garanzie di convergenza a punti che soddisfano opportune condizioni necessarie di ottimalità (Clarke-stazionarietà).

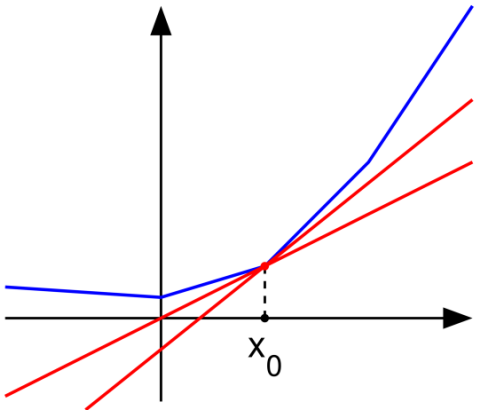


Figura 4: Subgradienti di una funzione nondifferenziabile in $\mathbb{R}$.

2.1.3 Metodi basati sul subgradiente

Il *subgradiente* è una generalizzazione del concetto di gradiente nel caso di funzioni e vincoli convessi non necessariamente continuamente differenziabili.

Più specificamente, un vettore $d \in \mathbb{R}^n$ è un subgradiente di una funzione convessa f in un punto $\bar{x}$ se

$$f(\bar{x}) + d^T(y - \bar{x}) \leq f(y) \text{ per ogni } y \in \mathbb{R}^n.$$

L'insieme dei subgradienti in un punto è detto *subdifferenziale* e si indica con $\partial f(\bar{x})$. Un punto $x^\star$ è di minimo non vincolato per una funzione convessa se e solo se

$$0 \in \partial f(x^\star).$$

Per problemi convessi non differenziabili, si possono usare i sottogradienti per definire metodi analoghi a quelli basati su gradiente nel caso differenziabile.

Per esempio, per il problema (5) si può usare un metodo di discesa identico al metodo del gradiente con l'unica differenza che al posto di $\nabla f(x^k)$ viene utilizzato, come direzione di discesa, un vettore $d \in \partial f(x^k)$ (tipicamente quello a norma minima). Questo metodo è dimostrato convergere ad un punto $x^\star$ tale che la condizione $0 \in \partial f(x^\star)$ è soddisfatta.

2.1.4 Riformulazione e Rilassamento

Spesso in ottimizzazione è possibile definire, attraverso opportune manipolazioni delle variabili, formulazioni equivalenti di problemi complessi in cui gli elementi più difficili da gestire in qualche modo scompaiono. Ad esempio, i problemi che coinvolgono la norma ℓ_1 possono essere riscritti, in

modo equivalente, con formulazioni completamente differenziabili, al costo di introdurre nuove variabili e vincoli.

Sia $x \in \mathbb{R}^n$. Siano poi $u, v \in \mathbb{R}^n$ tali che $u_i = \max\{0, x_i\}$ e $v_i = \max\{0, -x_i\}$ per ogni $i = 1, \ldots, n$. È facile accorgersi che $x = u - v$. Inoltre, $|x_i| = u_i + v_i$ e quindi $\|x\|_1 = 1^T(u + v)$.

Possiamo poi osservare che

$$\begin{cases} x = u - v \\ u_i = \max\{0, x_i\} \quad \forall\, i = 1 \ldots, n \\ v_i = \max\{0, -x_i\} \quad \forall\, i = 1 \ldots, n \end{cases} \iff \begin{cases} x = u - v \\ u_i v_i = 0 \quad \forall\, i = 1 \ldots, n \\ u_i, v_i \geq 0 \quad \forall\, i = 1 \ldots, n \end{cases}$$

Concentriamoci sul problema (5) con penalizzazione ℓ_1. Effettuando le opportune sostituzioni, otteniamo la seguente formulazione equivalente:

$$\begin{aligned} \min_{u,v \in \mathbb{R}^n} \; & f(u - v) + \lambda 1^T(u + v) \\ \text{s.t.} \; & u_i v_i = 0 \quad \forall\, i = 1 \ldots, n \\ & u, v \geq 0. \end{aligned} \tag{6}$$

Vediamo che la nondifferenziabilità intrinseca alla norma 1 è stata rimossa. Tuttavia, i vincoli di complementarietà del tipo $u_i v_i = 0$ sono ancora complessi da gestire.

Possiamo però considerare il seguente rilassamento della formulazione equivalente appena introdotta:

$$\begin{aligned} \min_{u,v \in \mathbb{R}^n} \; & f(u - v) + \lambda 1^T(u + v) \\ \text{s.t.} \; & u, v \geq 0, \end{aligned} \tag{7}$$

ottenuta rimuovendo i suddetti vincoli di complementarietà.

Questo rilassamento ha una interessantissima proprietà.

Proposizione 2.1. *Sia $(u^\star, v^\star)$ una soluzione ottimale del problema (7). Allora $u_i^\star v_i^\star = 0$ per ogni $i = 1, \ldots, n$, cioè $(u^\star, v^\star)$ è soluzione ottimale anche per il problema (6).*

Dimostrazione. Sia $(u^\star, v^\star)$ una soluzione ottimale del problema (7) e assumiamo per assurdo che la tesi non sia vera, cioè, esiste $h \in \{1, \ldots, n\}$ tale che $u_h^\star v_h^\star > 0$. Senza perdita di generalità, assumiamo che $u_h^\star \geq v_h^\star > 0$.

Definiamo $\hat{u}$ e $\hat{v}$ come segue:

$$\begin{aligned} \hat{u}_h &= u_h^\star - v_h^\star \geq 0, \quad \hat{u}_i = u_i^\star \; \forall\, i \neq h, \\ \hat{v}_h &= v_h^\star - v_h^\star = 0, \quad \hat{v}_i = v_i^\star \; \forall\, i \neq h. \end{aligned}$$

Osserviamo che $\hat{u} - \hat{v} = u^\star - v^\star$.

Valutando l'obiettivo del problema in $(\hat{u}, \hat{v})$ otteniamo

$$\begin{aligned} f(\hat{u}-\hat{v}) + \lambda 1^T(\hat{u}+\hat{v}) &= f(u^\star - v^\star) + \lambda 1^T(\hat{u}+\hat{v}) \\ &= f(u^\star - v^\star) + \lambda \left(\sum_{i \neq h} u_i^\star + \sum_{i \neq h} v_i^\star + (u_h^\star - v_h^\star) + (v_h^\star - v_h^\star) \right) \\ &= f(u^\star - v^\star) + \lambda \left(\sum_{i} u_i^\star + \sum_{i} v_i^\star - 2v_h^\star \right) \\ &< f(u^\star - v^\star) + \lambda \left(\sum_{i} u_i^\star + \sum_{i} v_i^\star \right) \\ &= f(u^\star - v^\star) + \lambda 1^T (u^\star + v^\star) \end{aligned}$$

che è assurdo, essendo $(u^\star, v^\star)$ ottimale per il problema rilassato. □

Si può quindi risolvere un problema differenziabile con soli vincoli di non-negatività, più semplice del problema originario, ma le cui soluzioni ottimali lo sono anche per quest'ultimo.

2.2 Approssimazione Concava

Un approccio di approssimazione della pseudo-norma zero, utilizzato spesso specialmente nel caso della Formulazione III quando $f(x)$ è una funzione affine, è quello basato su tecniche di programmazione concava.

Consideriamo la funzione $\|x\|_0$. Possiamo notare che questa può essere scritta come

$$\|x\|_0 = \sum_{i=1}^{n} s(|x_i|), \qquad s(t) = \begin{cases} 1 & \text{se } t > 0 \\ 0 & \text{altrimenti.} \end{cases}$$

Chiaramente la funzione s è discontinua, mentre i termini $|x_i|$ sono non differenziabili. Supponendo che $f(x) = Ax - b$, possiamo a questo punto scrivere equivalentemente la Formulazione 3 nel seguente modo:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \sum_{i=1}^{n} s(|x_i|) \\ \text{s.t. } & Ax - b \leq 0. \end{aligned}$$

Introducendo n nuove variabili e $2n$ vincoli lineari di disuguaglianza, possiamo eliminare la nondifferenziabilità in funzione obiettivo:

$$\begin{aligned} \min_{x,y \in \mathbb{R}^n} \ & \sum_{i=1}^{n} s(y_i) \\ \text{s.t. } & Ax - b \leq 0 \\ & -y_i \leq x_i \leq y_i. \end{aligned}$$

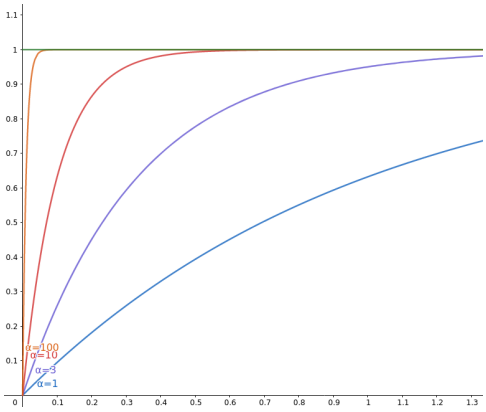


Figura 5: $1 - e^{-\alpha t}$ per diversi valori di α.

La formulazione è equivalente perché, dai vincoli, $y_i \geq |x_i|$ e poiché la funzione s è monotona crescente y_i^* sarà scelta, fissato x_i^*, più piccola possibile, andando quindi ad imporre che $y_i^* = |x_i^*|$.

Per rendere effettivamente gestibile il problema, è necessario ora eliminare la discontinuità in funzione obiettivo; è quindi opportuno sostituire la funzione $s(t)$ con una funzione monotona crescente che la approssimi. Ad esempio

$$\hat{s}(t) = 1 - e^{-\alpha t}$$

è monotona crescente, concava ed approssima la funzione scalino per $\alpha \to \infty$.

Il problema che può quindi essere risolto è il seguente:

$$\begin{aligned} \min_{x,y \in \mathbb{R}^n} \ & \sum_{i=1}^{n} 1 - e^{-\alpha y_i} \\ \text{s.t.} \ & Ax - b \leq 0 \\ & -y_i \leq x_i \leq y_i. \end{aligned} \tag{8}$$

Il problema (8) ha obiettivo concavo continuamente differenziabile e limitato inferiormente e vincoli lineari (cioè, l'insieme ammissibile è un poliedro). Valgono le seguenti proprietà:

- Se il poliedro ammissibile è non vuoto, allora il problema ammette minimo.
- Se il poliedro ammissibile ammette almeno un vertice, allora esiste un vertice che è soluzione ottima del problema.

Proposizione 2.2. *Assumiamo che $Ax - b \leq 0$ definisca un poliedro non vuoto che ammette un vertice. Allora esiste $\bar{\alpha}$ tale che per ogni $\alpha \geq \bar{\alpha}$ esiste un vertice soluzione ottima del problema* (8) *che è anche soluzione del problema* (3)*.*

Dimostrazione. Supponiamo per assurdo che esista una sequenza $\{\alpha^k\}$ tale che $\alpha_k \to \infty$ e il vertice ottimo $\{x^k, y^k\}$ per il problema (8) associato ad α^k non sia soluzione del problema (3) per ogni k.

Sia (x^*, y^*) un vertice soluzione ottima del problema (3). Per ipotesi, $\|y^*\|_0 < \|y^k\|_0$ per ogni k.

Il numero di vertici di un poliedro è finito, quindi esiste una sottosequenza $K \subseteq \{0, 1, \ldots\}$ tale che $(x^k, y^k) = (\bar{x}, \bar{y})$ per ogni $k \in K$.

Poiché $(\bar{x}, \bar{y})$ è soluzione del problema (8) per ogni $k \in K$, vale

$$\sum_{i=1}^{n} 1 - e^{-\alpha^k \bar{y}_i} \leq \sum_{i=1}^{n} 1 - e^{-\alpha^k y_i^*}.$$

Dunque, prendendo i limiti per $k \to \infty$, $k \in K$, abbiamo $\|\bar{y}\|_0 \leq \|y^*\|_0 < \|\bar{y}\|_0$, dove l'ultima disuguaglianza segue dall'ipotesi assurda. Abbiamo infine una contraddizione. □

Nella pratica, è quindi sufficiente risolvere il problema approssimato (8), identificando i vertici ottimi, e poi valutare quale tra questi abbia l'obiettivo migliore rispetto al problema originario, cioè, scegliere quello a norma zero minima.

3 Approcci diretti

Gli approcci che sfruttano approssimazioni come quelli visti in Sezione 2 consentono di ottenere spesso, in tempi computazionali ragionevoli, buone soluzioni per i problemi di ottimizzazione sparsa. Buone, ma pur sempre sub-ottimali per il problema originario.

Da qui in avanti considereremo i problemi di ottimizzazione sparsa nella loro formulazione "esatta", in cui è presente la pseudo-norma ℓ_0. Considereremo approcci che, al prezzo di un maggiore costo computazionale, vanno a trattare in modo esplicito la norma zero. Sebbene i problemi in cui compaiono termini di $\|\cdot\|_0$ siano $\mathcal{NP}$-hard, e sia quindi irragionevole pensare di ottenere una soluzione con certificato di ottimalità globale, le soluzioni ottenute considerando le formulazioni "esatte" sono spesso di qualità molto superiore a quelle ottenute con approssimazioni.

3.1 Riformulazione Mista-Intera con Variabili Indicatrici

La pseudo-norma zero può essere rimodellata, in modo del tutto equivalente, attraverso l'introduzione di variabili binarie indicatrici e passando da un problema di ottimizzazione continua ad un problema di programmazione matematica mista-intera.

Consideriamo infatti un vettore $x \in \mathbb{R}^n$ e il valore di $\|x\|_0$. Definiamo poi un vettore di variabili binarie, $\delta \in \{0, 1\}^n$, indicatrici dello stato di x,

cioè tale che

$$\delta_i = \begin{cases} 1 & \text{se } x_i \neq 0, \\ 0 & \text{altrimenti.} \end{cases}$$

Il valore di δ_i indica quindi se la i-esima componente di x è attiva ($\delta_i = 1$) oppure no ($\delta_i = 0$).

Possiamo notare che vale immediatamente l'equivalenza

$$\|x\|_0 = \sum_{i=1}^{n} \delta_i.$$

La relazione tra x e δ può essere definita formalmente con le seguenti implicazioni logiche:

$$\delta_i = 1 \implies x_i \neq 0, \qquad \delta_i = 0 \implies x_i = 0.$$

In realtà, se δ è utilizzata per modellare la pseudo-norma zero in un problema di minimizzazione con un termine ℓ_0 in obiettivo, o in un problema con un vincolo di upper bound su $\|x\|_0$, la prima implicazione è superflua. Infatti in questi casi, quando x_i è zero, un ottimizzatore sceglie $\delta_i = 0$ anche se questa è lasciata libera di assumere un qualsiasi valore in $\{0, 1\}$; scegliere $\delta_i = 1$ quando $x_i = 0$ incrementa la "norma" $\sum_i \delta_i$ senza alcun beneficio.

In questi casi, il *vincolo logico* può essere trasformato in equivalenti vincoli lineari, sfruttando una costante M che sia sufficientemente grande:

$$-M\delta_i \leq x_i \leq M\delta_i.$$

Vediamo infatti che se δ_i è 1, x_i può assumere un qualsiasi valore che abbia senso (deve appartenere all'intervallo $[-M, M]$). Se al contrario $\delta_i = 0$, la x_i corrispondente è forzata ad essere 0.

Possiamo quindi introdurre nei problemi (1), (2) e (3) queste variabili binarie, questi vincoli lineari e sostituire i termini di norma zero con $1^T\delta$ per ottenere i seguenti problemi di programmazione matematica mista intera equivalenti agli originali:

$$\begin{aligned} \min_{\substack{x\in\mathbb{R}^n,\\ \delta\in\{0,1\}^n}} \quad & f(x) \\ \text{s.t.} \quad & -M\delta_i \leq x_i \leq M\delta_i \quad \forall\, i = 1, \ldots, n \\ & \sum_{i=1}^{n} \delta_i \leq s \end{aligned}$$

$$\begin{aligned} \min_{\substack{x\in\mathbb{R}^n,\\ \delta\in\{0,1\}^n}} \quad & f(x) + \lambda \sum_{i=1}^{n} \delta_i \\ \text{s.t.} \quad & -M\delta_i \leq x_i \leq M\delta_i \quad \forall\, i = 1, \ldots, n \end{aligned}$$

$$\begin{aligned} \min_{\substack{x\in\mathbb{R}^n,\\ \delta\in\{0,1\}^n}} \quad & \sum_{i=1}^{n} \delta_i \\ \text{s.t.} \quad & f(x) \leq \tau \\ & -M\delta_i \leq x_i \leq M\delta_i \quad \forall\, i = 1, \ldots, n \end{aligned}$$

Questi tre problemi possono essere risolti con approcci di tipo *branch and bound*, tipici della ottimizzazione mista-intera.

In particolare, esistono software che implementano risolutori molto efficienti per problemi di programmazione mista-intera lineare e quadratica (Gurobi, CBC, CPLEX, GLPK). Questi possono essere impiegati in modo fruttuoso quando:

- l'obiettivo è lineare o quadratico convesso;
- i vincoli sono lineari;
- la dimensione del problema è ragionevole ($n \sim 10^3$ per problemi lineari, $n \sim 10^2$ per problemi quadratici).

In questi scenari, i solver sono in grado di produrre, in tempi computazionali ragionevoli, soluzioni con certificato di ottimalità globale per il problema. Nella pratica, dal punto di vista dell'efficienza dei solver, è importante scegliere bene il valore di M: troppo piccolo introduce dei box indesiderati sulle variabili, ma troppo grande peggiora sensibilmente l'efficienza del solver, che deve esplorare una porzione dello spazio ammissibile eccessivamente grande.

3.2 Problemi con vincolo di cardinalità

Se si ha a che fare con problemi di ottimizzazione sparsa sufficientemente semplici (in cui la principale complessità è data dai termini di norma zero), la strategia di riformulazione mista-intera è certamente la più indicata.

Quando però nei problemi ci sono elementi di forte nonlinearità e/o nonconvessità, oppure sono a più larga scala, non c'è alternativa a trattarli con tecniche di ottimizzazione continua.

Da qui in avanti, per semplicità, faremo riferimento al problema con vincoli di cardinalità (1). Questa formulazione è forse la più utilizzata nelle applicazioni e sicuramente la più studiata dal punto di vista teorico. Per le altre due formulazioni comunque si possono introdurre concetti e metodologie simili a quelle che vedremo nel seguito. Per semplicità inoltre, non considereremo i casi in cui siano presenti vincoli aggiuntivi oltre a quello di sparsità.

3.2.1 Condizioni di Ottimalità

Il problema (1) è un problema di ottimizzazione continua nonconvesso. È quindi ragionevole studiare condizioni necessarie di ottimalità globale e di ottimalità locale per esso.

Iniziamo introducendo un importante concetto relativo a soluzioni di problemi con vincoli di cardinalità: *il supporto*, cioè l'insieme delle variabili attive di una soluzione $\bar{x}$.

Definizione 3. Sia $x \in \mathbb{R}^n$ una soluzione ammissibile del problema (1); il *supporto* di x è l'insieme indicato con $I_1(x)$ e definito da

$$I_1(x) = \{i | x_i \neq 0\}.$$

Definiamo poi il complementare di $I_1(x)$, $I_0(x) = \{1, \ldots, n\} \setminus I_1(x)$, cioè l'insieme delle componenti nulle di x. Una soluzione x si dice *a supporto completo* se $\|x\|_0 = s$.

Per cominciare, notiamo una caratteristica del problema (1). Indichiamo con $\mathcal{X}$ l'insieme ammissibile del problema, cioè, $\mathcal{X} = \{x \in \mathbb{R}^n \mid \|x\|_0 \leq s\}$.

Essendo un problema di ottimizzazione continua, vale la classica definizione di ottimalità locale: $\bar{x} \in \mathcal{X}$ è minimo locale se esiste una sfera $B(\bar{x}, \epsilon)$ tale che $f(\bar{x}) \leq f(y)$ per ogni $y \in \mathcal{X} \cap B(\bar{x}, \epsilon)$.

Tuttavia, per questo particolare problema, ogni minimo locale di f rispetto ad un dato supporto di s variabili è minimo locale anche rispetto a tutte le variabili. Pertanto il numero di minimi locali di (1) cresce come $\binom{n}{s}$: il concetto di ottimalità locale è quindi di limitata utilità pratica.

Ha quindi senso concentrarsi su condizioni necessarie di ottimalità globale, tenendo presente che condizioni più stringenti consentono di restringere il numero di candidati ottimi globali e scartare punti in cui l'obiettivo è localmente buono ma globalmente pessimo.

Iniziamo quindi ad introdurre una prima condizione necessaria di ottimalità per il problema (1), che estende le condizioni di ottimalità basate su informazioni del primo ordine in problemi non vincolati differenziabili al caso in cui è presente il vincolo nondifferenziabile e nonconvesso dato dalla ℓ_0.

Definizione 4. Un punto $\bar{x} \in \mathcal{X}$ è una soluzione *basic feasible* (BF) se:

- $\|\bar{x}\|_0 = s$ e $\nabla_i f(\bar{x}) = 0$ per ogni $i \in I_1(\bar{x})$, oppure
- $\|\bar{x}\|_0 < s$ e $\nabla f(\bar{x}) = 0$.

Proposizione 3.1. *Sia $x^\star$ una soluzione ottimale di* (1)*. Allora $x^\star$ è basic feasible per* (1)*.*

La basic feasibility è una condizione di ottimalità basata su informazioni di tipo locale. In effetti, è anche una condizione necessaria di ottimalità locale, quindi è piuttosto debole. D'altra parte, il problema (1) ha una natura

combinatoria di cui la proprietà BF non tiene conto: non considera infatti potenziali cambi di supporto vantaggiosi a partire dal punto considerato.

In definitiva, si può vedere la proprietà di basic feasibility come una stazionarietà rispetto alle variabili nel supporto, fissate le altre, quando il supporto è completo, mentre quando $\|x\|_0 < s$ coincide con la stazionarietà del caso non vincolato.

Una condizione di ottimalità più forte è la seguente:

Definizione 5. Un punto $\bar{x} \in \mathcal{X}$ è detto Minimo Component-Wise (CW-minimum) per (1) se

- $\|\bar{x}\|_0 < s$ e per ogni $i \in \{1, \dots, n\}$ vale

$$f(\bar{x}) = \min_t f(\bar{x} + te_i),$$

  oppure

- $\|\bar{x}\|_0 = s$ e per ogni $i \in \{1, \dots, n\}$ e $j \in I_1(\bar{x})$ vale

$$f(\bar{x}) \leq \min_t f(\bar{x} + te_i - \bar{x}_j e_j).$$

Un punto è CW-ottimale se non c'è modo di migliorare l'obiettivo cambiando il valore di un'unica componente, oppure, quando il supporto è pieno, non è neppure possibile migliorare la soluzione effettuando uno scambio di variabili: una nel supporto viene azzerata e un'altra può essere mossa. È evidente che un ottimo globale per il problema di ottimizzazione con vincolo di cardinalità è un punto CW-ottimale.

Si capisce anche che, rispetto alla basic feasibility, il concetto di CW-ottimalità consente di tenere in considerazione eventuali cambi di supporto vantaggiosi, sebbene limitati ad uno scambio tra due variabili. È quindi una condizione di ottimalità più forte.

Non è quindi sorprendente la seguente proposizione.

Proposizione 3.2. *Sia $x^\star \in \mathcal{X}$; valgono le seguenti implicazioni:*

- *se $x^\star$ è una soluzione ottimale per il problema* (1)*, Allora $x^\star$ è CW-minimum.;*

- *se $x^\star$ è CW-ottimale per* (1)*, allora $x^\star$ è basic feasible.*

La ottimalità CW è una proprietà abbastanza valida, ma ha un paio di difetti: il primo, ovviamente, è che considera come unico potenziale cambio di supporto lo scambio tra due variabile; il secondo limite è che è basata su informazioni globali: verificare se un punto è CW-ottimale richiede di calcolare l'ottimo globale di funzioni scalari possibilmente nonconvesse, e non è quindi sempre possibile.

Vorremmo quindi una condizione basata esclusivamente su informazione locale ma che al tempo stesso consenta di tenere in considerazione di possibili cambi nel supporto.

Introduciamo una nuova riformulazione mista-intera del problema (1):

$$\begin{aligned} \min_{\substack{x\in\mathbb{R}^n,\\ y\in\{0,1\}^n}} \quad & f(x) \\ \text{s.t. } & e^T y \geq n - s \\ & x_i y_i = 0 \quad \forall\, i = 1, \ldots, n, \end{aligned} \tag{9}$$

basata su vincoli di complementarietà. Si vede che solo le variabili x_i corrispondenti a y_i nulli possono essere nulle; in qualche modo, il vettore di variabili binarie y definisce quali sono le variabili attive del vettore x che possono essere non nulle; il vincolo $e^T y \geq n - s$ consente di tagliare tutte le soluzioni in cui ci sono più di s elementi di y nulli e quindi, potenzialmente, un numero di componenti di x non nulle maggiore di s.

Possiamo formalizzare quanto appena detto come segue. Abbiamo che

$$\{i \mid y_i = 0\} = I_0(y) \supseteq I_1(x);$$

e $I_0(y) = I_1(x)$ se vale la complementarietà stretta (in questo caso particolare le variabili nulle di y identificano il supporto di x); in generale comunque vale quindi

$$\|x\|_0 = |I_1(x)| \leq |I_0(y)|,$$

e pertanto

$$e^T y = |I_1(y)| = n - |I_0 y()| \leq n - \|x\|_0.$$

Allora osserviamo che imporre $e^T \geq n - s$ implica che $n - s \leq n - \|x\|_0$ e quindi $\|x\|_0 \leq s$.

A partire da questa riformulazione, è possibile definire opportune condizioni di ottimalità per il problema di partenza che tengano conto della natura combinatoria del problema. Queste condizioni sono basate sul concetto di *intorno discreto*.

Definizione 6. Dato un punto (x, y) ammissibile per il problema (9), un *intorno discreto* $\mathcal{N}(x, y)$ è un insieme finito di punti ammissibili, vicini ad (x, y) secondo un qualche criterio e contenente (x, y) stesso.

Un esempio di intorno, utile per studiare il problema (1), è l'intorno $\mathcal{N}_\rho(x, y)$ dato da tutti quei punti $(\hat{x}, \hat{y})$ tali che $d_H(y, \hat{y}) \leq \rho$, dove d_H è la distanza di Hamming (cioè il numero di componenti diverse tra due vettori binari), e

$$\hat{x}_i = \begin{cases} x_i & \text{se } y_i = \hat{y}_i \\ 0 & \text{altrimenti} \end{cases}$$

Esempio 1. Consideriamo il problema (9) con $n = 3$ e $s = 2$ e sia $\rho = 2$. Sia poi (x, y) un punto ammissibile definito come segue

$$(x, y) = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

L'intorno $\mathcal{N}_\rho(x, y)$ è dato da

$$\mathcal{N}_2(x,y) = \left\{ \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$$

Fissato un particolare intorno, come appunto $\mathcal{N}_\rho$, possiamo definire un opportuno concetto di stazionarietà.

Definizione 7. Un punto $x^* \in \mathcal{X}$ è detto *$\mathcal{N}$-stazionario* se esiste un $y^* \in \{0, 1\}^n$ tale che

(i) (x^*, y^*) è ammissibile per (9);

(ii) x^* è stazionario per il problema continuo

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x_i = 0 \; \forall \, i : y^* = 1; \end{aligned}$$

(iii) ogni $(\hat{x}, \hat{y}) \in \mathcal{N}_\rho(x^*, y^*)$ soddisfa $f(\hat{x}) \geq f(x^*)$ e se $f(\hat{x}) = f(x^*)$, il punto $\hat{x}$ è stazionario per il problema continuo

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x_i = 0 \; \forall \, i : \hat{y}_i = 1. \end{aligned}$$

È facile rendersi conto che vale il seguente risultato

Proposizione 3.3. *Sia x^* un punto di minimo del problema* (1)*. Allora x^* è $\mathcal{N}$-stazionario.*

Questo concetto di stazionarietà, se utilizzato con un opportuno intorno discreto, consente di tenere conto di potenziali cambi al supporto, implicitamente considerati tramite le variazioni della variabile y. Per esempio, $\mathcal{N}_\rho$ con $\rho = 2$ considera soluzioni ottenute facendo uno scambio di variabili nel supporto, o l'aggiunta di due variabili contemporaneamente, mentre $\rho = 4$ addirittura tiene conto di scambi doppi e in generale di variazioni del supporto che coinvolgono fino a 4 variabili. Chiaramente, maggiore è il valore di ρ e maggiore è il numero di punti nell'intorno, e quindi la forza della $\mathcal{N}$-stazionarietà come condizione di ottimalità, ma aumenta proporzionalmente anche il costo computazionale per verificarla.

È importante notare che la $\mathcal{N}_\rho$-stazionarietà implica la basic feasibility.

Proposizione 3.4. *Sia $x^\star$ $\mathcal{N}_\rho$-stazionario per il problema* (1)*. Allora $x^\star$ è un punto BF.*

Nelle prossime sezioni vedremo alcuni algoritmi di ottimizzazione progettati specificamente per produrre soluzioni che soddisfano alcune delle condizioni di ottimalità viste fin qui per il problema (1).

3.3 Iterative Hard Thresholding

Consideriamo la funzione indicatrice dell'insieme ammissibile $\mathcal{X}$:

$$\Omega_{\mathcal{X}}(x) = \begin{cases} 0 & \text{se } x \in \mathcal{X}, \\ +\infty & \text{altrimenti.} \end{cases}$$

Abbiamo visto in precedenza che, quando Ω è la funzione indicatrice di un insieme convesso, l'operatore di prossimità associato al problema

$$\min_x \ f(x) + \Omega(x)$$

coincide con la proiezione sull'insieme stesso e l'algoritmo di tipo proximal gradient corrispondente non è altro che il metodo del gradiente proiettato.

Una situazione analoga si ritrova nel caso di $\Omega_{\mathcal{X}}$, anche se $\mathcal{X}$ è nonconvesso. Abbiamo infatti che

$$\mathrm{Prox}_{\Omega_{\mathcal{X}}}(x) = \mathrm{proj}_{\mathcal{X}}(x) \in \arg\min_{y \in \mathcal{X}} \|y - x\|_2^2.$$

La proiezione sull'insieme nonconvesso $\mathcal{X}$ non ha soluzione unica; tuttavia, una soluzione può essere calcolata in forma chiusa in questo caso. Infatti

$$\min_{\|y\|_0 \le s} \|y - x\|_2^2 = \min_{\|y\|_0 \le s} \sum_{i=1}^{n} (y_i - x_i)^2.$$

Scegliendo opportunamente i valori di y rispettando il vincolo di cardinalità, possiamo azzerare fino ad s elementi della sommatoria, pagando un costo x_i^2 per ognuno degli altri $n - s$. È immediato quindi rendersi conto che la soluzione $y^\star$ tale che

$$y_i^\star = \begin{cases} x_i & \text{per le } s \text{ componenti con } |x_i| \text{ più grande,} \\ 0 & \text{altrimenti,} \end{cases}$$

sia ottimale per il problema.

È quindi possibile definire uno schema di tipo proximal gradient per il problema (1):

$$x^{k+1} = \mathrm{proj}_{\mathcal{X}} \left(x^k - \frac{1}{L} \nabla f(x^k) \right).$$

L'algoritmo è detto *Iterative Hard Thresholding*; il nome deriva dal fatto che, dopo un passo di discesa lungo l'antigradiente, viene effettuata una operazione di sogliatura che azzera tutte le $n - s$ variabili più piccole in valore assoluto. Nonostante l'operazione di sogliatura sia abbastanza drastica e quindi il supporto finale della soluzione trovata sia molto influenzato dal punto di partenza e dai primi passi, l'algoritmo ha buone proprietà di convergenza, che enunciamo di seguito:

Proposizione 3.5. *Supponiamo che ∇f sia una funzione Lipshcitz-continua di constante $L(f)$. Sia $\{x^k\}$ la sequenza prodotta dall'algoritmo IHT con $L > L(f)$. Allora*

(i) $\{f(x^k)\}$ *è monotona decrescente e convergente;*

(ii) $f(x^{k+1}) < f(x^k)$ *se* $x^{k+1} \neq x^k$*;*

(iii) $\|x^{k+1} - x^k\| \to 0$*;*

(iv) Ogni punto limite $\bar{x}$ *di* $\{x^k\}$ *è un punto basic feasible.*

3.4 Greedy Sparse-Simplex

L'algoritmo *Greedy Sparse-Simplex* è specificamente progettato per ottenere punti che soddisfano la condizione di CW-ottimalità (e quindi con garanzie di ottimalità superiori a quelle date dai punti ottenuti con IHT).

Analogamente alla condizione di CW-ottimalità, le iterazioni dell'algoritmo dipendono dalla cardinalità della soluzione corrente:

- se $\|x^k\|_0 < s$:
  - per ogni $i = 1, \ldots, n$, calcola $x^i = x^k + t_i e_i$ dove
    $$t_i \in \arg\min_t f(x^k + te_i)$$
  - $x^{k+1} \in \arg\min_{x^i,\, i=1,\ldots,n} f(x)$;
- se $\|x^k\|_0 = s$:
  - per ogni $i = 1, \ldots, n$, e $j \in I_1(x^k)$, calcola $x^{i,j} = x^k + t_i e_i - x^k_j e_j$ dove
    $$t_i \in \arg\min_t f(x^k + te_i - x^k_j e_j)$$
  - $x^{k+1} \in \arg\min_{x^{i,j}} f(x)$;

Quando il supporto non è pieno, c'è ancora spazio per aggiungere variabili: l'algoritmo effettua lo spostamento migliore in assoluto tra tutti

quelli che prevedono il movimento di una singola variabile. Quando invece il supporto è completo, le mosse contemplate sono quelle che prevedono l'azzeramento di una variabile non nulla e lo spostamento ottimale di una delle altre; in questo caso quindi vengono tenuti in considerazione possibili cambi di supporto ottenuto effettuando degli "swap".

Si vede bene come l'algoritmo ricalchi la definizione di ottimalità CW; dovrebbero pertanto apparire ragionevoli le proprietà di convergenza che possono essere dimostrate per l'algoritmo.

Proposizione 3.6. *Sia $\{x^k\}$ la sequenza prodotta dall'algoritmo GSS. Allora*

(i) $f(x^{k+1}) \leq f(x^k)$*;*

(ii) $f(x^{k+1}) = f(x^k)$ *solo se* $x^{k+1} = x^k$*; in tal caso* x^k *è CW-ottimale;*

(iii) Se $\{x^k\}$ è una sequenza infinita, ogni punto di accumulazione $\bar{x}$ è una soluzione CW-ottimale.

La condizione di ottimalità CW soddisfatta dai punti limite dell'algoritmo GSS è certamente buona. Tuttavia, questo ha comunque dei limiti di applicabilità:

- le iterazioni di GSS sono costose; in particolare, quando $\|x^k\|_0 = s$, devono essere risolti ns problemi scalari;
- la convergenza è lenta, a causa del fatto che ad ogni iterazione viene spostata una sola variabile; anche quando il supporto "ottimale" è stato identificato, servono molte iterazioni per arrivare ad una condizione di stazionarietà perché non viene mai effettuata un'ottimizzazione congiunta rispetto a tutte le variabili libere;
- richiede di identificare minimi globali di problemi in una variabile e non è quindi utilizzabile senza ipotesi di convessità component-wise.

3.5 Penalty Decomposition

Consideriamo un generico problema di ottimizzazione con un vincolo di uguaglianza:

$$\begin{aligned} \min_x \ & f(x) \\ \text{s.t. } & x \in X, \\ & h(x) = 0, \end{aligned}$$

dove X è un insieme convesso.

Possiamo associare a questo problema una *penalty function*:

$$q_\tau(x) = f(x) + \tau \|h(x)\|_2^2$$

dove $\tau > 0$ è un parametro detto *parametro di penalità*. Notiamo che q_τ consente di qualificare un punto $x \in X$, bilanciando il valore della funzione obiettivo in quel punto con la norma della violazione del vincolo di uguaglianza. Per $\tau \to \infty$, q_τ vale $+\infty$ per ogni punto che non soddisfa il vincolo $h(x) = 0$, mentre vale $f(x)$ per ogni punto ammissibile; pertanto un ottimo per q_τ su X è ottimo vincolato su $X \cap \{x \mid h(x) = 0\}$ rispetto ad f.

Questa considerazione è alla base del *Quadratic Penalty Method*; questo algoritmo di ottimizzazione vincolata ripete ciclicamente due passi:

- Calcola
$$x^{k+1} \in \arg\min_{x \in X} q_{\tau_k}(x);$$
- Aggiorna $\tau_{k+1} = \alpha\tau_k$, dove $\alpha > 1$.

L'algoritmo risolve una sequenza di sottoproblemi vincolati in X per valori crescenti di τ. Al limite, la soluzione ottenuta è ammissibile e ottimale per il problema di partenza. Nella pratica è utile costruire una sequenza di sottoproblemi, piuttosto che risolvere direttamente un problema con τ molto grande, per questioni di stabilità numerica e perché la soluzione del problema associato a τ_k è un buon punto di partenza per un algoritmo iterativo applicato al problema associato a τ_{k+1}.

Torniamo ora al problema (1). Possiamo sfruttare la cosiddetta strategia del *variable splitting*. Questa strategia consiste nel duplicare le variabili di un problema per disaccoppiare obiettivo e vincoli, per poi imporre l'uguaglianza tra i due blocchi di variabili tramite un vincolo di uguaglianza. Il seguente problema è equivalente al problema (1):

$$\begin{aligned} &\min_{x,y} f(x) \\ &\text{s.t. } \|y\|_0 \leq s \\ &\qquad x - y = 0. \end{aligned} \tag{10}$$

Anche se $\mathcal{X}$ non è convesso, si può pensare di applicare il quadratic penalty method al problema (**??**), a cui è associata una penalty function (coerciva)

$$q_\tau(x, y) = f(x) + \tau\|x - y\|_2^2.$$

Tuttavia, minimizzare $q_\tau(x, y)$ su $x \in \mathbb{R}^n$ e $y \in \mathcal{X}$ è difficile tanto quanto risolvere il problema originario, in quanto si vuole minimizzare una funzione in x e y con vincoli di cardinalità; di fatto, il quadratic penalty method richiederebbe di risolvere una sequenza di sottoproblemi complessi quanto lo stesso problema originario.

Possiamo però notare che, fissando y, quello che rimane è un problema di ottimizzazione continua nonvincolata con obiettivo coercivo. Inoltre, fissando $x = \bar{x}$, il minimizzare $q_\tau(\bar{x}, y)$ significa risolvere

$$\min_{y \in \mathcal{X}} \tau\|\bar{x} - y\|_2^2,$$

che non è altro che il problema della proiezione di $\bar{x}$ su $\mathcal{X}$.

L'approccio di *Penalty Decomposition* sfrutta queste considerazioni per modificare lo schema base del quadratic penalty method e renderlo efficiente nel caso del problema (**??**).

Lo schema di Penalty Decomposition per l'iterazione k-esima è il seguente:

- Inizializza $\hat{x}, \hat{y} = x^k, y^k$;
- **While** ($\|\nabla_x q_{\tau_k}(\hat{x}, \hat{y})\| \geq \varepsilon_k$):
  - $\hat{x} \in \arg\min_x \ f(x) + \tau_k \|x - \hat{y}\|^2$
  - $\hat{y} = \text{proj}_{\mathcal{X}}(\hat{x})$
- $x^{k+1}, y^{k+1} = \hat{x}, \hat{y}$
- $\tau_{k+1} = \alpha \tau_k$

Il passo di ottimizzazione globale e congiunta rispetto ad x e y è sostituito da un ciclo di tipo Gauss-Seidel sui due blocchi di variabili. Questo ciclo si arresta quando la coppia di variabili "temporanee" $(\hat{x}, \hat{y})$ è un punto ε_k-stazionario per q_{τ_k} rispetto ad x, cioè quando $\|\nabla_x q_{\tau_k}(\hat{x}, \hat{y})\| < \varepsilon_k$.

È ragionevole chiedere che uno schema di tipo Gauss-Seidel termini quando viene raggiunta una stazionarietà approssimata; non si richiede una stazionarietà esatta per poter garantire la terminazione finita del ciclo interno; un algoritmo iterativo ha infatti proprietà di convergenza asintotica: potrebbero essere richieste infinite iterazioni per produrre un punto strettamente stazionario e in quel caso l'intero schema di Penalty Decomposition non sarebbe formalmente ben definito. La sequenza ε_k è scelta a priori ed è tale che $\varepsilon_k \to 0$. In questo modo la precisione sul criterio di arresto del ciclo interno cresce progressivamente. Notiamo che si richiede una stazionarietà rispetto alle sole variabili x perché, al termine di ogni iterazione del ciclo interno, $\hat{y}$ è per definizione ottimo globale del problema rispetto a y.

Le proprietà di convergenza dell'algoritmo sono riportate nella seguente proposizione.

Proposizione 3.7. *Sia $\{x^k, y^k\}$ la sequenza generata dall'algoritmo di Penalty Decomposition applicato al problema (**??**), con $\{\varepsilon_k\}$ tale che $\varepsilon_k > 0$ per ogni k e $\varepsilon_k \to 0$. Allora*

(i) Ad ogni iterazione, il punto (x^{k+1}, y^{k+1}) viene prodotto in un tempo finito (terminazione finita del ciclo interno);

*(ii) Ogni punto di accumulazione $(\bar{x}, \bar{y})$ della sequenza è ammissibile per (**??**), cioè $\bar{x} = \bar{y}$ e $\|\bar{x}\|_0 = \|\bar{y}\|_0 \leq s$, e quindi $\bar{x}$ è ammissibile per il problema (1);*

(iii) Ogni punto di accumulazione $\bar{x}$ tale che $x^k \to \bar{x}$ per $k \to \infty$, $k \in K \subseteq \{0, 1, \ldots\}$ è basic feasible per il problema (1) se vale una delle seguenti condizioni:

- $\|\bar{x}\|_0 = s$;
- *per ogni $k \in K$ sufficientemente grande vale* $\|x^k\|_0 = \|\bar{x}\|_0$;
- *per ogni $k \in K$ sufficientemente grande vale* $\|x^k\|_0 < s$.

Notiamo che la proprietà di basic feasibility dei punti limite è garantita *a posteriori*; è possibile assicurarsi che la proprietà BF valga, senza bisogno di guardare il gradiente, semplicemente controllando la struttura del punto limite o la sequenza di punti che lo ha generato. Tuttavia, quando l'algoritmo viene lanciato, non si può avere la certezza che la soluzione ottenuta soddisferà questa proprietà (e questa certezza a priori è quella che si vorrebbe per gli algoritmi di ottimizzazione). È comunque vero che il caso di gran lunga più probabile, in pratica, è quello in cui $\|\bar{x}\|_0 = s$ e quindi la soluzione è BF.

L'algoritmo ha proprietà di convergenza più deboli di IHT e GSS. Inoltre condivide con GSS il limite di richiedere un'operazione di argmin su una funzione potenzialmente nonconvessa; infine, dal punto di vista computazionale mostra una scarsa velocità di convergenza se le sequenze τ_k ed ε_k non sono calibrate in modo opportuno.

D'altra parte il metodo di Penalty Decomposition presenta anche dei vantaggi rispetto ai succitati algoritmi:

- A differenza di IHT, che richiede una proiezione sull'insieme ammissibile, e di GSS, che richiede che le variabili siano separabili rispetto ai vincoli, il metodo di Penalty Decomposition è facilmente estendibile al caso in cui siano presenti vincoli aggiuntivi del tipo $h(x) = 0$ e $g(x) \leq 0$. È sufficiente infatti modificare la funzione di penalità come segue

  $$q_\tau(x, y) = f(x) + \tau \left(\|x - y\|^2 + \|h(x)\|^2 + \| \max\{0, g(x)\} \|^2 \right)$$

  per ottenere le stesse proprietà di convergenza del caso senza ulteriori vincoli.

- L'operazione di argmin su x può essere sostituita da un passo di discesa lungo l'antigradiente, con lunghezza del passo calcolata con il metodo di Armijo. Questa modifica non inficia le proprietà di convergenza dell'algoritmo, che può quindi essere usato con funzioni obiettivo nonconvesse.

- Se le sequenze $\{\tau_k\}$ ed $\{\varepsilon_k\}$ sono calibrate in modo opportuno, l'algoritmo risulta essere particolarmente efficiente almeno nel trovare soluzioni se non sono richiesti livelli di precisione particolarmente elevati.

3.6 Sparse Neighborhood Search*

In questa sezione descriviamo brevemente l'algoritmo di Sparse Neighborhood Search; il metodo in questione sfrutta il concetto di intorno discreto associato alla riformulazione (9) per ottenere punti $\mathcal{N}$-stazionari. Ovviamente, le performance dell'algoritmo variano in funzione della scelta dell'intorno discreto $\mathcal{N}$, sia in termini di efficienza che di efficacia. In particolare, più ampio è l'intorno, più forti saranno le proprietà di convergenza dell'algoritmo ma al contempo saranno pure maggiori i costi computazionali. Le operazioni effettuate dall'algoritmo possono essere riassunte come segue:

- A partire dal punto (x^k, y^k), calcola $\mathcal{N}(x^k, y^k)$
- Per ogni punto $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$:
  - effettua una ricerca ricerca locale rispetto a x, partendo da $\hat{x}^k$, tenendo fissato $\hat{y}^k$ (possono quindi essere spostate solo le variabili x_i corrispondenti a y_i nulli): ottieni $\hat{x}^{k+1}$
  - se $f(\hat{x}^{k+1}) \leq f(x^k) - \nu_k$ allora poni

    $$x^{k+1} = \hat{x}^{k+1}$$

    ed esci dal ciclo;
  - se la lista dei punti nell'intorno viene scorsa senza ottenere un sufficiente decremento, imposta $\nu_{k+1} < \nu_k$ e passa all'iterazione successiva.

L'algoritmo SNS presenta le seguenti proprietà di convergenza.

Proposizione 3.8. *Sia $\{x^k, y^k\}$ la sequenza prodotta dall'algoritmo di Sparse Neighborhood Search applicato al problema* (9)*. Allora la sequenza ammette punti di accumulazione $(\bar{x}, \bar{y})$ ammissibili per il problema* (9) *e tali che $\bar{x}$ è $\mathcal{N}$-stazionario per il problema* (1)*.*