Lezione4

IPM per l'Ottimizzazione Conica: SOCP e SDP

Andrea Cassioli

Dipartimento di Sistemi e Informatica

Universitá di Firenze

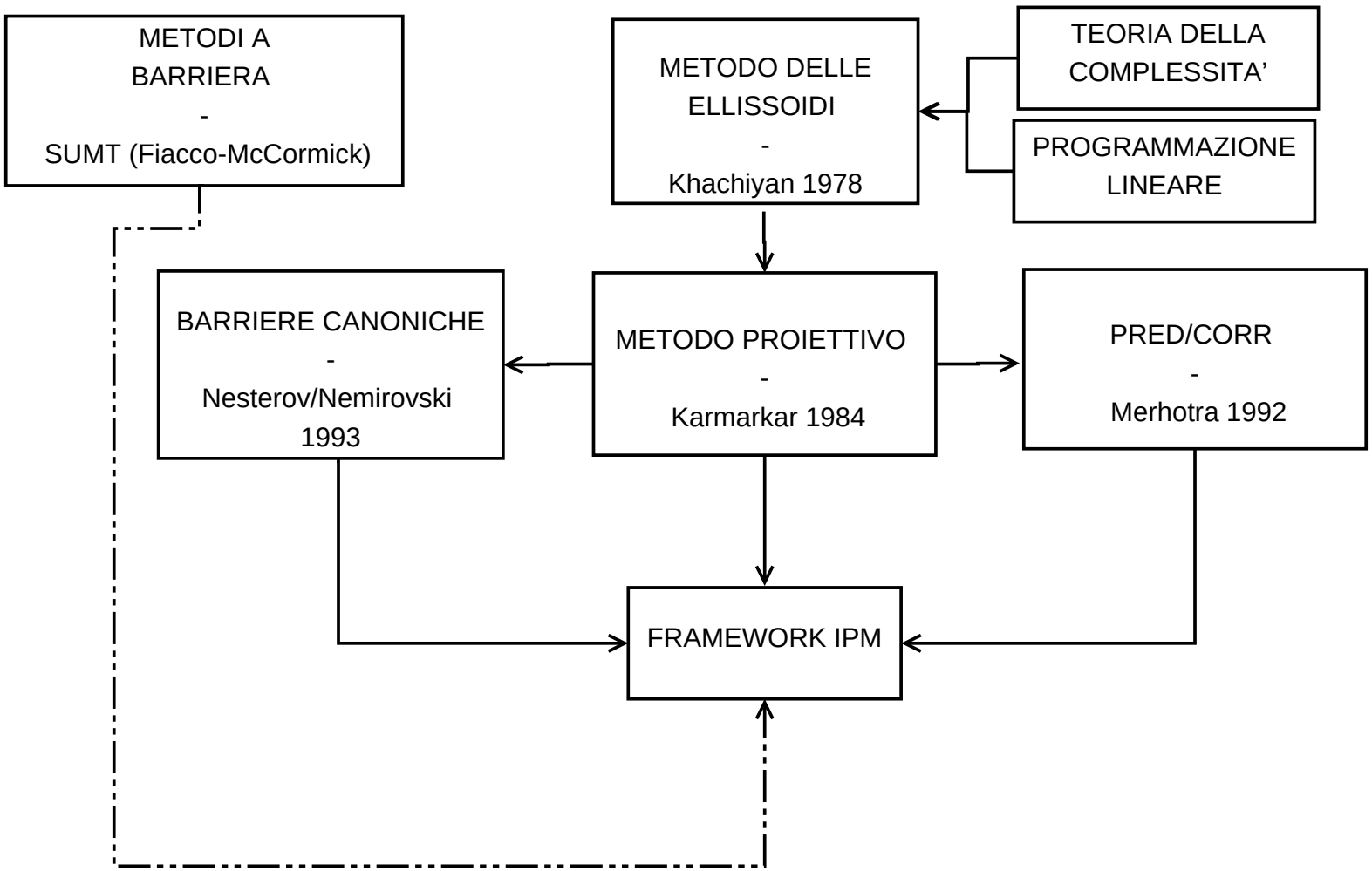
http://gol.dsi.unifi.it/users/cassioli

cassioli@dsi.unifi.it

28/05/2008

Evoluzione IPM

Cronogramma



Convessitá

Convergenza metodi a Barriera

Esistono diversi risultati sulla convergenza :

- se il problema e' convesso e sappiamo risolverlo allora si converge all'ottimo globale;
- sotto minori assunzioni, se si converge, allora troviamo un punto ammissibile e con le LICQ anche ottimo;

Tuttavia i metodi a barriera non sono stati concepiti in modo specifico per il caso convesso.

Convessitá

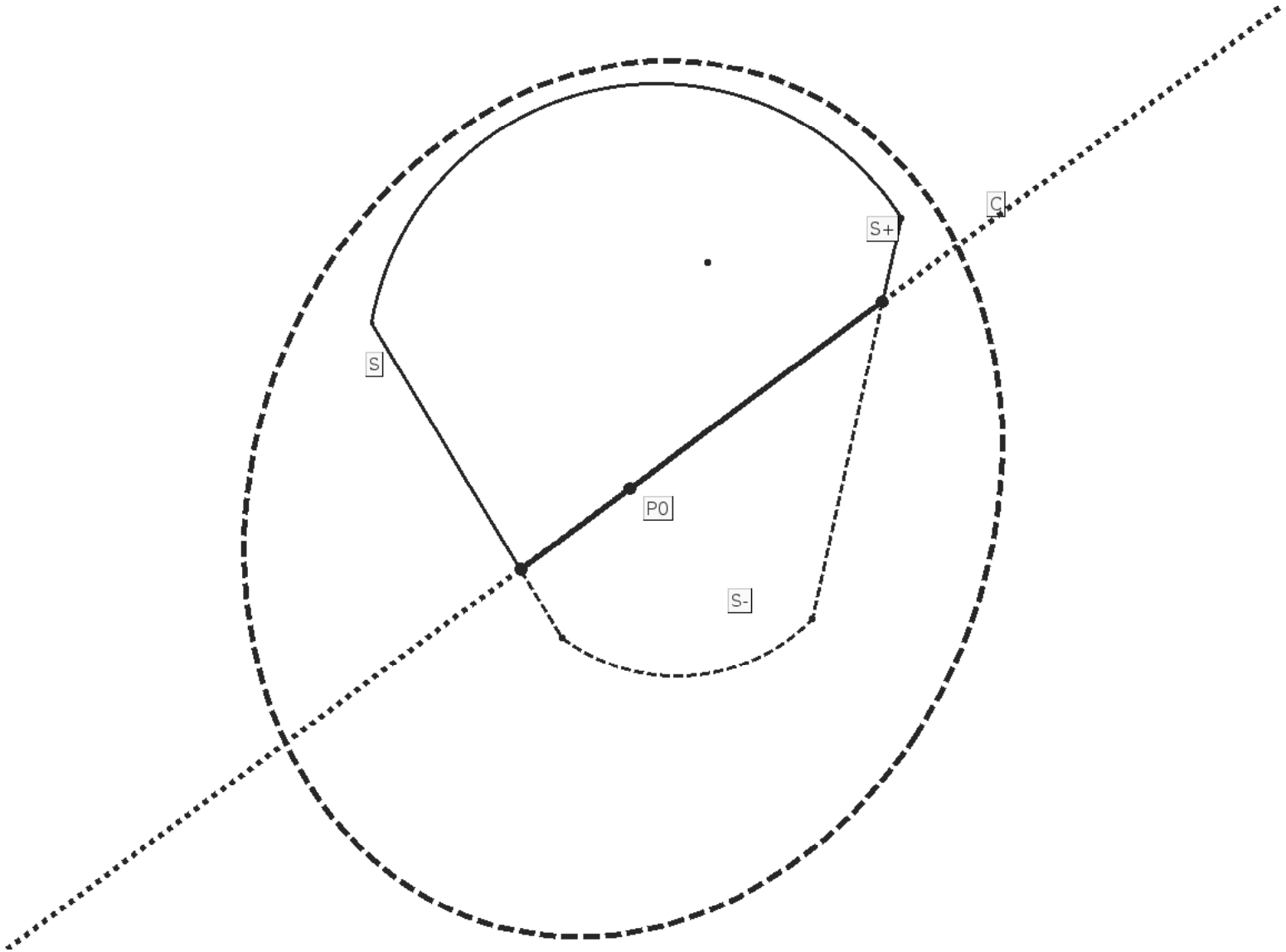
Estensione Algoritmo Ellissoidi

L'algoritmo delle ellissoidi deriva da un metodo (Yudin 1976) per generici problemi convessi :

- l'ellissoide puó avvolgere un qualunque corpo convesso;
- il taglio viene effettuato con il subgradiente;
- si ipotizza di disporre di un oracolo che separi la parte di ellissoide che contiene la regione ammissibile da l'altra;

Convessitá

Estensione Algoritmo Ellissoidi - 2



Convessitá

Estensione LCP Monotono e CQP

I casi di LCP Monotono e CQP si estendono direttamente col framework per la PL:

- sono entrambi problemi convessi;
- le KKT sono analoghe al caso PL;
- contengono la PL come caso speciale;

Convessitá

Proprietá CP

Se la barriera é scelta ragionevolmente, nel caso convesso il CP risulta una curva:

- analitica;
- parametrica rispetto al peso della barriera;
- terminante nell'insieme ottimo;

Convessitá

Proprietá CP

Se la barriera é scelta ragionevolmente, nel caso convesso il CP risulta una curva:

- analitica;
- parametrica rispetto al peso della barriera;
- terminante nell'insieme ottimo;

questo permette:

- introdurre il concetto di vicinanza al CP;
- definire lo schema Path-Following
- legare l'avanzamento dell'algoritmo al CP per conoscere la complessitá

Commenti

Dalle osservazioni precedenti ci potremmo porre il seguente

Quesito

Il buon comportamento (polinomialitá) degli IPM per problemi di PL é dovuto alla convessitá?

Commenti

Dalle osservazioni precedenti ci potremmo porre il seguente

Quesito

Il buon comportamento (polinomialitá) degli IPM per problemi di PL é dovuto alla convessitá?

La risposta data da Nesterov e Nemirovski nel 1994 é parzialmente negativa: la convessitá non é sufficiente!

IPM e Polinomialitá

Barriere Canoniche

La forma del CP viene determinata anche dallas barriera usata: risulta fondamentale l'uso di una funzione barriera che goda della proprietá di *self-concordance*.

La forma del CP viene determinata anche dallas barriera usata: risulta fondamentale l'uso di una funzione barriera che goda della proprietá di *self-concordance*.

Funzioni Self-Concordant (Nesterov e Nemirovski 1994)

una funzione convessa $\varphi : F_0 \to \mathbb{R}$, con $F_0 \subseteq \mathbb{R}^n$ convessa, é θ-self-concordant in F_0 se:

- φ é $\mathbb{C}^3$ in F_0
- $\forall y \in F_0, \forall h \in \mathbb{R}^n \quad |\nabla^3 \varphi(y)[h,h,h]| \leq 2\theta(h^T \nabla^2 \varphi(y) h)^{3/2}$

Questa classe di funzioni barriera prende il nome di *barriere canoniche*

Riassumendo

- La convessitá del problema garantisce buone proprietá di convergenza e di struttura del CP.
- La convessitá non basta per garantire la polinomialitá degli IPM ma occorre garantire una certa forma del CP tramite opportune barriere.
- La polinomialitá viene ottenuta usando funzioni barriera canoniche

Per la PL, tramite la barriera logaritmica (che é self-concordant), abbiamo giá i risultati attesi: é naturale provare ad estendere la PL per cercare risultati analoghi!

Partiamo dalla PL

Osserviamo l'insieme ammissibile della PL in modo diverso:

$$Ax - b \geq 0 \quad \Leftrightarrow \quad Ax - b = 0, x \geq 0$$

In entrambi i casi imponiamo disuguaglianze vettoriali, ovvero definamo un ordinamento parziale su $\mathbb{R}^n$ grazie alla proprietá di:

- riflessivitá - $a \geq a$
- antisimmetria - se $a \geq b$ e $b \geq a$ allora $a = b$
- transitivitá - se $a \geq b$ e $b \geq c$ allora $a \geq c$
- omogenitá - se $a \geq b$ e $\lambda \geq 0$ allora $a = \lambda b$
- additivitá - se $a \geq b$ e $c \geq d$ allora $a + b \geq d + c$

Programmazione Conica

La strutturazione forte della PL si basa sull'ordinamento parziale indotto da $\mathbb{R}^n_+$, ma altri ordinamenti possono essere usati su altri insiemi.

In particolare queste proprietá caratterizzano gli insiemi conici che inducono un ordinamento parziale, per cui:

$$Ax - b \geq_K 0 \quad \Leftrightarrow \quad Ax - b = 0, x \geq_K 0$$

dove K é l'insieme conico utilizzato.

Definizione Cono

Un insieme $\mathcal{K}$ é un *cono* se e solo se:

$$x \in \mathcal{K} \Rightarrow \gamma x \in \bar{\mathcal{K}} \quad \forall \gamma \geq 0$$

con $\bar{\mathcal{K}}$ la chiusura di $\mathcal{K}$.

Definizione Cono

Un insieme $\mathcal{K}$ é un *cono* se e solo se:

$$x \in \mathcal{K} \Rightarrow \gamma x \in \bar{\mathcal{K}} \quad \forall \gamma \geq 0$$

con $\bar{\mathcal{K}}$ la chiusura di $\mathcal{K}$.

Definizione Cono Convesso

Un insieme $\mathcal{K} \subseteq \mathbb{R}^m$ é un *cono* se e solo se:

$$x, y \in \mathcal{K} \Rightarrow \gamma x + \beta y \in \bar{\mathcal{K}} \quad \forall \gamma, \beta \geq 0$$

con $\bar{\mathcal{K}}$ la chiusura di $\mathcal{K}$.

Definizione Cono Convesso Puntato

Un cono convesso $\mathcal{K} \subseteq \mathbb{R}^m$ é detto *puntato* se e solo se non contiene linee.

Definizione Cono Convesso Puntato

Un cono convesso $\mathcal{K} \subseteq \mathbb{R}^m$ é detto *puntato* se e solo se non contiene linee.

Definizione Cono Proprio

Un cono $\mathcal{K} \subseteq \mathbb{R}^m$ é detto *proprio* se e solo se é:

- convesso
- puntato
- chiuso
- dotato di interno non vuoto

Coni

Esempi

- Ogni sottospazio vettoriale di $\mathbb{R}^n$;
- Ortante spazio euclideo;
- Poliedro illimitato;
- Cono Gelato
- Matrici Semidefinite;
- Matrici di Distanza Euclidee;

Ci concentreremo sul caso di coni propri per i quali si puó sviluppare una teoria di dualitá analoga alla programmazione lineare.

Dato un cono proprio $\mathcal{K} \subseteq \mathbb{R}^m$, definiamo il suo cono duale $\mathcal{K}^*$ come:

$$\mathcal{K}^* = \left\{ y \in \mathbb{R}^m \quad | \quad y^T x \geq 0 \quad \forall x \in \mathcal{K} \right\}$$

$\mathcal{K}^*$ é anch'esso proprio e risulta $(\mathcal{K}^*)^* = \mathcal{K}$.

Programmazione Conica

Definizione

PRIMALE		DUALE	
$\min_x$	$c^T x$	$\max_{y,s}$	$b^T y$
$s.t.$	$Ax - b = 0$	$s.t.$	$A^T y + s - c = 0$
	$x \in \mathcal{K}$		$s \in \mathcal{K}^*$

PRIMALE		DUALE	
$\min_x$	$c^T x$	$\max_y$	$b^T y$
$s.t.$	$Ax - b \geq_{\mathcal{K}} 0$	$s.t.$	$A^T y - c = 0$
			$y \geq_{\mathcal{K}^*} 0$

La classe di problemi é ancora molto vasta, per cui ci focalizziamo su due tipi di coni propri di interesse pratico:

Cono di Lorentz chiamato anche *Second-Order Cone* e definito come:

$$\mathbb{L}^m = \left\{ x \in \mathbb{R}^m : x_m \geq \sqrt{\sum_{i=1}^{m-1} x_i^2} \right\}$$

Cono Matrici Semidefinite Positive

$$\mathbb{S}_+^m = A \in \mathbb{R}^{m \cdot m} : A = A^T, \forall x \in \mathbb{R}^m x^T A x \geq 0$$

Uso dei Coni

Questi coni propri sono candidati perfetti per applicare gli IPM nel caso convesso seguendo le indicazioni di Nesterov e Nemirovski sulle barriere canoniche.

In particolare consideriamo di interesse un problema conico nella forma:

$$\begin{array}{ll} \min_x & c^T x \\ s.t. & Ax - b \geq_{\mathcal{K}} 0 \end{array}$$

dove il cono K é somma diretta di un numero finito di coni SDP o Lorentz:

$$\mathcal{K} = \mathbb{S}_+^{k_1} \times \mathbb{S}_+^{k_2} \ldots \mathbb{S}_+^{k_m} \times \mathbb{L}_+^{t_1} \times \mathbb{L}_+^{t_2} \ldots \mathbb{L}_+^{t_n}$$

Cono di Lorentz

Funzioni Barriera

Per il cono di Lorentz si scelga come funzione barriera:

$$B_k^L = -\log(x_k^2 - \sum_{i=1}^{k-1} x_1^2) = -\log(x^T J x)$$

con $x \in \mathbb{R}^k$ ed

$$J = \begin{Bmatrix} -1 & & & \\ & \ddots & & \\ & & -1 & \\ & & & 1 \end{Bmatrix}$$

B_k^L é 2-selfconcordant.

Cono di Semidefinito

Funzioni Barriera

Per il cono di SDP si scelga come funzione barriera:

$$B_k^S = -\log\det(X)$$

che risulta $k-$selfconcordant.

Nel caso della PL, ricordando che $X = Diag(x) \in \mathbb{R}^{n \cdot n}$, otteniamo $det(X) = \prod x_i$ da cui:

$$B_k^S = -\log\det(X) = -\log\prod_{i=1}^{n} x_i = -\sum_{i=1}^{n}\log x_i$$

Funzioni Barriera

Proprietá

Complessivamente per il cono $\mathcal{K}$ la barriera canonica assumerá la forma:

$$B(\mathcal{K}) = \sum_{i=1}^{m} B_i^S + \sum_{i=1}^{n} B_i^L$$

che risulterá essere $\theta-$selfconcordant con:

$$\theta = \sum_{i=1}^{m} \theta(B_i^S) + \sum_{i=1}^{n} \theta(B_i^L) = \sum_{i=1}^{m} k_i + 2n$$

Cammino Centrale

Proprietá

Ritroviamo la definizione del CP:

$$x_*(\mu) = \left\{x : \exists \mu \leftarrow x = \arg\min_x c^T x + \mu B(x)\right\}$$

$$s_*(\mu) = \left\{s : \exists \mu \leftarrow s = \arg\min_s -b^T y + \mu B(s)\right\}$$

In particolare useremo l'immagine del CP tramite i vincoli:

$$X_*(\mu) = Ax_*(\mu) - b$$

$$S_*(\mu) = A^T s_*(\mu) - c$$

Barriere Canoniche

Utilizzo

La struttura del CP indotta porta a due risultati:

Peso μ il peso della barriera deve decrescere ad un tasso almeno lineare, come ad esempio:

$$\mu_{k+1} = \mu_k \left(\frac{\theta}{\sqrt{\theta + 0.1}} \right)$$

Newton se μ viene aggiornata adeguatamente, un passo del metodo di Newton sará in grado di generare una soluzione tale che:

$$\begin{cases} x_k \in \mathcal{N}_{cp} \Rightarrow x_{k+1} \in \mathcal{N}_{cp} \\ \\ x_k, x_{k+1} \in S \end{cases}$$

Proprietá CP

Teorema (Legame CP Primale/Duale [Ben-Tal and Nemirovski, 2001])

Scelto $\mu > 0$, i corrispondenti punti $X_*(\mu), S_*(\mu)$ (strettamente ammissibili) del CP soddisfano le seguenti relazioni:

- $X_*(\mu) = -\mu \nabla B(S_*(\mu))$
- $S_*(\mu) = -\mu \nabla B(X_*(\mu))$

nel caso LP, essendo $\nabla B(Y) = -Y^{-1}$ otteniamo:

$$X_*(\mu)S_*(\mu) = \mu I \quad \Rightarrow \quad X_*(\mu)S_*(\mu)e = \mu e$$

questa proprietá é detta *Augmented Complementary Slackness*.

Proprietá CP

Gap di Dualitá

Dato un valore μ al peso barriera ed individuata la coppia $(X_*(\mu), S_*(\mu))$ di punti sul CP, il gap di dualitá risulta:

$$\Delta G(X_*(\mu), S_*(\mu)) = \mu\theta = \epsilon$$

ovvero $(X_*(\mu), S_*(\mu))$ é una soluzione $\epsilon-$approssimata strettamente ammissibile.

Se x_k é sufficientemente vicino al CP (secondo una qualche norma), allora l'errore che commettiamo risulta:

$$c^T x_k - \min_{y \in X} c^T y \leq 2\mu\theta$$

ovvero decresce linearmente con il parametro μ!

IPM e Polinomialitá

Complessitá - 2

Una volta individuata una coppia (x_0, μ_0) vicina al punto $x(\mu_0)$ sul CP, il metodo impiega non piú di

$$K = \mathcal{O}(1)\sqrt{\theta}\log\left(1 + \frac{\mu_0\theta}{\epsilon}\right)$$

passi per ottenere una soluzione $\epsilon-$ottima strettamente ammissibile.

Uso della programmazione conica

Molti problemi pratici sono formulabili attraverso la programmazione conica, utilizzando:

Second Order Conic Optimization - SOCP con solo coni di Lorentz

Semidefinite Programming - SDP con solo coni semidefiniti

SDP generalizza SOCP, ma trattarle separatemente permette risparmi sia implementativi (come la gestione della memoria) sia algebrici (direzione di Newton e misure di centralitá ad hoc) che danno grande beneficio pratico.

Second-Order Conic Programming

Definizione [Alizadeh and Goldfarb, 2003]

PRIMALE		DUALE	
$\min_x$	$\sum c_i^T x_i$	$\max_{y,s}$	$b^T y$
$s.t.$	$\sum A_i x_i = b$	$s.t.$	$A_i^T y + s_i = c_i$
	$x_i \succeq_{\mathcal{Q}} 0 \quad i = 1 \ldots k$		$s_i \succeq_{\mathcal{Q}} 0 \quad i = 1 \ldots k$

con $\succeq_{\mathcal{Q}}$ é la disuguaglianza conica

Second-Order Conic Programming

Applicazioni

- Ottimizzazione lineare robusta
- Controllo ottimo TD
- Minimizzazione di somme di norme
- Sintesi di antenne radio
- Equilibrio di forze

Esempio SOCP

Ottimizzazione Lineare Robusta

Supponiamo di lavorare con un modello lineare:

$$\begin{array}{ll} \min_x & c^T x \\ s.t. & Ax - b \geq 0 \end{array}$$

ma una certa componente di imprecisione/aleatorietà può caratterizzare c, b, A, ovvero siamo interessati alle soluzioni del piú generale:

$$\begin{array}{ll} \min_{x,z} & z \\ s.t. & c^T x \leq z \\ & Ax - b \geq 0 \\ & A, b, c \in \mathcal{U} \end{array}$$

dove $\mathcal{U}$ rappresenta l'insieme delle possibili realizzazioni di c, b, A.

Esempio SOCP

Al variare della struttura di $\mathcal{U}$ il problema cambia caratteristiche. Supporremo che c, b, A siano indipendenti e possano variare il loro valore all'interno di ellissoidi centrati nei valori nominali c_*, b^*, A^*:

$$\mathcal{U} = \left\{ c, b, A \quad | \quad \exists u_0, u_1 \ldots u_m, \|u_k\| \leq 1, \left[\begin{array}{l} c = c_* + P_0 u_0 \\ b_i = b_i^* + P_i u_i, i = 1..m \\ a_i = a_i^* + P_i u_i, i = 1..m \end{array} \right] \right\}$$

con P_k matrice delle perturbazioni.

Esempio SOCP

Ottimizzazione Lineare Robusta - 3

Una soluzione x del problema é robusta se per $i = 1..m$:

$$0 \leq \min_{u,\|u\|\leq 1} \left\{ a_i^T(u_i)x - b_i(u_i) : \left[\begin{array}{c} b_i = b_i^* + P_i u_i, i = 1..m \\ a_i = a_i^* + P_i u_i, i = 1..m \end{array} \right] \right\} =$$

$$= a_i^* x - b_i^* + \min_{u,\|u\|\leq 1} \{xPu_i - Pu_i\} =$$

$$= a_i^* x - b_i^* + \left\| P_i^T \begin{pmatrix} x \\ -1 \end{pmatrix} \right\|_2$$

Esempio SOCP

Ottimizzazione Lineare Robusta - 3

Con analogo ragionamento si ottiene un vincolo conico per la funzione obiettivo.

$$-t \leq -\min_{u,\|u\|\leq 1} \left\{ c(u_i)^T x : c(u_i) = c_* + P_0 u_i \right\} =$$

$$= -c_*^T x - \min_{u,\|u\|\leq 1} x P_0 u_i =$$

$$= -c_*^T x - \left\| P_0^T x \right\|_2$$

Esempio SOCP

Il problema robusto in forma conica prende quindi la forma:

$$\min_{x,t} \quad t$$

$$s.t. \quad \begin{cases} a_i^* x - b_i^* + \left\| P_i^T \begin{pmatrix} x \\ -1 \end{pmatrix} \right\|_2 \geq 0 \quad i = 1 \dots m \\ c_*^T x + \left\| P_0^T x \right\|_2 \leq t \end{cases}$$

IPM per SOCP

Condizioni di ottimalitá

Da quanto visto in precedenza, uno schema IPM per SOCP puó avere complessitá polinomiale, scriviamo quindi le KKT per derivare il passo di Newton:

$$\begin{cases} \sum_{i=1}^{m} A_i x_i - b = 0 \\ A_i^T y + s_i - c_i = 0 \quad i = 1 \dots m \\ (x_i, s_i \succ_{\mathcal{Q}} 0) \\ x_i \circ s_i = 2\mu e \end{cases}$$

dove

$$x \circ s = \hat{X}\hat{S}e \quad \hat{Y} = \begin{pmatrix} y_0 & \hat{y}^T \\ \hat{y} & y_0 I \end{pmatrix} \quad \hat{y} = (y_1 \dots y_m)^T$$

IPM per SOCP

Sistema di Newton

$$\begin{cases} \sum_{i=1}^{m} A_i \Delta x_i = b - \sum_{i=1}^{m} A_i x_i \\ A_i^T \Delta y + \Delta s_i = c_i - A_i^T y - s_i \qquad i = 1..m \\ x_i \circ \Delta s_i + s_i \circ \Delta s_i = 2\mu e - x_i \circ s_i \end{cases}$$

La direzione di Newton che otteniamo per il sistema precedente risulta:

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \hat{Z} & 0 & \hat{X} \end{pmatrix} \begin{pmatrix} \Delta_x \\ \Delta_y \\ \Delta_s \end{pmatrix} = - \begin{pmatrix} Ax - b \\ A^T y + s - c \\ X \circ S - 2\mu e \end{pmatrix}$$

Schemi PF possono essere implementati con opportune misure di centralitá ed i corrispondenti intorni del CP:

$$\mathcal{N}_\bullet(\gamma) = \{(x, y, s) : (x, y, s)\text{é strettamente ammissibile}, d_\bullet \leq \gamma\mu\}$$

dove $d_\bullet$ e' una norma opportuna (si veda [Alizadeh and Goldfarb, 2003]). L'intorno é caratterizzato da un ampiezza γ.

IPM per SOCP

Note Finali

- Molto lavoro occorre per trovare formulazioni e trasformazioni che permettano un costo computazionale ragionevole.
- Ci sono molti dettagli che riguardano l'instabilitá numerica del SOCP nella formulazione di base ed affrontate con l'uso di opportune fattorizzazioni.
- Fissata l'ampiezza dell'intorno γ ed il tasso di decremento σ, la complessita', in termini di iterazioni necessarie alla riduzione del gap di dualitá di un fattore costante, é circa $\mathcal{O}(m^k), k \geq 1$
[Alizadeh and Goldfarb, 2003].

Semi-Definite Programming

Definizione

PRIMALE

$$\min_x \quad c^T x$$

$$s.t. \quad F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0$$

$$x, c \in \mathbb{R}^m$$

$$F_i = F_i^T \in \mathbb{R}^{n \cdot n}$$

DUALE

$$\max_Z \quad -tr(F_0 Z)$$

$$s.t. \quad tr(F_i Z) = c_i \quad i = 1 \ldots m$$

$$Z \succeq 0$$

Semi-Definite Programming

Applicazioni

I campi di applicazione per i problemi SDP sono moltissimi ([Ben-Tal and Nemirovski, 2001]):

- Sintesi di Filtri
- MAXCUT (utilizzando rilassamenti SDP)
- Analisi di stabilitá
- Design strutturale
- Design di circuiti elettronici
- Problemi di embedding geometrico
- Pattern separation

Semi-Definite Programming

Rappresentazioni

Un aspetto delicato per i problemi SDP é come si possano convertire problemi all'apparenza molto diversi in forma di LMI.

- spesso si puó direttamente lavorare sulla definizione del problema;

Semi-Definite Programming

Rappresentazioni

Un aspetto delicato per i problemi SDP é come si possano convertire problemi all'apparenza molto diversi in forma di LMI.

- spesso si puó direttamente lavorare sulla definizione del problema;
- un passo comune é quello di spostare la f.o. come vincolo, riportandola quindi a lineare;

Semi-Definite Programming

Rappresentazioni

Un aspetto delicato per i problemi SDP é come si possano convertire problemi all'apparenza molto diversi in forma di LMI.

- spesso si puó direttamente lavorare sulla definizione del problema;
- un passo comune é quello di spostare la f.o. come vincolo, riportandola quindi a lineare;
- molti vincoli non lineari si possono trasformare in LMI tramite risultati come il lemma di Schur.

Semi-Definite Programming

Rappresentazioni

Un aspetto delicato per i problemi SDP é come si possano convertire problemi all'apparenza molto diversi in forma di LMI.

- spesso si puó direttamente lavorare sulla definizione del problema;
- un passo comune é quello di spostare la f.o. come vincolo, riportandola quindi a lineare;
- molti vincoli non lineari si possono trasformare in LMI tramite risultati come il lemma di Schur.

La riformulazione non é univoca e puó drasticamente influenzare la prestazioni!

Esempio SDP

Design Strutturale - 1

- k elementi elastici di sezione x_i e lunghezza nota l_i;
- p giunti di snodo dove applicare forze f_i con conseguente scostamento d_i dalla posizione iniziale;
- energia meccanica da minimizzare $E = \frac{1}{2} f^t d$
- limite sul volume totale v fissata la lunghezza l degli elementi e sulle sezioni utilizzabili;
- il legame forze-scostamenti é regolato dalla relazione $f = A(x)d = \sum_{i=1}^{k} A_i x_i d \quad , \quad A_i = A_i^T$;

$$\begin{aligned} \min_{d,x} \quad & f^t d \\ & f = A(x)d \\ & \sum l_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \overline{x}_i \end{aligned}$$

Esempio SDP

Design Strutturale - 2

Sostituendo nell'obiettivo:

$$\begin{aligned} \min_{d,x} \quad & f^T A^{-1}(x) f \\ & \sum_{i=1}^{k} l_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \overline{x}_i \quad i = 1 \dots k \end{aligned}$$

$$\begin{aligned} \min_{d,x,t} \quad & t \\ & f^T A^{-1}(x) f \leq t \\ & \sum_{i=1}^{k} l_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \overline{x}_i \quad i = 1 \dots k \end{aligned}$$

Esempio SDP

Design Strutturale - 3

Applichiamo il Lemma di Schur che sancisce l'equivalenza:

$$A \succeq 0 \quad \Leftrightarrow \quad D - CB^{-1}C^T \succeq 0$$

con

$$A = A^T = \begin{pmatrix} B & C^T \\ C & D \end{pmatrix} \quad , \quad B \in \mathbb{S}$$

nel nostro caso otteniamo:

$$f^T A^{-1}(x) f \leq t \Rightarrow \begin{bmatrix} t & f^T \\ f & A(x) \end{bmatrix} \succeq 0$$

Esempio SDP

Design Strutturale - 4

Otteniamo infine:

$$\min_{d,x,t} \quad t$$

$$s.t. \quad \begin{cases} \begin{bmatrix} t & f^t \\ f & A(x) \end{bmatrix} \succeq 0 \\ \sum_{i=1}^{k} l_i x_i \leq v \\ \underline{x}_i \leq x_i \leq \overline{x}_i \quad i = 1 \dots k \end{cases}$$

IPM per SDP

Condizioni KKT primali/duali

$$\begin{cases} Y = F_0 + \sum_{i=1}^{k} x_i F_i \\ tr(F_i Z) = c_i \qquad\qquad i = 1 \dots k \\ (Y, Z \succeq 0) \\ ZY = \mu I \end{cases}$$

da cui la direzione di Newton come soluzione di:

$$\begin{cases} \sum_{i=1}^{m} x_i F_i = 0 \qquad\qquad i = 1 \dots k \\ tr(F_i \Delta Z) = 0 \qquad\qquad i = 1 \dots k \\ Z\Delta F + F\Delta Z = \mu I - ZF \end{cases}$$

IPM per SDP

Schemi Path-Following

Anche per la SDP si possono introdurre schemi PF, simili a quelli per SOCP, utilizzando intorni del CP tipo:

$$\begin{cases} \mathcal{N}_{-\infty} = \{(Z, x, F) \text{ strettamente ammissibile} \quad : \quad \delta_{-\infty} \leq \gamma\mu(Z, F)\} \\ \delta_{-\infty} = \max_{i=1\ldots k} (\mu - \lambda_i(ZF)) \\ \mu(Z, F) = \dfrac{tr(ZF)}{k} \end{cases}$$

con $\lambda_i(A)$ l' i-esimo autovalore della matrice A.

IPM per SDP

Complessitá

Come per tutti gli IPM, la complessitá dipende da molti fattori come l'ampiezza dell'intorno del CP, la politica di riduzione di μ e cosí via.

IPM per SDP

Complessitá

Come per tutti gli IPM, la complessitá dipende da molti fattori come l'ampiezza dell'intorno del CP, la politica di riduzione di μ e cosí via.

Ad esempio, gli IPM SS-PF, con intorno $\mathcal{N}_{-\infty}$ mostrano una complessitá per raggiungere una soluzione $\epsilon-$ottima (in termini di iterazioni) di $\mathcal{O}(\left(\sqrt{n}\frac{1}{\epsilon}\right))$.

IPM per SDP

Complessitá

Come per tutti gli IPM, la complessitá dipende da molti fattori come l'ampiezza dell'intorno del CP, la politica di riduzione di μ e cosí via.

Ad esempio, gli IPM SS-PF, con intorno $\mathcal{N}_{-\infty}$ mostrano una complessitá per raggiungere una soluzione $\epsilon-$ottima (in termini di iterazioni) di $\mathcal{O}(\left(\sqrt{n}\frac{1}{\epsilon}\right))$.

Il costo in termini di operazioni elementari risulta molto piú alto e dipende fortemente dalla struttura del problema e delle matrici che lo contraddistinguono.

IPM per SDP

Complessitá

Come per tutti gli IPM, la complessitá dipende da molti fattori come l'ampiezza dell'intorno del CP, la politica di riduzione di μ e cosí via.

Ad esempio, gli IPM SS-PF, con intorno $\mathcal{N}_{-\infty}$ mostrano una complessitá per raggiungere una soluzione $\epsilon-$ottima (in termini di iterazioni) di $\mathcal{O}(\left(\sqrt{n}\frac{1}{\epsilon}\right))$.

Il costo in termini di operazioni elementari risulta molto piú alto e dipende fortemente dalla struttura del problema e delle matrici che lo contraddistinguono.

Schemi Predictor/Corrector vengono implementati per migliorare le prestazioni nella maggior parte dei codici disponibili.

IPM per SDP

Note

- Riformulare un problema come SDP porta spesso ad un aumento in dimensione sia come variabili e come vincoli.
- Occorre una gestione oculata di memoria e dell'algebra lineare necessaria.
- I risolutori traggono grande beneficio da fattorizzazioni e precondizionamenti, in particolare operazioni di scaling per migliorare la struttura della matrice di Newton.

Per dettagli si veda ad esempio [Vandenberghe and Boyd, 1996].

Implementazioni SDP-IPM

Lista

Codici disponibili per SDP:

NOME	URL	NOTE
Sedumi	http://sedumi.mcmaster.ca	C,Matlab
Mosek	http://www.mosek.com	Commerciale
SDPA	http://homepage.mac.com/klabtitech/sdpa-homepage/	C++, Matlab
Csdp	https://projects.coin-or.org/Csdp/	C++, COIN-OR
SBmethod	http://www-user.tu-chemnitz.de/∼helmberg/SBmethod/	C++
LOQO	http://www.princeton.edu/∼rvdb/	Commerciale
CPLEX	http://www.cplex.com/	Commerciale
PENSDP	http://www.penopt.com/pensdp.html	Commerciale
SDPT3	http://www.math.nus.edu.sg/∼mattohkc/sdpt3.html	Matlab
SDPLR	http://dollar.biz.uiowa.edu/∼burer/software/SDPLR/	C, Matlab

Implementazioni SDP-IPM

Benchmarks

Test sul benchmark SDPLIB (`http://infohost.nmt.edu/~sdplib/`) su macchine con P4 a 3.2 GHz, 4GB RDRAM e Linux-2.6.8.1.
(Il test misura il tempo totale in secondi di CPU)

solver	Csdp	DSDP	SDPA	SDPT3	SeDuMi	PENsdp
media	156.889	132.741	144.885	57.847	1262.231	250.260
mediana	35.000	137.546	150.342	59.841	1310.471	259.158
varianza	111967.411	142.122	154.548	58.719	1359.412	264.905

altri benchmarks ad esempio su `http://plato.asu.edu/bench.html`

Riassumendo

- Teoria unificata per IPM su problemi convessi (coni propri):
  - Uso delle Barriere Canoniche
  - Duality Gap decrescente col peso della barriera
  - Convergenza Polinomiale;
- Second-Order Cone Programming
- Semidefinite Programming


Alizadeh, F. and Goldfarb, D. (2003).
Second-order cone programming.
Mathematical Programming, 95(1):3–51.


Ben-Tal, A. and Nemirovski, A. (2001).
Lecture notes on modern convex optimization: analysis, algorithms, and engineering applications.
SIAM, Philadelphia.


Dattorro, J. (2008).
Convex Optimization and Euclidean Distance Geometry.
Meboo Publishing, Palo Alto, CA 94302 USA.


Renegar, J. (2001).
A mathematical view of interior-point methods in convex optimization.
MPS-SIAM series on Optimization. SIAM, Philadelphia (USA).


Vandenberghe, L. and Boyd, S. (1996).
Semidefinite programming.
SIAM Review, 38(1):49–95.


Vanderbei, R. J. (2001).
Linear Programming: Foundations and Extensions.
2 edition.


Ye, Y. (1997).
Interior Point Algorithms Theory and Analysis.
Wiley Interscience.