

Scalatura di matrici

Luca Ferragina

28 giugno 2017

Indice

1	Introduzione	2
1.1	Definizioni di base	2
1.2	Applicazioni	3
2	Scalatura troncata come problema di ottimizzazione	6
2.1	Problema primale	7
2.2	Problema duale	8
3	Scalatura di equivalenza	11
3.1	Relazione con la scalatura troncata	11
3.2	Relazione del duale con gli altri problemi	13
3.3	Un risultato teorico di esistenza	15
4	Algoritmo di scalatura	19
4.1	Creazione dell'algoritmo per la scalatura non troncata.	19
4.2	Sperimentazione numerica	20

Capitolo 1

Introduzione

Ciò che si intende informalmente per *bilanciamento* di una matrice è il seguente

Problema 1. *Data in input una matrice rettangolare A calcolare una matrice X "vicina" ad A che soddisfi certe condizioni lineari sui propri elementi.*

Come vedremo in seguito possono essere modellizzati in questo maniera diversi problemi riguardanti le applicazioni più disparate. Chiaramente una definizione di questo tipo è mal posta e dipende da cosa si intenda con il termine "vicina" e da quali condizioni lineari vengano imposte al problema.

Per tutto l'elaborato tratteremo di problemi in cui la relazione di vicinanza di X ad A è espressa tramite l'esistenza di matrici diagonali D_1 e D_2 di taglia opportuna, con elementi positivi lungo la diagonale e tali che $X = D_1 A D_2$. Con questa richiesta il problema prende il nome di *scalatura* di una matrice.

Vedremo diversi tipi di problemi di scalatura che saranno classificati dalle dimensioni delle matrici in gioco e dai vincoli lineari imposti, quella che segue è una serie di definizioni necessaria alla loro formulazione.

1.1 Definizioni di base

Definizione 1.1. Una matrice quadrata $X \in \mathbb{R}^{n \times n}$ è detta **bilanciata** se

$$\sum_{j=1}^n X_{ij} = \sum_{j=1}^n X_{ji} \quad \text{per } i = 1, \dots, n$$

Osservazione 1.1. La condizione di bilanciamento può essere scritta in forma compatta come

$$X \mathbf{1}_n = X^T \mathbf{1}_n$$

dove $\mathbf{1}_n \in \mathbb{R}^n$ è il vettore le cui componenti sono tutte uguali ad 1.

Definizione 1.2. Date due matrici quadrate a valori reali non negativi $A, L \in \mathbb{R}^{n \times n}$ e una matrice $U \in (\mathbb{R} \cup \{+\infty\})^{n \times n}$ a valori non negativi, la tripla (A, L, U) si dice **consistente** se

- $0 \leq L \leq U$
- esiste $X \in \mathbb{R}^{n \times n}$ tale che $L \leq X \leq U$ e $X_{ij} = 0$ ogni volta che $A_{ij} = 0$.

Con le stesse ipotesi su L, U chiamiamo **operatore di troncamento** rispetto a L ed U l'applicazione $T_{[L,U]} : \mathbb{R}^{n \times n} \mapsto \mathbb{R}^{n \times n}$ definita da

$$T_{[L,U]}(Y)_{ij} = T_{[L_{ij}, U_{ij}]}(Y_{ij})$$

dove con $T_{[l,u]}(y)$ indichiamo la funzione di troncamento di un numero reale y all'intervallo $[l, u]$:

$$T_{[l,u]}(y) = \begin{cases} l & \text{se } y \leq l \\ y & \text{se } l \leq y \leq u \\ u & \text{se } u \leq y \end{cases}$$

Problema 2 (Scalatura troncata). *Data una tripla consistente (A, L, U) , trovare una matrice diagonale $D = \text{diag}(d_1, \dots, d_n)$ tale che $d_i > 0$ per ogni i e che $X = T_{[L,U]}(DAD^{-1})$ sia bilanciata.*

Problema 3 (Scalatura di similitudine). *Data una matrice quadrata $A \in \mathbb{R}^{n \times n}$ con elementi non negativi, trovare una matrice diagonale $D = \text{diag}(d_1, \dots, d_n)$ tale che $d_i > 0$ per ogni i e che $X = DAD^{-1}$ sia bilanciata*

Problema 4 (Scalatura di equivalenza). *Dati una matrice rettangolare $A \in \mathbb{R}^{n \times m}$ con elementi non negativi e un vettore $v \in \mathbb{R}^{n+m}$ con elementi positivi tale che $\sum_{i=1}^m v_i = \sum_{i=m+1}^{n+m} v_i$, trovare $D_1 \in \mathbb{R}^{m \times m}$ e $D_2 \in \mathbb{R}^{n \times n}$ diagonali con elementi diagonali positivi tali che $X = D_1AD_2$ risolve*

$$\sum_{j=1}^m X_{ij} = v_i \quad i = 1, \dots, m \quad \sum_{i=1}^n X_{ij} = v_{m+j} \quad j = 1, \dots, n$$

Esamineremo nel seguito in dettaglio le relazioni fra i tre problemi che abbiamo definito, per il momento ci limitiamo ad osservare che il problema 3 si può ottenere facilmente come caso particolare del problema 2 ponendo L uguale alla matrice nulla e U uguale alla matrice con tutti gli elementi pari a $+\infty$.

Osservazione 1.2. Nel problema 2 la richiesta che (A, L, U) sia consistente elimina le istanze non ammissibili.

1.2 Applicazioni

Quelle che seguono sono solo alcune dei tanti problemi reali che in letteratura sono stati approcciati tramite modelli di scalatura di matrici (per una visione approfondita di queste ed altre applicazioni si rimanda a [6] e [7]).

Economia: Matrice di Contabilità Sociale. Una matrice di contabilità sociale (SAM dall'inglese Social Accounting Matrix) è una matrice quadrata A che rappresenta il flusso di fondi attraverso gli agenti di un dato mercato. Poichè solitamente questo strumento viene usato da istituti governativi per tenere sotto controllo l'economia interna possiamo pensare agli indici di A come agli agenti (aziende, negozi, istituzioni, famiglie, etc.) nel mercato nazionale e alle componenti a_{ij} di A come al flusso di denaro dall'agente i all'agente j in un dato lasso di tempo. Chiaramente la somma degli elementi sulla riga i -esima indica il flusso in uscita dall'agente i mentre la somma degli elementi sulla colonna n indica il flusso in entrata.

Un problema comune nella gestione di queste matrici deriva dal fatto che i dati rappresentati dagli elementi a_{ij} sono ottenuti in maniera non precisa, perchè frutto di stime o perchè semplicemente ottenuti in maniera non univoca da diversi organi delegati al calcolo. La spesa e le

entrate totali di ogni agente al contrario sono più accurati ed aggiornati, perchè generalmente disponibili tramite fonti governative.

Il modello può essere descritto come segue: data una matrice di contabilità calcolarne una "simile" che abbia somma degli elementi delle righe e delle colonne uguali a certi valori fissati. Se questa similitudine è cercata tramite modelli di scalatura ci troviamo di fronte un problema analogo al problema 4.

Un'altro interesse riguardo le matrici di contabilità risiede nello studio dell'equilibrio del mercato, ovvero una situazione ideale in cui le entrate di ogni agente si equivalgono con le uscite. In questo contesto quindi cerchiamo trasformazioni della matrice che la rendano tale da soddisfare condizioni tipo quella del problema 2.

Pianificazione dei trasporti. Immaginiamo di avere a disposizione un insieme di m origini e n destinazioni e di voler predire il numero X_{ij} di spostamenti che avverranno da i a j in un dato intervallo di tempo. Supponiamo di avere a disposizione il numero totale di spostamenti totali v_i in uscita da i per $1 \leq i \leq m$, il numero di spostamenti totali v_{m+j} in entrata in j per $1 \leq i \leq n$ è una stima, anche grossolana, A_{ij} del valore cercato (ad esempio A può essere la matrice ottenuta nell'intervallo precedente). Se assumiamo che X si ottenga da A tramite il prodotto a destra e a sinistra per matrici diagonali, otteniamo un'istanza del problema 4, in effetti i problemi di pianificazione del traffico ([3]) sono stati tra i primi ad essere modellizzati tramite problemi di scalatura di matrici.

PageRank e passeggiate aleatorie. La versione originale del famoso algoritmo di PageRank fondamentalmente lavora prendendo in input la matrice A di adiacenza di un grafo e scalando le sue colonne in modo che la matrice ottenuta AD^{-1} sia stocastica per colonne. Il ranking è poi calcolato considerando il vettore di probabilità invariante ottenuto dalla catena di Markov associata a quest'ultima matrice. Osserviamo che in questo approccio il calcolo della scalatura è banale e il grosso del lavoro consiste nel calcolo del vettore invariante.

Alcuni autori hanno descritto ([2]), ispirandosi ai problemi di traffico, un approccio differente al problema del PageRank che consiste nel cercare una scalatura $X = D_1 A D_2$ della matrice di adiacenza che sia doppiamente stocastica. In questo modo se chiamiamo $D_1 = \text{diag}(r_1, \dots, r_n)$ e $D_2 = \text{diag}(c_1, \dots, c_m)$, i vettori r e c rappresenteranno (visti al rovescio) due ranking diversi delle pagine più "importanti" rispettivamente in entrata e in uscita, ovvero se una pagina i contiene link a pagine "importanti" allora il valore c_i sarà relativamente basso, similmente se molte pagine "importanti" contengono link ad i , allora il valore di r_i sarà relativamente basso.

Analisi Numerica: Precondizionamento. Supponiamo di voler calcolare gli autovalori della matrice

$$A = \begin{bmatrix} 1 & 0 & 10^{-4} \\ 1 & 1 & 10^{-2} \\ 10^4 & 10^2 & 1 \end{bmatrix}$$

com'è noto i metodi numerici che abbiamo a disposizione hanno tipicamente un errore all'indietro dell'ordine di $\eta \|A\|$, dove η è la precisione di macchina. Lavorando ad esempio con la norma di Frobenius, si ottiene $\|A\|_F \sim 10^4$ per cui l'errore è molto amplificato. Scalando la matrice A per mezzo di $D = \text{diag}(10^2, 1, 10^{-2})$ si ottiene

$$B = DAD^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 10^{-2} & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Osserviamo che la matrice B è approssimativamente bilanciata e che $\|B\|_F \sim 2.6$, quindi qualsiasi algoritmo applicato a B piuttosto che ad A avrà un errore all'indietro contenuto.

In generale si può mostrare ([1]) che la scalatura bilanciata di una matrice ha sempre norma minore della matrice stessa, ha quindi senso risolvere un problema tipo il problema 2 prima di effettuare il calcolo numerico degli autovalori di una matrice.

Capitolo 2

Scalatura troncata come problema di ottimizzazione

Prima di descrivere la modellizzazione della scalatura come problema di ottimizzazione, andiamo ad elencare alcune notazioni ad alcuni strumenti relativi alla teoria dei grafi che ci saranno utili successivamente.

Un *grafo* (orientato) è una coppia (V, E) dove V è un insieme finito di *nodi* e $E \subseteq V \times V$ è un insieme di *archi*. Useremo la notazione $e = (i, j)$ per indicare l'arco uscente dal nodo i ed entrante nel nodo j . Per ogni nodo $i \in V$ definiamo gli insiemi degli archi uscenti da i

$$\delta^+(i) = \{e \in E \text{ tali che esiste } j \in V \text{ per cui } e = (i, j)\}$$

ed entranti in i

$$\delta^-(i) = \{e \in E \text{ tali che esiste } j \in V \text{ per cui } e = (j, i)\}$$

La *matrice di incidenza* di (V, E) è la matrice M le cui righe sono indicizzate da V e le cui colonne sono indicizzate da E tale che

$$M_{i,e} = \begin{cases} +1 & \text{se } e = (i, j) \text{ per } j \neq i \\ -1 & \text{se } e = (j, i) \text{ per } j \neq i \\ 0 & \text{altrimenti} \end{cases}$$

In generale per un insieme finito S , usiamo la notazione \mathbb{R}^S per indicare l'insieme delle funzioni $x : S \mapsto \mathbb{R}$ riferendoci ad esse come se fossero "vettori", cioè chiamiamo x_s il valore della funzione x in $s \in S$.

Adoperando questa notazione, chiamiamo *funzione peso* per (V, E) un elemento $a \in \mathbb{R}^E$ e definiamo *grafo pesato* la tripla (V, E, a) dove a è una funzione peso.

Infine, data una matrice quadrata A con elementi non negativi definiamo il *grafo associato* ad A , e lo indichiamo con $G(A)$, il grafo pesato (V, E, a) dove

- $V = \{1, \dots, n\}$
- $E = \{(i, j) \in V \times V : A_{ij} > 0\}$
- $a_e = A_{ij}$ per $e = (i, j) \in E$

Osservazione 2.1. Si vede facilmente che le matrici non negative sono in corrispondenza biunivoca con i grafi diretti con pesi positivi.

2.1 Problema primale

Concentriamo la nostra attenzione sul problema 2 con l'intento di trasformarlo in un problema di ottimizzazione; il primo passo è considerare, data la tripla consistente (A, L, U) , il grafo pesato $G = G(A) = (V, E, a)$ associato ad A .

Poichè la matrice X deve rispettare il vincolo $X_{ij} = 0$ quando $A_{ij} = 0$, il vettore incognito sarà $x \in \mathbb{R}^E$ tale che $x_e = X_{ij} = T_{[L_{ij}, U_{ij}]}(d_i A_{ij} d_j^{-1})$ con $e = (i, j)$.

Definiamo ora i vettori $a, l, u \in \mathbb{R}^E$ associati alle matrici A, L, U tali che $a_e = A_{ij}$, $l_e = L_{ij}$ e $u_e = U_{ij}$ per $e = (i, j) \in E$, il vincolo $L \leq X \leq U$ si traduce in

$$l \leq x \leq u$$

mentre il vincolo di bilanciamento si esprime tramite l'equazione

$$Mx = 0$$

dove M è la matrice di incidenza del grafo G ; infatti si può osservare che nella riga i -esima di M compare $+1$ per i nodi e relativi agli elementi di A non nulli sulla riga i -esima, e compare -1 per i nodi e relativi agli elementi di A non nulli sulla colonna i -esima, per cui dire che la somma della colonna i -esima di X è uguale alla somma della riga i -esima di X è equivalente a dire che il prodotto scalare del vettore x per l' i -esima riga di M è nullo.

Al fine di costruire la funzione obiettivo diamo alcune definizioni preliminari:

- dato $\alpha \in \mathbb{R}$, chiamiamo *entropia* la funzione $Ent_\alpha : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\}$ tale che:

$$Ent_\alpha(x) = \begin{cases} x[\ln(\frac{x}{\alpha}) - 1] & \text{se } x > 0 \\ 0 & \text{se } x = 0 \\ +\infty & \text{se } x < 0 \end{cases}$$

- dato $[a, b]$ intervallo reale, chiamiamo *indicatrice* di $[a, b]$ la funzione $\chi_{[a,b]} : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\}$

$$\chi_{[a,b]}(x) = \begin{cases} 0 & \text{se } x \in [a, b] \\ +\infty & \text{altrimenti} \end{cases}$$

Relativamente al nostro problema consideriamo per ogni $e \in E$ la seguente funzione

$$f_e(x) = Ent_{a_e}(x) + \chi_{[l_e, u_e]}(x) = \begin{cases} x[\ln(\frac{x}{a_e}) - 1] & \text{se } x \in [l_e, u_e] \\ +\infty & \text{altrimenti} \end{cases}$$

e definiamo la funzione obiettivo del problema primale come

$$f(x) = \sum_{e \in E} f_e(x_e)$$

Si può osservare come la funzione $f_e(x)$ assuma valore minimo nel punto $x = a_e$, per cui minimizzare $f(x)$ nell'insieme composto dai vincoli $Mx = 0$ e $l \leq x \leq u$ vuol dire intuitivamente trovare una matrice X (del tipo $T_{[L,U]}(DAD^{-1})$) con elementi non troppo distanti da A e che rispetti la condizione di somma uguale su righe e colonne corrispondenti.

Osservazione 2.2. Mantenendo le proprietà che abbiamo appena descritto, avremmo potuto scegliere in alternativa alla funzione entropia una funzione lineare del tipo $|x - a_e|$ o quadratica del tipo $(x - a_e)^2$.

Queste considerazioni informali ci portano a definire il problema di ottimizzazione associato al problema 2 come segue

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Mx = 0 \\ & l \leq x \leq u \\ & x \in \mathbb{R}^E \end{aligned}$$

Osserviamo inoltre che il secondo vincolo è in realtà sovrabbondante, in quanto ogni funzione $f_e(x_e)$ non può assumere valore minimo fuori dall'intervallo $[l_e, u_e]$. Per questo motivo scriviamo il problema primale nella sua forma finale come

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Mx = 0 \\ & x \in \mathbb{R}^E \end{aligned} \tag{2.1}$$

Osservazione 2.3. Poichè $f_e(x)$ è una funzione convessa, (2.1) è classificabile come problema di ottimizzazione convessa su vincoli lineari. Inoltre è un problema separabile, queste proprietà saranno fondamentali nella prossima sezione.

2.2 Problema duale

Sia $g : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$, ricordiamo che la coniugata di Fenchel di g è la funzione convessa $g^* : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ definita da:

$$g^*(p) = \sup_{x \in \mathbb{R}^n} \{p^T x - g(x)\} = - \inf_{x \in \mathbb{R}^n} \{g(x) - p^T x\}$$

Poichè (2.1) è un problema di minimo può essere riscritto equivalentemente come

$$\inf_{x \in \mathbb{R}^E} \sup_{p \in \mathbb{R}^V} \mathcal{L}(x, p)$$

dove $\mathcal{L}(x, p) = f(x) - p^T Mx$ è la funzione Lagrangiana associata a (2.1), come sappiamo, il problema duale può essere quindi scritto come

$$\sup_{p \in \mathbb{R}^V} \inf_{x \in \mathbb{R}^E} f(x) - p^T Mx = \sup_{p \in \mathbb{R}^V} -f^*(M^T p) = - \inf_{p \in \mathbb{R}^V} f^*(M^T p)$$

e quindi risulta essere equivalente a

$$\begin{aligned} \min \quad & \Phi(p) \\ \text{s.t.} \quad & p \in \mathbb{R}^V \end{aligned} \tag{2.2}$$

Grazie alla proprietà di separabilità ([5]) del problema primale, la funzione obiettivo di (2.2) ha la seguente forma

$$\Phi(p) = f^*(M^T p) = \sum_{e \in E} f_e^*((M^T p)_e)$$

per cui per calcolarla esplicitamente basta conoscere la coniugata di Fenchel di $f_e(x)$. Per calcolare quest'ultima, bisogna trovare l'estremo superiore al variare di x di

$$\begin{cases} yx - x[\ln(\frac{x}{a_e}) - 1] & \text{se } x \in [l_e, u_e] \\ -\infty & \text{altrimenti} \end{cases}$$

Appare chiaro che l'estremo superiore possa essere raggiunto solo in $[l_e, u_e]$, inoltre osserviamo come la funzione $yx - x[\ln(\frac{x}{a_e}) - 1]$ sia concava e differenziabile in x quindi ammette un unico massimo in $[l_e, u_e]$ che viene raggiunto nel punto in cui si annulla la derivata, se questo si trova all'interno dell'intervallo, oppure nell'estremo dell'intervallo più vicino ad esso. Esplicitando l'idea precedente si ottiene il seguente risultato:

$$f_e^*(y) = \begin{cases} l_e y - l_e [\ln(\frac{l_e}{a_e}) - 1] & \text{se } a_e e^y \leq l_e \\ a_e e^y & \text{se } l_e \leq a_e e^y \leq u_e \\ u_e y - u_e [\ln(\frac{u_e}{a_e}) - 1] & \text{se } a_e e^y \geq u_e \end{cases} \quad (2.3)$$

La funzione f_e^* ha delle buone proprietà, essa è infatti continua, derivabile e con derivata continua:

$$(f_e^*)'(y) = \begin{cases} l_e & \text{se } a_e e^y \leq l_e \\ a_e e^y & \text{se } l_e \leq a_e e^y \leq u_e \\ u_e & \text{se } a_e e^y \geq u_e \end{cases}$$

Il teorema che segue stabilisce l'effettiva equivalenza fra il problema 2 e la modellizzazione che abbiamo effettuato, legando l'esistenza della sua soluzione a quella del problema duale.

Teorema 2.1. *Sia (A, L, U) una tripla consistente. I seguenti fatti sono equivalenti:*

1. *Il problema 2 ammette una soluzione data dalla matrice $D = \text{diag}(d_1, \dots, d_n)$.*
2. *In (2.2) si ottiene il minimo con il vettore $p \in \mathbb{R}^V$ dove $p_i = \ln d_i$.*

Dimostrazione. La funzione $\Phi(p)$ eredita le proprietà di f_e^* , cioè è derivabile con derivata continua e convessa, per cui una condizione necessaria e sufficiente affinché il minimo sia raggiunto nel punto p (e quindi affinché il problema duale, non vincolato, ammetta minimo in p) è che $\nabla\Phi(p) = 0$. Calcolando le derivate parziali otteniamo che

$$\begin{aligned} 0 &= \frac{\partial\Phi}{\partial p_i}(p) = \sum_{e \in E} (f_e^*)' M_{ie} \\ &= \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} (f_e^*)'(p_i - p_j) - \sum_{\substack{e \in \delta^-(i) \\ e=(j,i)}} (f_e^*)'(p_j - p_i) \\ &= \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} T_{[l_e, u_e]}(a_e e^{p_i - p_j}) - \sum_{\substack{e \in \delta^-(i) \\ e=(j,i)}} T_{[l_e, u_e]}(a_e e^{p_j - p_i}) \\ &= \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} T_{[l_e, u_e]}(d_i a_e d_j^{-1}) - \sum_{\substack{e \in \delta^-(i) \\ e=(j,i)}} T_{[l_e, u_e]}(d_j a_e d_i^{-1}) \\ &= \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} x_e - \sum_{\substack{e \in \delta^-(i) \\ e=(j,i)}} x_e = (X \mathbf{1}_n)_i - (X^T \mathbf{1}_n)_i \end{aligned}$$

□

Nel caso del problema 3, ovvero quello di scalatura non troncata, è facile vedere che il problema duale ha come funzione obiettivo

$$\Phi(p) = \sum_{\substack{e \in E \\ e=(i,j)}} a_e e^{p_i - p_j}$$

Attuando il cambio di variabili $d_i = e_i^p$ il problema duale diventa

$$\begin{aligned} \min \quad & \sum_{\substack{e \in E \\ e=(i,j)}} d_i a_e d_j^{-1} \\ \text{s.t.} \quad & d > 0 \end{aligned} \tag{2.4}$$

Dal teorema precedente si ottiene immediatamente il seguente

Corollario 2.1. *Sia $A \in \mathbb{R}^{n \times n}$ con elementi non negativi. I seguenti fatti sono equivalenti:*

1. *Il problema 3 ammette una soluzione data dalla matrice $D = \text{diag}(d_1, \dots, d_n)$.*
2. *In (2.4) si ottiene il minimo con (d_1, \dots, d_n) .*

Capitolo 3

Scalatura di equivalenza

In questo capitolo ci concentreremo maggiormente sul problema 4, in particolare andremo a mettere in luce le relazioni che lo legano ai problemi 2 e 3 dal punto di vista delle istanze vere e proprie e dal punto di vista dei problemi di ottimizzazione associati.

Definizione 3.1. Sia $A \in \mathbb{R}^{m \times n}$ una matrice con elementi non negativi e sia $v \in \mathbb{R}^{m+n}$ un vettore con elementi positivi. La coppia (A, v) si dice **consistente** se esiste $X \in \mathbb{R}^{m \times n}$ tale che

- $X_{ij} = 0$ ogni volta che $A_{ij} = 0$
- $\sum_{j=1}^n X_{ij} = v_i$ per $i = 1, \dots, m$
- $\sum_{i=1}^m X_{ij} = v_{m+j}$ per $i = 1, \dots, n$.

Chiamando $v^{(m)}$ e $v^{(n)}$ i vettori formati rispettivamente dalle prime m e dalle ultime n componenti di v , le ultime due condizioni della definizione di consistenza possono essere espresse come

$$X \mathbf{1}_n = v^{(m)} \quad X^T \mathbf{1}_m = v^{(n)}$$

Lemma 3.1. Se (A, v) è una coppia consistente, allora $\sum_{i=1}^m v_i = \sum_{i=m+1}^{m+n} v_i$.

Dimostrazione.

$$\sum_{i=1}^m v_i = \sum_{i=1}^m \sum_{j=1}^n X_{ij} = \sum_{j=1}^n \sum_{i=1}^m X_{ij} = \sum_{j=1}^n v_{m+j} = \sum_{i=m+1}^{m+n} v_i$$

□

3.1 Relazione con la scalatura troncata

In questa sezione andremo a trasformare un'istanza del problema 4 in un'istanza del problema 2 costruendo, a partire da una coppia (A, v) , un grafo pesato (V, E, a) e due vettori (l, u) tali che le matrici associate (A', L, U) producano l'istanza desiderata.

Definiamo il grafo (G, V) come segue

$$V = \{0, 1, 2, \dots, m+n\} \quad E = E_0 \cup E_1$$

dove

$$E_0 = \{(0, i) : i = 1, \dots, m\} \cup \{(i, 0) : i = m + 1, \dots, m + n\}$$

$$= \{(0, 1), (0, 2), \dots, (0, m), (m + 1, 0), \dots, (m + n, 0)\}$$

e

$$E_1 = \{(i, j + m) : i = 1, \dots, m \quad j = 1, \dots, n \quad A_{ij} > 0\}$$

Definiamo i vettori a, l, u distinguendo due casi

- Se $(i, j + m) \in E_1$ allora poniamo

$$a_e = A_{ij} \quad l_e = 0 \quad u_e = +\infty$$

- Se $(i, j) \in E_0$ allora poniamo

$$a_e = l_e = u_e = v_i \quad 1 \leq i \leq m \quad \text{per } e = (0, i)$$

$$a_e = l_e = u_e = v_i \quad m + 1 \leq i \leq m + n \quad \text{per } e = (i, 0)$$

Osservazione 3.1. La matrice $A' \in \mathbb{R}^{(m+n+1) \times (m+n+1)}$ è della forma

$$A' = \left[\begin{array}{c|c|c} 0 & (v^{(m)})^T & (0^{(n)})^T \\ \hline 0^{(m)} & 0^{(m \times m)} & A \\ \hline 0^{(n)} & 0^{(n \times m)} & 0^{(n \times n)} \end{array} \right]$$

dove $0^{(n \times m)} \in \mathbb{R}^{n \times m}$ e $0^{(n)} \in \mathbb{R}^n$ sono rispettivamente la matrice e il vettore nulli.

Lemma 3.2. *Se (A, v) è una coppia consistente nel senso del problema 4, allora (A', L, U) è una tripla consistente nel senso del problema 2.*

Dimostrazione. Consideriamo la matrice $X' \in \mathbb{R}^{(m+n+1) \times (m+n+1)}$ tale che

$$X' = \left[\begin{array}{c|c|c} 0 & (v^{(m)})^T & (0^{(n)})^T \\ \hline 0^{(m)} & 0^{(m \times m)} & X \\ \hline 0^{(n)} & 0^{(n \times m)} & 0^{(n \times n)} \end{array} \right]$$

dove $X \in \mathbb{R}^{m \times n}$ è tale che $X_{ij} = 0$ quando $A_{ij} = 0$ e inoltre

$$X \mathbf{1}_n = v^{(m)} \quad X^T \mathbf{1}_m = v^{(n)}$$

Per il lemma 3.1 si ha immediatamente che la somma delle componenti della prima riga è uguale alla somma delle componenti della prima colonna. Per $1 \leq i \leq m$ si ha che

$$\sum_{j=0}^{m+n} X'_{ij} = \sum_{j=0}^m X'_{ij} = v_i = \sum_{j=0}^{m+n} X'_{ji}$$

Per $m + 1 \leq i \leq m + n$ si ha che

$$\sum_{j=0}^{m+n} X'_{ij} = v_i = \sum_{j=1}^m X_{ji} = \sum_{j=0}^{m+n} X'_{ji}$$

Da questo segue che X' prova la consistenza di (A', L, U) poichè ovviamente vale che $X'_{ij} = 0$ quando $A'_{ij} = 0$. \square

Teorema 3.1. Sia (A, v) una coppia consistente e sia (A', L, U) la tripla ad essa associata relativa al problema 2. I seguenti fatti sono equivalenti:

1. Il problema 4 per la coppia (A, v) ammette una soluzione data dalle matrici

$$D_1 = \text{diag}(d_1, \dots, d_m) \quad D_2 = \text{diag}(d_{m+1}, \dots, d_{m+n})$$

2. Il problema 2 per la tripla (A', L, U) ammette una soluzione data dalla matrice

$$D_1 = \text{diag}(1, d_1, \dots, d_m, d_{m+1}^{-1}, \dots, d_{m+n}^{-1})$$

Dimostrazione. Osserviamo che valgono le seguenti uguaglianze

$$\begin{aligned} T_{[L,U]}(DA'D^{-1}) &= T_{[L,U]} \left(D \left[\begin{array}{c|c|c} 0 & (v^{(m)})^T & (0^{(n)})^T \\ \hline 0^{(m)} & 0^{(m \times m)} & A \\ \hline 0^{(n)} & 0^{(n \times m)} & 0^{(n \times n)} \end{array} \right] D^{-1} \right) \\ &= T_{[L,U]} \left(\left[\begin{array}{c|c|c} 0 & (v^{(m)})^T D_1^{-1} & (0^{(n)})^T \\ \hline 0^{(m)} & 0^{(m \times m)} & D_1 A D_2 \\ \hline D_2^{-1} v^{(n)} & 0^{(n \times m)} & 0^{(n \times n)} \end{array} \right] \right) \\ &= \left[\begin{array}{c|c|c} 0 & (v^{(m)})^T & (0^{(n)})^T \\ \hline 0^{(m)} & 0^{(m \times m)} & X \\ \hline v^{(n)} & 0^{(n \times m)} & 0^{(n \times n)} \end{array} \right] = X' \end{aligned}$$

Osserviamo come nell'ultimo passaggio è stata usata la definizione delle matrici L ed U . La matrice X' è soluzione del problema 2 se e solo se

$$X' \mathbf{1}_{m+n+1} = \begin{bmatrix} (v^{(m)})^T \mathbf{1}_m \\ X \mathbf{1}_n \\ v^{(n)} \end{bmatrix} = (X')^T \mathbf{1}_{m+n+1} = \begin{bmatrix} (v^{(n)})^T \mathbf{1}_n \\ v^{(m)} \\ X^T \mathbf{1}_m \end{bmatrix}$$

ovvero se e solo se

$$\begin{cases} (v^{(m)})^T \mathbf{1}_m = (v^{(n)})^T \mathbf{1}_n \\ X \mathbf{1}_n = v^{(m)} \\ v^{(n)} = X^T \mathbf{1}_m \end{cases}$$

che sono esattamente le condizioni che deve soddisfare X per essere soluzione del problema 4. \square

3.2 Relazione del duale con gli altri problemi

Grezie al risultato del lemma 3.1 definiamo la quantità $w = \sum_{i=1}^m v_i = \sum_{i=m+1}^{m+n} v_i$.

Come abbiamo visto nella precedente sezione un'istanza del problema 4 può essere trasformata in un'istanza del problema 2, quello che faremo adesso è andare ad esaminare la struttura del problema duale ottenuto dalla tripla (A', L, U) sfruttando la loro forma particolare.

Osserviamo che l'insieme dei nodi è partizionato dai due insiemi E_0 ed E_1 , e vale che

- se $e \in E_1$ si ha $l_e = 0$ e $u_e = +\infty$, per cui su questi nodi la funzione obiettivo del duale è certamente data dall'espressione con l'esponenziale,

- se $e \in E_0$ i limiti superiore e inferiore coincidono $l_e = a_e = u_e = v_i$ per cui la funzione duale è data dalla prima (o equivalentemente dalla terza) equazione di (2.3) calcolata in v_i , è facile verificare che si ottiene

$$f_e^*(y) = v_i(y + 1)$$

Queste osservazioni giustificano il seguente calcolo della funzione obiettivo del duale

$$\begin{aligned} \Phi(p) &= f^*(M^T p) = \sum_{\substack{e \in E \\ e=(i,j)}} f_e^*(p_i - p_j) \\ &= \sum_{\substack{e \in E_1 \\ e=(i,j)}} a_e e^{p_i - p_j} + \sum_{\substack{e \in E_0 \\ e=(0,i)}} v_i(p_0 - p_i + 1) + \sum_{\substack{e \in E_0 \\ e=(0,m+i)}} v_i(p_i - p_0 + 1) \\ &= \sum_{\substack{e \in E_1 \\ e=(i,j)}} a_e e^{p_i - p_j} + \sum_{i=1}^m v_i(p_0 - p_i + 1) + \sum_{i=m+1}^{m+n} v_i(p_i - p_0 + 1) \\ &= \sum_{\substack{e \in E_1 \\ e=(i,j)}} a_e e^{p_i - p_j} - \sum_{i=1}^m v_i p_i + \sum_{i=m+1}^{m+n} v_i p_i + 2w \end{aligned}$$

Osservando che la funzione obiettivo ha perso la dipendenza dalla variabile p_0 , scriviamo il problema duale associato a (A', L, U)

$$\min_{p \in \mathbb{R}^V} \sum_{\substack{e \in E_1 \\ e=(i,j)}} a_e e^{p_i - p_j} - \sum_{i=1}^m v_i p_i + \sum_{i=m+1}^{m+n} v_i p_i + 2w \quad (3.1)$$

Dai teoremi 3.1 e 2.1 possiamo ottenere banalmente il seguente corollario che lega la soluzione di quest'ultimo problema di ottimizzazione con la soluzione del problema 4.

Corollario 3.1. *Sia (A, v) una coppia consistente. I seguenti fatti sono equivalenti:*

1. *Il problema 4 ammette una soluzione data dalle matrici*

$$D_1 = \text{diag}(d_1, \dots, d_m) \quad D_2 = \text{diag}(d_{m+1}, \dots, d_{m+n})$$

2. *In (3.1) si ottiene il minimo con (p_1, \dots, p_{m+n}) dove*

$$p_i = \begin{cases} \ln d_i & \text{se } 1 \leq i \leq m \\ -\ln d_i & \text{se } m+1 \leq i \leq m+n \end{cases}$$

3.3 Un risultato teorico di esistenza

Con le stesse notazioni che abbiamo usato in precedenza, consideriamo il seguente problema di ottimizzazione lineare con vincoli non convessi:

$$\begin{aligned}
 \min \quad & \sum_{\substack{e \in E_1 \\ e=(i,j)}} d_i a_e d_j \\
 \text{s.t.} \quad & \prod_{i=1}^m d_i^{v_i} = 1 \\
 & \prod_{i=m+1}^{m+n} d_i^{v_i} = 1 \\
 & 0 < d \in \mathbb{R}^{m+n}
 \end{aligned} \tag{3.2}$$

Diversi autori in letteratura ([4]) hanno usato quest'ultimo come approccio al problema 4, quello che faremo in questa sezione è dimostrare che esso è equivalente a (3.1) e ricavare da questo fatto delle condizioni necessarie all'esistenza di una soluzione al problema 4.

Trasformiamo il problema precedente attuando il cambio di variabili $p_i = \ln d_i$ e applicando il logaritmo ai vincoli, otteniamo il seguente problema di ottimizzazione convessa con vincoli lineari

$$\begin{aligned}
 \min \quad & \sum_{\substack{e \in E_1 \\ e=(i,j)}} a_e e^{p_i+p_j} \\
 \text{s.t.} \quad & \sum_{i=1}^m v_i p_i = 0 \\
 & \sum_{i=m+1}^{m+n} v_i p_i = 0 \\
 & p \in \mathbb{R}^{m+n}
 \end{aligned} \tag{3.3}$$

Teorema 3.2. *Valgono i seguenti fatti:*

- Sia $p \in \mathbb{R}^{m+n}$ soluzione ottima per (3.1), e siano

$$\alpha := \frac{1}{w} \sum_{i=1}^m v_i p_i \quad e \quad \beta := \frac{1}{w} \sum_{i=m+1}^{m+n} v_i p_i$$

allora $q \in \mathbb{R}^{m+n}$ è soluzione ottima per (3.3) dove

$$q_i = \begin{cases} p_i - \alpha & 1 \leq i \leq m \\ \beta - p_i & m+1 \leq i \leq m+n \end{cases}$$

- Sia $q \in \mathbb{R}^{m+n}$ soluzione ottima per (3.3), e sia

$$\gamma := \ln \left(\frac{1}{w} \sum_{e \in E} a_e e^{p_i+p_j} \right)$$

allora $p \in \mathbb{R}^{m+n}$ è soluzione ottima per (3.1) dove

$$p_i = \begin{cases} q_i - \gamma & 1 \leq i \leq m \\ q_i & m+1 \leq i \leq m+n \end{cases}$$

Osservazione 3.2. I problemi di cui tratta l'ultimo teorema hanno funzione obiettivo convessa e vincoli lineari, pertanto le condizioni KKT sono necessarie e sufficienti per verificare se una soluzione è ottima. Pertanto possiamo dire che:

- Il vettore p risolve (3.1) se e solo se

$$\begin{aligned} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_i - p_j} - v_i &= 0 \quad 1 \leq i \leq m \\ v_i - \sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{p_i - p_j} &= 0 \quad m+1 \leq i \leq m+n \end{aligned}$$

- Il vettore q risolve (3.3) se e solo se esistono $\lambda, \mu \in \mathbb{R}$ tali che

$$\begin{aligned} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{q_i + q_j} - \lambda v_i &= 0 \quad 1 \leq i \leq m \\ \sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{q_i + q_j} - \mu v_i &= 0 \quad m+1 \leq i \leq m+n \\ \sum_{i=1}^m v_i q_i &= 0 \\ \sum_{i=m+1}^{m+n} v_i q_i &= 0 \end{aligned}$$

Dimostrazione. 1. Sia p un vettore che soddisfa le condizioni KKT per (3.1) scritte nell'osservazione precedente allora vale che

$$\begin{aligned} \sum_{i=1}^m v_i q_i &= \sum_{i=1}^m v_i (p_i - \alpha) = \sum_{i=1}^m v_i p_i - w\alpha = \sum_{i=1}^m v_i p_i - \sum_{i=1}^m v_i p_i = 0 \\ \sum_{i=m+1}^{m+n} v_i q_i &= \sum_{i=m+1}^{m+n} v_i (p_i - \beta) = \sum_{i=m+1}^{m+n} v_i p_i - w\beta = \sum_{i=m+1}^{m+n} v_i p_i - \sum_{i=m+1}^{m+n} v_i p_i = 0 \end{aligned}$$

per $1 \leq i \leq m$ sia $\lambda = a^{\beta-\alpha}$:

$$\begin{aligned} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{q_i + q_j} - \lambda v_i &= \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{q_i + q_j} - e^{\beta-\alpha} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_i - p_j} \\ &= \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_i - p_j - \alpha + \beta} - e^{\beta-\alpha} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_i - p_j} = 0 \end{aligned}$$

per $m + 1 \leq i \leq m + n$ sia $\mu = a^{\beta - \alpha}$:

$$\begin{aligned} \sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{q_i + q_j} - \mu v_i &= \sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{q_i + q_j} - e^{\beta - \alpha} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_i - p_j} \\ &= \sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{p_j - p_i - \alpha + \beta} - e^{\beta - \alpha} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_j - p_i} = 0 \end{aligned}$$

2. Sia q un vettore che soddisfa le condizioni KKT per (3.3) scritte nell'osservazione precedente. Osserviamo che per come è definito il problema (3.1) ogni nodo $e \in E$ è del tipo $e = (i, i + m)$ con $1 \leq i \leq n$. Per questo motivo sommando fra loro le prime m equazioni delle condizioni KKT di (3.3) si ottiene

$$\sum_{\substack{e \in E \\ e=(i,j)}} a_e e^{q_i + q_j} = \lambda w$$

Allo stesso modo sommando le successive n equazioni otteniamo

$$\sum_{\substack{e \in E \\ e=(i,j)}} a_e e^{q_i + q_j} = \mu w$$

da cui segue che

$$e^\gamma = \frac{1}{w} \sum_{\substack{e \in E \\ e=(i,j)}} a_e e^{q_i + q_j} = \lambda = \mu$$

Verifichiamo le condizioni KKT per (3.1): per $1 \leq i \leq m$ si ha

$$\sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{p_i - p_j} = \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{q_i + q_j - \gamma} = e^{-\gamma} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{q_i + q_j} = e^{-\gamma} \lambda v_i = v_i$$

per $m + 1 \leq i \leq m + n$ si ha

$$\sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{p_j - p_i} = \sum_{\substack{e \in \delta^-(i) \\ e=(i,j)}} a_e e^{q_j + q_i - \gamma} = e^{-\gamma} \sum_{\substack{e \in \delta^+(i) \\ e=(i,j)}} a_e e^{q_i + q_j} = e^{-\gamma} \mu v_i = v_i$$

□

Dal teorema 3.2 e dal corollario 3.1 otteniamo immediatamente il seguente risultato

Corollario 3.2. *Sia (A, v) una coppia consistente. I seguenti fatti sono equivalenti:*

1. *Il problema 4 ammette una soluzione data dalle matrici*

$$D_1 = \text{diag}(d_1, \dots, d_m) \quad D_2 = \text{diag}(d_{m+1}, \dots, d_{m+n})$$

2. *In (3.3) si ottiene il minimo con (p_1, \dots, p_{m+n}) dove*

$$p_i = \begin{cases} \ln d_i - \alpha & \text{se } 1 \leq i \leq m \\ \ln d_i - \beta & \text{se } m + 1 \leq i \leq m + n \end{cases}$$

con α e β definiti nel teorema precedente.

Ovviamente il corollario precedente può essere legato anche al problema (3.2), possiamo sfruttare questo collegamento per dimostrare il seguente teorema classico.

Teorema 3.3 (Sinkhorn). *Sia $A \in \mathbb{R}^{m \times n}$ una matrice con elementi strettamente positivi. Per ogni $v \in \mathbb{R}^{m+n}$ l'istanza del problema 4 data dalla coppia (A, v) ammette soluzione.*

Dimostrazione. Come sappiamo dal corollario 3.2, il problema 4 ha soluzione se e solo se (3.2) ammette minimo. La funzione obiettivo di (3.2) può essere scritta in modo equivalente tramite la forma bilineare $x^T A y$ con $x \in \mathbb{R}^m$ e $y \in \mathbb{R}^n$ che soddisfano i vincoli assegnati. Definendo $a := \min_{i,j} A_{ij}$, vale che

$$x^T A y \geq \|x\|_1 \|y\|_1 a \geq \|x\|_\infty \|y\|_\infty a \geq a$$

Da questo segue che $x^T A y$ è una funzione limitata dal basso da una costante positiva ed è coerciva per $\|x\|_\infty + \|y\|_\infty \rightarrow \infty$, per cui ammette minimo. \square

Riportiamo infine un ultimo risultato di esistenza, omettiamo la dimostrazione in quanto molto lunga e tecnica (la si può trovare in [4]).

Definizione 3.2. Una matrice $A \in \mathbb{R}^{n \times n}$ con elementi non negativi si dice *pienamente indecomponibile* se non esiste alcuna coppia di matrici di permutazione P e Q tali che

$$PAQ = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix}$$

con A e C matrici quadrate. Una matrice $A \in \mathbb{R}^{m \times n}$ è *pienamente indecomponibile* se ogni sua sottomatrice di taglia $m \times m$ lo è.

Teorema 3.4. *Sia $A \in \mathbb{R}^{m \times n}$ pienamente indecomponibile e siano $r, c \in \mathbb{R}$ tali che $mr = nc$. Allora il problema 4 per la coppia (A, v) ammette soluzione, dove*

$$v = \begin{bmatrix} c \mathbf{1}_m \\ r \mathbf{1}_n \end{bmatrix}$$

Capitolo 4

Algoritmo di scalatura

In questa sezione ci occuperemo di creare un algoritmo di scalatura sfruttando la formulazione dei problemi duali effettuata in precedenza. L'algoritmo preso in considerazione è un metodo iterativo di discesa applicato al problema duale (2.2) che informalmente consiste iterazione nello scegliere ad ogni iterazione una coordinata k e minimizzare la funzione obiettivo lungo la retta $p + tu_k$ dove u_k è il k -esimo vettore della base canonica. Dal punto di vista del problema primale, questo concetto si traduce nel considerare il k -esimo vincolo e ridurre la quantità per cui questo è violato.

4.1 Creazione dell'algoritmo per la scalatura non troncata.

L'idea del metodo di discesa sul problema duale porta alla scrittura di un algoritmo particolarmente semplice nel per il problema 3. In questo caso infatti, il k -esimo vincolo è dato da

$$0 = \sum_{j=1}^n d_k A_{kj} d_j^{-1} - \sum_{j=1}^n d_j A_{jk} d_k^{-1} = d_k \sum_{j=1}^n A_{kj} d_j^{-1} - d_k^{-1} \sum_{j=1}^n A_{jk} d_j$$

Possiamo forzare il vincolo ad essere soddisfatto aggiornando il valore di d_k con

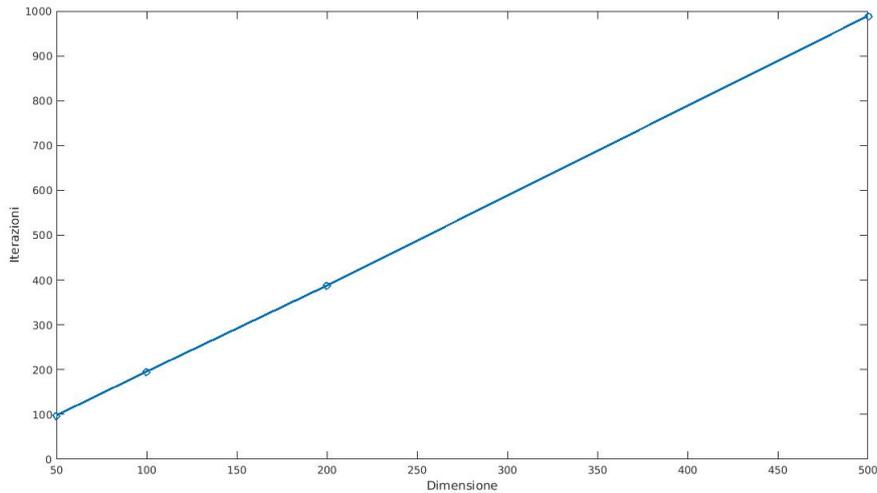
$$d_k^{(agg)} = \sqrt{\frac{\sum_{j=1}^n A_{jk} d_j}{\sum_{j=1}^n A_{kj} d_j^{-1}}}$$

che ci permette di descrivere l'algoritmo tramite lo schema:

- Inizializziamo i dati del problema ponendo $d_i = 1$ per ogni i , questo coincide a porre inizialmente $D = I_n$ e $X = A$.
- Chiamiamo $r(X) = X\mathbf{1}_n - X^T\mathbf{1}_n$ il vettore dei residui che indica quanto la matrice $X = DAD^{-1}$ sia vicina all'essere bilanciata e ϵ una certa soglia di tolleranza fissata. Se $\|r(X)\|_\infty > \epsilon$ passiamo all'operazione successiva altrimenti terminiamo.
- Selezioniamo l'indice k in cui si ottiene la norma infinito di $r(X)$.
- Sostituiamo la componente k -esima d_k della diagonale di D con il valore $d_k^{(agg)}$ e torniamo al secondo punto.

4.2 Sperimentazione numerica

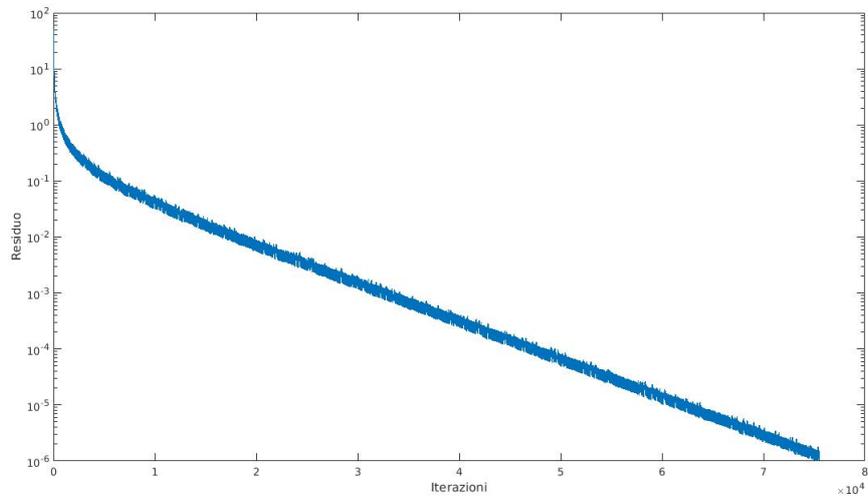
Il primo esperimento effettuato ha lo scopo di investigare la crescita del numero di iterazioni che l'algoritmo effettua al crescere della dimensione delle matrici in gioco. Abbiamo considerato in input matrici composte da elementi scelti in maniera casuale con distribuzione uniforme nell'intervallo $(0, 1)$. Per ogni dimensione $n = 50, 100, 200, 500$ abbiamo fatto girare l'algoritmo su un campione composto da 100 matrici di questo tipo e poi abbiamo fatto la media del numero di iterazioni che sono state necessarie ad ottenere una tolleranza pari a $\epsilon = 10^{-2}$. Abbiamo ottenuto il seguente grafico



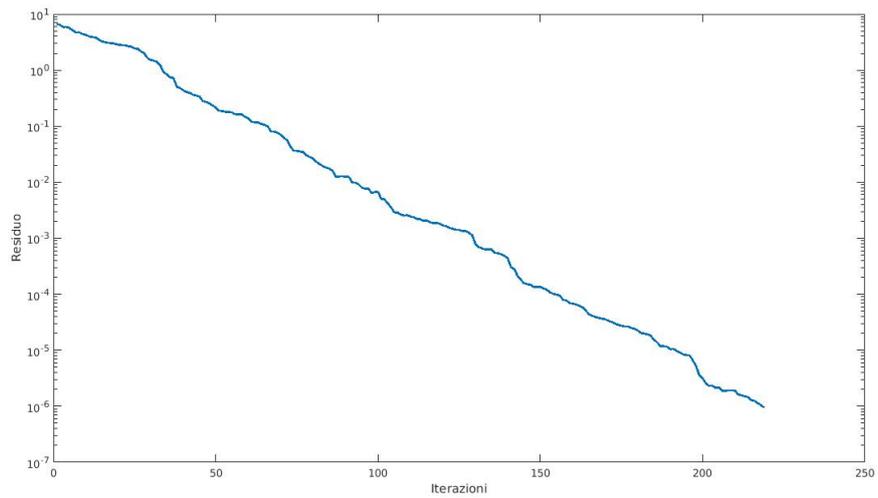
Come si può apprezzare dal grafico la crescita è pressochè lineare. Tuttavia questo risultato dipende fortemente dalla struttura della matrice in input, come possiamo vedere nel secondo esperimento: fissiamo ad esempio $n = 50$ e consideriamo la seguente matrice di Hessenberg

$$H_n = \begin{bmatrix} 1 & \dots & \dots & \dots & 1 \\ 1 & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 1 \end{bmatrix}$$

facendo girare l'algoritmo su H_n otteniamo il seguente grafico (in scala semilogaritmica) che indica la decrescita della norma infinito del vettore residuo



In questo caso la decrescita non è molto veloce e l'algoritmo impiega un numero di iterazioni dell'ordine delle migliaia per ottenere una precisione di 10^{-6} . Un grafico analogo per una matrice della stessa dimensione ma del tipo del primo esperimento, risulta



Possiamo concludere che l'algoritmo risulta particolarmente efficace per matrici non sparse ovvero il cui grafo associato sia molto connesso.

Bibliografia

- [1] Tzu-Yi Chen and James W. Demmel. Balancing sparse matrices for computing eigenvalues. In *Proceedings of the International Workshop on Accurate Solution of Eigenvalue Problems (University Park, PA, 1998)*, volume 309, pages 261–287, 2000.
- [2] Philip A. Knight. The Sinkhorn-Knopp algorithm: convergence and applications. *SIAM J. Matrix Anal. Appl.*, 30(1):261–275, 2008.
- [3] B. Lamond and N. F. Stewart. Bregman’s balancing method. *Transportation Res. Part B*, 15(4):239–248, 1981.
- [4] Albert W. Marshall and Ingram Olkin. Scaling of matrices to achieve specified row and column sums. *Numer. Math.*, 12:83–90, 1968.
- [5] R. T. Rockafellar. *Network flows and monotropic optimization*. Pure and Applied Mathematics (New York). John Wiley & Sons, Inc., New York, 1984. A Wiley-Interscience Publication.
- [6] Michael H. Schneider. Matrix scaling, entropy minimization, and conjugate duality. II. The dual problem. *Math. Programming*, 48(1, (Ser. B)):103–124, 1990.
- [7] Michael H Schneider and Stavros A Zenios. A comparative study of algorithms for matrix balancing. *Operations research*, 38(3):439–455, 1990.