

Clustering spettrale via processo di mescolamento

Luca Ferragina

Università di Pisa

5 settembre 2016

Piano della presentazione

- 1 Il problema del Clustering
- 2 Processo di mescolamento
- 3 Analisi di convergenza
- 4 Costruzione e costo computazionale dell'algoritmo

Il Clustering

Consideriamo un insieme composto da n oggetti $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$.

Lo scopo del *Clustering* è partizionare \mathcal{V} in modo che oggetti appartenenti allo stesso sottinsieme, *cluster*, della partizione siano simili fra loro, mentre oggetti appartenenti a cluster differenti siano dissimili.

Per formalizzare il problema possiamo immaginare che $\mathcal{V} \subset \mathbb{R}^d$ per un certo d , e che su $\mathcal{V} \times \mathcal{V}$ sia definita una funzione di similarità $s(\mathbf{v}_i, \mathbf{v}_j)$, che abbrevieremo in $s(i, j)$, tale che:

- $s(i, j) = s(j, i)$
- $s(i, j) \geq 0$
- $s(i, i) = 0$

Il Clustering

Consideriamo un insieme composto da n oggetti $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$. Lo scopo del *Clustering* è partizionare \mathcal{V} in modo che oggetti appartenenti allo stesso sottinsieme, *cluster*, della partizione siano simili fra loro, mentre oggetti appartenenti a cluster differenti siano dissimili.

Per formalizzare il problema possiamo immaginare che $\mathcal{V} \subset \mathbb{R}^d$ per un certo d , e che su $\mathcal{V} \times \mathcal{V}$ sia definita una funzione di similarità $s(\mathbf{v}_i, \mathbf{v}_j)$, che abbrevieremo in $s(i, j)$, tale che:

- $s(i, j) = s(j, i)$
- $s(i, j) \geq 0$
- $s(i, i) = 0$

Il Clustering

Consideriamo un insieme composto da n oggetti $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$. Lo scopo del *Clustering* è partizionare \mathcal{V} in modo che oggetti appartenenti allo stesso sottinsieme, *cluster*, della partizione siano simili fra loro, mentre oggetti appartenenti a cluster differenti siano dissimili.

Per formalizzare il problema possiamo immaginare che $\mathcal{V} \subset \mathbb{R}^d$ per un certo d , e che su $\mathcal{V} \times \mathcal{V}$ sia definita una funzione di similarità $s(\mathbf{v}_i, \mathbf{v}_j)$, che abbrevieremo in $s(i, j)$, tale che:

- $s(i, j) = s(j, i)$
- $s(i, j) \geq 0$
- $s(i, i) = 0$

Il Clustering

Consideriamo un insieme composto da n oggetti $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$. Lo scopo del *Clustering* è partizionare \mathcal{V} in modo che oggetti appartenenti allo stesso sottinsieme, *cluster*, della partizione siano simili fra loro, mentre oggetti appartenenti a cluster differenti siano dissimili.

Per formalizzare il problema possiamo immaginare che $\mathcal{V} \subset \mathbb{R}^d$ per un certo d , e che su $\mathcal{V} \times \mathcal{V}$ sia definita una funzione di similarità $s(\mathbf{v}_i, \mathbf{v}_j)$, che abbrevieremo in $s(i, j)$, tale che:

- $s(i, j) = s(j, i)$
- $s(i, j) \geq 0$
- $s(i, i) = 0$

Matrice e grafo di similarità

Definizione

La *matrice di similarità* W associata ai punti \mathbf{v}_i e alla funzione s è tale che $w_{ij} = s(i, j)$.

Definizione

Il *grafo di similarità* \mathcal{G} associato ai punti \mathbf{v}_i e alla funzione s è un grafo pesato e non orientato che ha come nodi l'insieme \mathcal{V} . Vi è un arco che congiunge i , il nodo relativo a \mathbf{v}_i , e j , il nodo relativo a \mathbf{v}_j se $s(i, j) > 0$ e in tal caso il peso dell'arco è dato proprio da $s(i, j)$.

Matrice e grafo di similarità

Definizione

La *matrice di similarità* W associata ai punti \mathbf{v}_i e alla funzione s è tale che $w_{ij} = s(i, j)$.

Definizione

Il *grafo di similarità* \mathcal{G} associato ai punti \mathbf{v}_i e alla funzione s è un grafo pesato e non orientato che ha come nodi l'insieme \mathcal{V} . Vi è un arco che congiunge i , il nodo relativo a \mathbf{v}_i , e j , il nodo relativo a \mathbf{v}_j se $s(i, j) > 0$ e in tal caso il peso dell'arco è dato proprio da $s(i, j)$.

Matrice e grafo di similarità

Definizione

La *matrice di similarità* W associata ai punti \mathbf{v}_i e alla funzione s è tale che $w_{ij} = s(i, j)$.

Definizione

Il *grafo di similarità* \mathcal{G} associato ai punti \mathbf{v}_i e alla funzione s è un grafo pesato e non orientato che ha come nodi l'insieme \mathcal{V} . Vi è un arco che congiunge i , il nodo relativo a \mathbf{v}_i , e j , il nodo relativo a \mathbf{v}_j se $s(i, j) > 0$ e in tal caso il peso dell'arco è dato proprio da $s(i, j)$.

Esempi (Funzioni di similarità)

- Similarità Gaussiana

$$s(i, j) = \begin{cases} \exp\left(-\frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2}{2\sigma^2}\right) & \text{se } \|\mathbf{v}_i - \mathbf{v}_j\| < \delta, \\ 0 & \text{altrimenti.} \end{cases}$$

- Similarità p -nearest

$$s(i, j) = \begin{cases} 1 & \text{se } i \text{ è uno dei } p \text{ punti più vicini a } j \text{ o viceversa,} \\ 0 & \text{altrimenti.} \end{cases}$$

- Similarità ϵ -neighborhood

$$s(i, j) = \begin{cases} 1 & \text{se } \|\mathbf{v}_i - \mathbf{v}_j\| < \epsilon, \\ 0 & \text{altrimenti.} \end{cases}$$

Matrice e grafo di similarità

Osservazione

Trovare cluster in un insieme è equivalente a partizionare il grafo di similarità in modo che la somma dei pesi degli archi che uniscono due diversi sottografi sia piccola e che ogni sottografo sia denso.

Sia $d_i = \sum_{j=1}^n w_{ij}$ il grado del nodo i e sia D la matrice grado, ovvero la matrice diagonale che ha d_i come i -esimo elemento diagonale.

Assumeremo che ogni nodo abbia grado positivo, questo fatto ci permette di definire la matrice di similarità normalizzata

$$\overline{W} = D^{-1}W$$

Matrice e grafo di similarità

Osservazione

Trovare cluster in un insieme è equivalente a partizionare il grafo di similarità in modo che la somma dei pesi degli archi che uniscono due diversi sottografi sia piccola e che ogni sottografo sia denso.

Sia $d_i = \sum_{j=1}^n w_{ij}$ il grado del nodo i e sia D la matrice grado, ovvero la matrice diagonale che ha d_i come i -esimo elemento diagonale.

Assumeremo che ogni nodo abbia grado positivo, questo fatto ci permette di definire la matrice di similarità normalizzata $\overline{W} = D^{-1}W$

Matrice e grafo di similarità

Osservazione

Trovare cluster in un insieme è equivalente a partizionare il grafo di similarità in modo che la somma dei pesi degli archi che uniscono due diversi sottografi sia piccola e che ogni sottografo sia denso.

Sia $d_i = \sum_{j=1}^n w_{ij}$ il grado del nodo i e sia D la matrice grado, ovvero la matrice diagonale che ha d_i come i -esimo elemento diagonale.

Assumeremo che ogni nodo abbia grado positivo, questo fatto ci permette di definire la matrice di similarità normalizzata

$$\overline{W} = D^{-1}W$$

Processo di mescolamento

- Per visualizzare il nostro approccio, consideriamo un processo dinamico a tempo discreto in cui i punti \mathbf{v}_i si muovono, mescolandosi fra loro.
- Ad ogni passo \mathbf{v}_i si avvicina a \mathbf{v}_j di una distanza proporzionale a $s(i, j)$.
- Ci aspettiamo che un punto \mathbf{v}_i si avvicinerà ai punti con cui ha una forte similarità andando a formare un cluster significativamente lontano dai cluster dei punti a lui dissimili.

Processo di mescolamento

- Per visualizzare il nostro approccio, consideriamo un processo dinamico a tempo discreto in cui i punti \mathbf{v}_i si muovono, mescolandosi fra loro.
- Ad ogni passo \mathbf{v}_i si avvicina a \mathbf{v}_j di una distanza proporzionale a $s(i, j)$.
- Ci aspettiamo che un punto \mathbf{v}_i si avvicinerà ai punti con cui ha una forte similarità andando a formare un cluster significativamente lontano dai cluster dei punti a lui dissimili.

Processo di mescolamento

- Per visualizzare il nostro approccio, consideriamo un processo dinamico a tempo discreto in cui i punti \mathbf{v}_i si muovono, mescolandosi fra loro.
- Ad ogni passo \mathbf{v}_i si avvicina a \mathbf{v}_j di una distanza proporzionale a $s(i, j)$.
- Ci aspettiamo che un punto \mathbf{v}_i si avvicinerà ai punti con cui ha una forte similarità andando a formare un cluster significativamente lontano dai cluster dei punti a lui dissimili.

L'idea precedente può essere descritta tramite un modello in cui i punti si trovano nelle loro posizioni originali al tempo $k = 0$ e si muovono secondo la seguente equazione:

$$\begin{aligned} \mathbf{v}_i^{(k+1)} &= \mathbf{v}_i^{(k)} + \alpha \sum_{j=1}^n \bar{w}_{ij} (\mathbf{v}_j^{(k)} - \mathbf{v}_i^{(k)}) \\ &= (1 - \alpha) \mathbf{v}_i^{(k)} + \alpha \sum_{j=1}^n \bar{w}_{ij} \mathbf{v}_j^{(k)} \end{aligned} \quad (1)$$

Il parametro $\alpha \in [0, 1]$ controlla la velocità ad ogni passo.

Osservazione

Se il grafo è bipartito e $\alpha = 1$ ad ogni passo tutti i punti di una componente si muoveranno verso quelli dell'altra componente. Per cui i punti delle due componenti non si mescoleranno neanche dopo un elevato numero di iterazioni.

L'idea precedente può essere descritta tramite un modello in cui i punti si trovano nelle loro posizioni originali al tempo $k = 0$ e si muovono secondo la seguente equazione:

$$\begin{aligned} \mathbf{v}_i^{(k+1)} &= \mathbf{v}_i^{(k)} + \alpha \sum_{j=1}^n \bar{w}_{ij} (\mathbf{v}_j^{(k)} - \mathbf{v}_i^{(k)}) \\ &= (1 - \alpha) \mathbf{v}_i^{(k)} + \alpha \sum_{j=1}^n \bar{w}_{ij} \mathbf{v}_j^{(k)} \end{aligned} \tag{1}$$

Il parametro $\alpha \in [0, 1]$ controlla la velocità ad ogni passo.

Osservazione

Se il grafo è bipartito e $\alpha = 1$ ad ogni passo tutti i punti di una componente si muoveranno verso quelli dell'altra componente. Per cui i punti delle due componenti non si mescoleranno neanche dopo un elevato numero di iterazioni.

L'idea precedente può essere descritta tramite un modello in cui i punti si trovano nelle loro posizioni originali al tempo $k = 0$ e si muovono secondo la seguente equazione:

$$\begin{aligned} \mathbf{v}_i^{(k+1)} &= \mathbf{v}_i^{(k)} + \alpha \sum_{j=1}^n \bar{w}_{ij} (\mathbf{v}_j^{(k)} - \mathbf{v}_i^{(k)}) \\ &= (1 - \alpha) \mathbf{v}_i^{(k)} + \alpha \sum_{j=1}^n \bar{w}_{ij} \mathbf{v}_j^{(k)} \end{aligned} \quad (1)$$

Il parametro $\alpha \in [0, 1]$ controlla la velocità ad ogni passo.

Osservazione

Se il grafo è bipartito e $\alpha = 1$ ad ogni passo tutti i punti di una componente si muoveranno verso quelli dell'altra componente. Per cui i punti delle due componenti non si mescoleranno neanche dopo un elevato numero di iterazioni.

L'equazione precedente può essere espressa in forma matriciale come:

$$V^{(k+1)} = MV^{(k)} \quad (2)$$

In cui si è posto:

- 1 $V^{(k)}$ la matrice di taglia $n \times d$ la cui i -esima riga rappresenta la posizione del punto \mathbf{v}_i al tempo k .
- 2 $M = (1 - \alpha)I + \alpha\overline{W} \in \mathbb{R}^{n \times n}$

Osservazioni

- Possiamo vedere M come la matrice di transizione di una catena di Markov.
- Poichè M cattura la similarità dei punti ci si può aspettare che per k sufficientemente grande il processo descritto dall'ultima equazione rivelerà i cluster.

L'equazione precedente può essere espressa in forma matriciale come:

$$V^{(k+1)} = MV^{(k)} \quad (2)$$

In cui si è posto:

- 1 $V^{(k)}$ la matrice di taglia $n \times d$ la cui i -esima riga rappresenta la posizione del punto \mathbf{v}_i al tempo k .
- 2 $M = (1 - \alpha)I + \alpha\overline{W} \in \mathbb{R}^{n \times n}$

Osservazioni

- Possiamo vedere M come la matrice di transizione di una catena di Markov.
- Poichè M cattura la similarità dei punti ci si può aspettare che per k sufficientemente grande il processo descritto dall'ultima equazione rivelerà i cluster.

L'equazione precedente può essere espressa in forma matriciale come:

$$V^{(k+1)} = MV^{(k)} \quad (2)$$

In cui si è posto:

- 1 $V^{(k)}$ la matrice di taglia $n \times d$ la cui i -esima riga rappresenta la posizione del punto \mathbf{v}_i al tempo k .
- 2 $M = (1 - \alpha)I + \alpha\overline{W} \in \mathbb{R}^{n \times n}$

Osservazioni

- Possiamo vedere M come la matrice di transizione di una catena di Markov.
- Poichè M cattura la similarità dei punti ci si può aspettare che per k sufficientemente grande il processo descritto dall'ultima equazione rivelerà i cluster.

L'equazione precedente può essere espressa in forma matriciale come:

$$V^{(k+1)} = MV^{(k)} \quad (2)$$

In cui si è posto:

- 1 $V^{(k)}$ la matrice di taglia $n \times d$ la cui i -esima riga rappresenta la posizione del punto \mathbf{v}_i al tempo k .
- 2 $M = (1 - \alpha)I + \alpha\overline{W} \in \mathbb{R}^{n \times n}$

Osservazioni

- Possiamo vedere M come la matrice di transizione di una catena di Markov.
- Poichè M cattura la similarità dei punti ci si può aspettare che per k sufficientemente grande il processo descritto dall'ultima equazione rivelerà i cluster.

L'equazione precedente può essere espressa in forma matriciale come:

$$V^{(k+1)} = MV^{(k)} \quad (2)$$

In cui si è posto:

- 1 $V^{(k)}$ la matrice di taglia $n \times d$ la cui i -esima riga rappresenta la posizione del punto \mathbf{v}_i al tempo k .
- 2 $M = (1 - \alpha)I + \alpha\overline{W} \in \mathbb{R}^{n \times n}$

Osservazioni

- Possiamo vedere M come la matrice di transizione di una catena di Markov.
- Poichè M cattura la similarità dei punti ci si può aspettare che per k sufficientemente grande il processo descritto dall'ultima equazione rivelerà i cluster.

Prima idea euristica di algoritmo

Algoritmo

- 1 Calcoliamo le matrici V e $M = (1 - \alpha)I + \alpha\overline{W}$
- 2 Finchè non è soddisfatto un certo criterio d'arresto, calcoliamo $V^{(k+1)} = MV^{(k)}$ e incrementiamo k .
- 3 Troviamo i cluster dalle righe della matrice ottenuta nell'ultima iterazione applicando l'algoritmo k-means.

In questi termini l'algoritmo presenta due evidenti punti deboli:

- il costo computazionale è di $O(n^2d)$ per iterazione, se n è grande diventa molto oneroso.
- fallisce quando deve identificare cluster dentro altri cluster

Prima idea euristica di algoritmo

Algoritmo

- 1 Calcoliamo le matrici V e $M = (1 - \alpha)I + \alpha\overline{W}$
- 2 Finchè non è soddisfatto un certo criterio d'arresto, calcoliamo $V^{(k+1)} = MV^{(k)}$ e incrementiamo k .
- 3 Troviamo i cluster dalle righe della matrice ottenuta nell'ultima iterazione applicando l'algoritmo k-means.

In questi termini l'algoritmo presenta due evidenti punti deboli:

- il costo computazionale è di $O(n^2d)$ per iterazione, se n è grande diventa molto oneroso.
- fallisce quando deve identificare cluster dentro altri cluster

Prima idea euristica di algoritmo

Algoritmo

- 1 Calcoliamo le matrici V e $M = (1 - \alpha)I + \alpha\overline{W}$
- 2 Finchè non è soddisfatto un certo criterio d'arresto, calcoliamo $V^{(k+1)} = MV^{(k)}$ e incrementiamo k .
- 3 Troviamo i cluster dalle righe della matrice ottenuta nell'ultima iterazione applicando l'algoritmo k-means.

In questi termini l'algoritmo presenta due evidenti punti deboli:

- il costo computazionale è di $O(n^2d)$ per iterazione, se n è grande diventa molto oneroso.
- fallisce quando deve identificare cluster dentro altri cluster

Prima idea euristica di algoritmo

Algoritmo

- 1 Calcoliamo le matrici V e $M = (1 - \alpha)I + \alpha\overline{W}$
- 2 Finchè non è soddisfatto un certo criterio d'arresto, calcoliamo $V^{(k+1)} = MV^{(k)}$ e incrementiamo k .
- 3 Troviamo i cluster dalle righe della matrice ottenuta nell'ultima iterazione applicando l'algoritmo k-means.

In questi termini l'algoritmo presenta due evidenti punti deboli:

- il costo computazionale è di $O(n^2d)$ per iterazione, se n è grande diventa molto oneroso.
- fallisce quando deve identificare cluster dentro altri cluster

Prima idea euristica di algoritmo

Algoritmo

- 1 Calcoliamo le matrici V e $M = (1 - \alpha)I + \alpha\overline{W}$
- 2 Finchè non è soddisfatto un certo criterio d'arresto, calcoliamo $V^{(k+1)} = MV^{(k)}$ e incrementiamo k .
- 3 Troviamo i cluster dalle righe della matrice ottenuta nell'ultima iterazione applicando l'algoritmo k-means.

In questi termini l'algoritmo presenta due evidenti punti deboli:

- il costo computazionale è di $O(n^2d)$ per iterazione, se n è grande diventa molto oneroso.
- fallisce quando deve identificare cluster dentro altri cluster

Prima idea euristica di algoritmo

Algoritmo

- 1 Calcoliamo le matrici V e $M = (1 - \alpha)I + \alpha\overline{W}$
- 2 Finchè non è soddisfatto un certo criterio d'arresto, calcoliamo $V^{(k+1)} = MV^{(k)}$ e incrementiamo k .
- 3 Troviamo i cluster dalle righe della matrice ottenuta nell'ultima iterazione applicando l'algoritmo k-means.

In questi termini l'algoritmo presenta due evidenti punti deboli:

- il costo computazionale è di $O(n^2d)$ per iterazione, se n è grande diventa molto oneroso.
- fallisce quando deve identificare cluster dentro altri cluster

Per superare queste limitazioni, associamo ad ogni punto \mathbf{v}_i uno scalare x_i che chiameremo *agente* di \mathbf{v}_i e procediamo con il calcolo nello spazio agente. Gli agenti iniziali da associare ai punti sono scelti casualmente con distribuzione uniforme nell'intervallo $[0, b]$. L'equazione può essere scritta come:

$$\mathbf{x}^{(k+1)} = ((1 - \alpha)I + \alpha \overline{W})\mathbf{x}^{(k)} = M\mathbf{x}^{(k)} \quad (3)$$

Per superare queste limitazioni, associamo ad ogni punto \mathbf{v}_i uno scalare x_i che chiameremo *agente* di \mathbf{v}_i e procediamo con il calcolo nello spazio agente. Gli agenti iniziali da associare ai punti sono scelti casualmente con distribuzione uniforme nell'intervallo $[0, b]$. L'equazione può essere scritta come:

$$\mathbf{x}^{(k+1)} = ((1 - \alpha)I + \alpha \overline{W})\mathbf{x}^{(k)} = M\mathbf{x}^{(k)} \quad (3)$$

Chiameremo *processo di mescolamento* il processo descritto dall'equazione precedente.

- La speranza è che, dopo un numero sufficiente di iterazioni, le entrate del vettore $x^{(k)}$ riveleranno cluster sulla retta reale che rappresenteranno i cluster sui dati iniziali.
- L'idea che supporta questa speranza è che i processi (2) e (3) si mescolano con la stessa velocità che è governata dalle proprietà della matrice M .

Chiameremo *processo di mescolamento* il processo descritto dall'equazione precedente.

- La speranza è che, dopo un numero sufficiente di iterazioni, le entrate del vettore $\mathbf{x}^{(k)}$ riveleranno cluster sulla retta reale che rappresenteranno i cluster sui dati iniziali.
- L'idea che supporta questa speranza è che i processi (2) e (3) si mescolano con la stessa velocità che è governata dalle proprietà della matrice M .

Chiameremo *processo di mescolamento* il processo descritto dall'equazione precedente.

- La speranza è che, dopo un numero sufficiente di iterazioni, le entrate del vettore $\mathbf{x}^{(k)}$ riveleranno cluster sulla retta reale che rappresenteranno i cluster sui dati iniziali.
- L'idea che supporta questa speranza è che i processi (2) e (3) si mescolano con la stessa velocità che è governata dalle proprietà della matrice M .

Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (1)

M è diagonalizzabile

Dimostrazione.

$$\begin{aligned} M &= (1 - \alpha)I + \alpha \overline{W} \\ &= I - \alpha(I - D^{-1}W) \\ &= I - \alpha D^{-1/2} L D^{1/2} \\ &= D^{-1/2} (I - \alpha L) D^{1/2}. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (1)

M è diagonalizzabile

Dimostrazione.

$$\begin{aligned} M &= (1 - \alpha)I + \alpha \overline{W} \\ &= I - \alpha(I - D^{-1}W) \\ &= I - \alpha D^{-1/2} L D^{1/2} \\ &= D^{-1/2} (I - \alpha L) D^{1/2}. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (1)

M è diagonalizzabile

Dimostrazione.

$$\begin{aligned} M &= (1 - \alpha)I + \alpha \overline{W} \\ &= I - \alpha(I - D^{-1}W) \\ &= I - \alpha D^{-1/2} L D^{1/2} \\ &= D^{-1/2} (I - \alpha L) D^{1/2}. \end{aligned}$$

Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (1)

M è diagonalizzabile

Dimostrazione.

$$\begin{aligned} M &= (1 - \alpha)I + \alpha \overline{W} \\ &= I - \alpha(I - D^{-1}W) \\ &= I - \alpha D^{-1/2} L D^{1/2} \\ &= D^{-1/2} (I - \alpha L) D^{1/2}. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (1)

M è diagonalizzabile

Dimostrazione.

$$\begin{aligned} M &= (1 - \alpha)I + \alpha \overline{W} \\ &= I - \alpha(I - D^{-1}W) \\ &= I - \alpha D^{-1/2} L D^{1/2} \\ &= D^{-1/2} (I - \alpha L) D^{1/2}. \end{aligned}$$

Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (1)

M è diagonalizzabile

Dimostrazione.

$$\begin{aligned} M &= (1 - \alpha)I + \alpha \overline{W} \\ &= I - \alpha(I - D^{-1}W) \\ &= I - \alpha D^{-1/2} L D^{1/2} \\ &= D^{-1/2} (I - \alpha L) D^{1/2}. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (2)

Sia ϕ_i l'autovettore destro di L relativo all'autovalore λ_i , allora $D^{-1/2} \phi_i$ è l'autovettore destro di M relativo all'autovalore $\mu_i = 1 - \alpha \lambda_i$

Dimostrazione.

$$\begin{aligned} M D^{-1/2} \phi_i &= (I - \alpha D^{-1/2} L D^{1/2}) D^{-1/2} \phi_i \\ &= (1 - \alpha \lambda_i) D^{-1/2} \phi_i. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (2)

Sia ϕ_i l'autovettore destro di L relativo all'autovalore λ_i , allora $D^{-1/2} \phi_i$ è l'autovettore destro di M relativo all'autovalore $\mu_i = 1 - \alpha \lambda_i$

Dimostrazione.

$$\begin{aligned} M D^{-1/2} \phi_i &= (I - \alpha D^{-1/2} L D^{1/2}) D^{-1/2} \phi_i \\ &= (1 - \alpha \lambda_i) D^{-1/2} \phi_i. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (2)

Sia ϕ_i l'autovettore destro di L relativo all'autovalore λ_i , allora $D^{-1/2}\phi_i$ è l'autovettore destro di M relativo all'autovalore $\mu_i = 1 - \alpha\lambda_i$

Dimostrazione.

$$\begin{aligned} MD^{-1/2}\phi_i &= (I - \alpha D^{-1/2} L D^{1/2}) D^{-1/2}\phi_i \\ &= (1 - \alpha\lambda_i) D^{-1/2}\phi_i. \end{aligned}$$



Consideriamo $L = I - D^{-1/2} W D^{-1/2}$ il Laplaciano normalizzato del grafo \mathcal{G} .

Lemma (2)

Sia ϕ_i l'autovettore destro di L relativo all'autovalore λ_i , allora $D^{-1/2}\phi_i$ è l'autovettore destro di M relativo all'autovalore $\mu_i = 1 - \alpha\lambda_i$

Dimostrazione.

$$\begin{aligned} MD^{-1/2}\phi_i &= (I - \alpha D^{-1/2} L D^{1/2}) D^{-1/2}\phi_i \\ &= (1 - \alpha\lambda_i) D^{-1/2}\phi_i. \end{aligned}$$



Osservazioni

L è una matrice simmetrica semidefinita positiva con autovalori compresi nell'intervallo $[0, 2]$. Per cui:

- Se $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2$ sono gli autovalori di L , allora i corrispondenti autovalori di M sono $1 \geq \mu_1 \geq \mu_2 \geq \dots \geq \mu_n \geq 1 - 2\alpha$.
- Gli autovettori normalizzati di L formano una base ortonormale di \mathbb{R}^n , quindi possiamo scrivere $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T$. Usando le espressioni derivanti dalle dimostrazioni precedenti otteniamo:

$$\begin{aligned} M^k &= (D^{-1/2}(I - \alpha L)D^{-1/2})^k \\ &= D^{-1/2} \left(\sum_{i=1}^n \mu_i^k \phi_i \phi_i^T \right) D^{-1/2}. \end{aligned} \quad (4)$$

Osservazioni

L è una matrice simmetrica semidefinita positiva con autovalori compresi nell'intervallo $[0, 2]$. Per cui:

- Se $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2$ sono gli autovalori di L , allora i corrispondenti autovalori di M sono
 $1 \geq \mu_1 \geq \mu_2 \geq \dots \geq \mu_n \geq 1 - 2\alpha$.
- Gli autovettori normalizzati di L formano una base ortonormale di \mathbb{R}^n , quindi possiamo scrivere $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T$. Usando le espressioni derivanti dalle dimostrazioni precedenti otteniamo:

$$\begin{aligned} M^k &= (D^{-1/2}(I - \alpha L)D^{-1/2})^k \\ &= D^{-1/2} \left(\sum_{i=1}^n \mu_i^k \phi_i \phi_i^T \right) D^{-1/2}. \end{aligned} \quad (4)$$

Osservazioni

L è una matrice simmetrica semidefinita positiva con autovalori compresi nell'intervallo $[0, 2]$. Per cui:

- Se $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2$ sono gli autovalori di L , allora i corrispondenti autovalori di M sono
 $1 \geq \mu_1 \geq \mu_2 \geq \dots \geq \mu_n \geq 1 - 2\alpha$.
- Gli autovettori normalizzati di L formano una base ortonormale di \mathbb{R}^n , quindi possiamo scrivere $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T$. Usando le espressioni derivanti dalle dimostrazioni precedenti otteniamo:

$$\begin{aligned} M^k &= (D^{-1/2}(I - \alpha L)D^{-1/2})^k \\ &= D^{-1/2} \left(\sum_{i=1}^n \mu_i^k \phi_i \phi_i^T \right) D^{-1/2}. \end{aligned} \quad (4)$$

Osservazioni

L è una matrice simmetrica semidefinita positiva con autovalori compresi nell'intervallo $[0, 2]$. Per cui:

- Se $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2$ sono gli autovalori di L , allora i corrispondenti autovalori di M sono
 $1 \geq \mu_1 \geq \mu_2 \geq \dots \geq \mu_n \geq 1 - 2\alpha$.
- Gli autovettori normalizzati di L formano una base ortonormale di \mathbb{R}^n , quindi possiamo scrivere $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T$. Usando le espressioni derivanti dalle dimostrazioni precedenti otteniamo:

$$\begin{aligned} M^k &= (D^{-1/2}(I - \alpha L)D^{-1/2})^k \\ &= D^{-1/2} \left(\sum_{i=1}^n \mu_i^k \phi_i \phi_i^T \right) D^{-1/2}. \end{aligned} \tag{4}$$

Osservazioni

L è una matrice simmetrica semidefinita positiva con autovalori compresi nell'intervallo $[0, 2]$. Per cui:

- Se $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2$ sono gli autovalori di L , allora i corrispondenti autovalori di M sono
 $1 \geq \mu_1 \geq \mu_2 \geq \dots \geq \mu_n \geq 1 - 2\alpha$.
- Gli autovettori normalizzati di L formano una base ortonormale di \mathbb{R}^n , quindi possiamo scrivere $L = \sum_{i=1}^n \lambda_i \phi_i \phi_i^T$. Usando le espressioni derivanti dalle dimostrazioni precedenti otteniamo:

$$\begin{aligned} M^k &= (D^{-1/2}(I - \alpha L)D^{-1/2})^k \\ &= D^{-1/2} \left(\sum_{i=1}^n \mu_i^k \phi_i \phi_i^T \right) D^{-1/2}. \end{aligned} \quad (4)$$

Analisi del caso ideale

Consideriamo il caso in cui i punti formano cluster ben separati.

- Supponiamo che ci siano m cluster $\mathcal{V}_1, \dots, \mathcal{V}_m$ con n_1, \dots, n_m punti, rispettivamente, tali che $\cup_{i=1}^m \mathcal{V}_i = \mathcal{V}$ e $\sum_{i=1}^m n_i = n$.
- A meno di rinominare gli indici, possiamo assumere che i primi n_1 punti stiano in \mathcal{V}_1 , i successivi n_2 stiano in \mathcal{V}_2 , e così via.
- Il grafo di similarità \mathcal{G}^* avrà m componenti connesse $\mathcal{G}_1, \dots, \mathcal{G}_m$, la matrice di similarità W^* e il Laplaciano normalizzato L^* del grafo saranno entrambe diagonali a blocchi

$$W^* = \text{diag}(W_1, \dots, W_m) \quad L^* = \text{diag}(L_1, \dots, L_m)$$

In cui \mathcal{G}_i, W_i, L_i sono grafo, matrice di similarità e Laplaciano normalizzato del cluster \mathcal{V}_i .

Analisi del caso ideale

Consideriamo il caso in cui i punti formano cluster ben separati.

- Supponiamo che ci siano m cluster $\mathcal{V}_1, \dots, \mathcal{V}_m$ con n_1, \dots, n_m punti, rispettivamente, tali che $\cup_{i=1}^m \mathcal{V}_i = \mathcal{V}$ e $\sum_{i=1}^m n_i = n$.
- A meno di rinominare gli indici, possiamo assumere che i primi n_1 punti stiano in \mathcal{V}_1 , i successivi n_2 stiano in \mathcal{V}_2 , e così via.
- Il grafo di similarità \mathcal{G}^* avrà m componenti connesse $\mathcal{G}_1, \dots, \mathcal{G}_m$, la matrice di similarità W^* e il Laplaciano normalizzato L^* del grafo saranno entrambe diagonali a blocchi

$$W^* = \text{diag}(W_1, \dots, W_m) \quad L^* = \text{diag}(L_1, \dots, L_m)$$

In cui \mathcal{G}_i, W_i, L_i sono grafo, matrice di similarità e Laplaciano normalizzato del cluster \mathcal{V}_i .

Analisi del caso ideale

Consideriamo il caso in cui i punti formano cluster ben separati.

- Supponiamo che ci siano m cluster $\mathcal{V}_1, \dots, \mathcal{V}_m$ con n_1, \dots, n_m punti, rispettivamente, tali che $\cup_{i=1}^m \mathcal{V}_i = \mathcal{V}$ e $\sum_{i=1}^m n_i = n$.
- A meno di rinominare gli indici, possiamo assumere che i primi n_1 punti stiano in \mathcal{V}_1 , i successivi n_2 stiano in \mathcal{V}_2 , e così via.
- Il grafo di similarità \mathcal{G}^* avrà m componenti connesse $\mathcal{G}_1, \dots, \mathcal{G}_m$, la matrice di similarità W^* e il Laplaciano normalizzato L^* del grafo saranno entrambe diagonali a blocchi

$$W^* = \text{diag}(W_1, \dots, W_m) \quad L^* = \text{diag}(L_1, \dots, L_m)$$

In cui \mathcal{G}_i, W_i, L_i sono grafo, matrice di similarità e Laplaciano normalizzato del cluster \mathcal{V}_i .

Analisi del caso ideale

Consideriamo il caso in cui i punti formano cluster ben separati.

- Supponiamo che ci siano m cluster $\mathcal{V}_1, \dots, \mathcal{V}_m$ con n_1, \dots, n_m punti, rispettivamente, tali che $\cup_{i=1}^m \mathcal{V}_i = \mathcal{V}$ e $\sum_{i=1}^m n_i = n$.
- A meno di rinominare gli indici, possiamo assumere che i primi n_1 punti stiano in \mathcal{V}_1 , i successivi n_2 stiano in \mathcal{V}_2 , e così via.
- Il grafo di similarità \mathcal{G}^* avrà m componenti connesse $\mathcal{G}_1, \dots, \mathcal{G}_m$, la matrice di similarità W^* e il Laplaciano normalizzato L^* del grafo saranno entrambe diagonali a blocchi

$$W^* = \text{diag}(W_1, \dots, W_m) \quad L^* = \text{diag}(L_1, \dots, L_m)$$

In cui \mathcal{G}_i, W_i, L_i sono grafo, matrice di similarità e Laplaciano normalizzato del cluster \mathcal{V}_i .

Analisi del caso ideale

Consideriamo il caso in cui i punti formano cluster ben separati.

- Supponiamo che ci siano m cluster $\mathcal{V}_1, \dots, \mathcal{V}_m$ con n_1, \dots, n_m punti, rispettivamente, tali che $\cup_{i=1}^m \mathcal{V}_i = \mathcal{V}$ e $\sum_{i=1}^m n_i = n$.
- A meno di rinominare gli indici, possiamo assumere che i primi n_1 punti stiano in \mathcal{V}_1 , i successivi n_2 stiano in \mathcal{V}_2 , e così via.
- Il grafo di similarità \mathcal{G}^* avrà m componenti connesse $\mathcal{G}_1, \dots, \mathcal{G}_m$, la matrice di similarità W^* e il Laplaciano normalizzato L^* del grafo saranno entrambe diagonali a blocchi

$$W^* = \text{diag}(W_1, \dots, W_m) \quad L^* = \text{diag}(L_1, \dots, L_m)$$

In cui \mathcal{G}_i, W_i, L_i sono grafo, matrice di similarità e Laplaciano normalizzato del cluster \mathcal{V}_i .

Analisi del caso ideale

Definizione

Il *vettore caratteristico* $\chi_j \in \mathbb{R}^n$ della j -esima componente connessa \mathcal{G}_j è definito da:

$$\chi_j(i) = \begin{cases} 1 & \text{se } \mathbf{v}_i \in \mathcal{G}_j, \\ 0 & \text{altrimenti.} \end{cases}$$

Osservazione

$$M^* \chi_j = \chi_j$$

Lemma (3)

Per ogni $j = 1, \dots, m$, $D^{1/2} \chi_j$ è autovettore di L^* relativo all'autovalore 0.

Analisi del caso ideale

Definizione

Il *vettore caratteristico* $\chi_j \in \mathbb{R}^n$ della j -esima componente connessa \mathcal{G}_j è definito da:

$$\chi_j(i) = \begin{cases} 1 & \text{se } \mathbf{v}_i \in \mathcal{G}_j, \\ 0 & \text{altrimenti.} \end{cases}$$

Osservazione

$$M^* \chi_j = \chi_j$$

Lemma (3)

Per ogni $j = 1, \dots, m$, $D^{1/2} \chi_j$ è autovettore di L^ relativo all'autovalore 0.*

Analisi del caso ideale

Definizione

Il *vettore caratteristico* $\chi_j \in \mathbb{R}^n$ della j -esima componente connessa \mathcal{G}_j è definito da:

$$\chi_j(i) = \begin{cases} 1 & \text{se } \mathbf{v}_i \in \mathcal{G}_j, \\ 0 & \text{altrimenti.} \end{cases}$$

Osservazione

$$M^* \chi_j = \chi_j$$

Lemma (3)

Per ogni $j = 1, \dots, m$, $D^{1/2} \chi_j$ è autovettore di L^* relativo all'autovalore 0.

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$

Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$

Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$



Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$

Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$



Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$



Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Analisi del caso ideale

Dimostrazione.

$$\begin{aligned}L^* D^{1/2} \chi_j &= \frac{1}{\alpha} (I - D^{1/2} M^* D^{-1/2}) D^{1/2} \chi_j \\ &= \frac{1}{\alpha} (D^{1/2} \chi_j - D^{1/2} M^* \chi_j) = 0\end{aligned}$$



Perciò l'autovalore $\lambda_1 = 0$ di L^* ha molteplicità m con autovettori normalizzati linearmente indipendenti $\phi_1^*, \dots, \phi_m^*$ dove

$$\phi_j^* = \frac{D^{1/2} \chi_j}{\|D^{1/2} \chi_j\|}$$

Teorema di convergenza (caso ideale)

Teorema

Supponiamo di avere un insieme di punti ideale composto da m cluster e sia $\mathbf{x}^{(0)}$ un qualsiasi vettore tale che $x_i^{(0)} > 0$ per ogni i e $(\mathbf{x}^{(0)})^T \mathbf{1} = 1$, allora

$$\|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| \leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}} \quad (5)$$

dove $c_i = \frac{\chi_i^T D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i}$ e d_j indica il grado del j -esimo nodo.

Teorema di convergenza (caso ideale)

Teorema

Supponiamo di avere un insieme di punti ideale composto da m cluster e sia $\mathbf{x}^{(0)}$ un qualsiasi vettore tale che $x_i^{(0)} > 0$ per ogni i e $(\mathbf{x}^{(0)})^T \mathbf{1} = 1$, allora

$$\|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| \leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}} \quad (5)$$

dove $c_i = \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i}$ e d_j indica il grado del j -esimo nodo.

Teorema di convergenza (caso ideale)

Teorema

Supponiamo di avere un insieme di punti ideale composto da m cluster e sia $\mathbf{x}^{(0)}$ un qualsiasi vettore tale che $x_i^{(0)} > 0$ per ogni i e $(\mathbf{x}^{(0)})^T \mathbf{1} = 1$, allora

$$\|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| \leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}} \quad (5)$$

dove $c_i = \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i}$ e d_j indica il grado del j -esimo nodo.

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} M^{*k} \mathbf{x}^{(0)} &= D^{-1/2} \left(\sum_{i=1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \\ &= D^{-1/2} \left(\sum_{i=1}^m \phi_i^* \phi_i^{*T} + \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \end{aligned}$$

Svolgiamo il prodotto a sinistra per $D^{-1/2}$ e a destra per $D^{1/2} \mathbf{x}^{(0)}$ e isoliamo il termine della prima sommatoria:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} M^{*k} \mathbf{x}^{(0)} &= D^{-1/2} \left(\sum_{i=1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \\ &= D^{-1/2} \left(\sum_{i=1}^m \phi_i^* \phi_i^{*T} + \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \end{aligned}$$

Svolgiamo il prodotto a sinistra per $D^{-1/2}$ e a destra per $D^{1/2} \mathbf{x}^{(0)}$ e isoliamo il termine della prima sommatoria:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} M^{*k} \mathbf{x}^{(0)} &= D^{-1/2} \left(\sum_{i=1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \\ &= D^{-1/2} \left(\sum_{i=1}^m \phi_i^* \phi_i^{*T} + \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \end{aligned}$$

Svolgiamo il prodotto a sinistra per $D^{-1/2}$ e a destra per $D^{1/2} \mathbf{x}^{(0)}$ e isoliamo il termine della prima sommatoria:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} M^{*k} \mathbf{x}^{(0)} &= D^{-1/2} \left(\sum_{i=1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \\ &= D^{-1/2} \left(\sum_{i=1}^m \phi_i^* \phi_i^{*T} + \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \end{aligned}$$

Svolgiamo il prodotto a sinistra per $D^{-1/2}$ e a destra per $D^{1/2} \mathbf{x}^{(0)}$ e isoliamo il termine della prima sommatoria:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} M^{*k} \mathbf{x}^{(0)} &= D^{-1/2} \left(\sum_{i=1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \\ &= D^{-1/2} \left(\sum_{i=1}^m \phi_i^* \phi_i^{*T} + \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \end{aligned}$$

Svolgiamo il prodotto a sinistra per $D^{-1/2}$ e a destra per $D^{1/2} \mathbf{x}^{(0)}$ e isoliamo il termine della prima sommatoria:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} D^{-1/2} \phi_i^* \phi_i^{*T} D^{1/2} \mathbf{x}^{(0)} &= D^{-1/2} \frac{D^{1/2} \chi_i}{\|D^{1/2} \chi_i\|} \frac{(D^{1/2} \chi_i)^T}{\|D^{1/2} \chi_i\|} D^{1/2} \mathbf{x}^{(0)} \\ &= \frac{\chi_i \chi_i^T D^{1/2} D^{1/2} \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \\ &= \frac{\chi_i^T D \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \chi_i \end{aligned}$$

Osserviamo inoltre che la quantità $\|D^{1/2} \chi_i\|^2$ è uguale alla somma dei gradi dei vertici (volume) in \mathcal{G}_i che può anche essere espressa come $\mathbf{1}^T D \chi_i$. Perciò possiamo scrivere:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} D^{-1/2} \phi_i^* \phi_i^{*T} D^{1/2} \mathbf{x}^{(0)} &= D^{-1/2} \frac{D^{1/2} \chi_i}{\|D^{1/2} \chi_i\|} \frac{(D^{1/2} \chi_i)^T}{\|D^{1/2} \chi_i\|} D^{1/2} \mathbf{x}^{(0)} \\ &= \frac{\chi_i \chi_i^T D^{1/2} D^{1/2} \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \\ &= \frac{\chi_i^T D \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \chi_i \end{aligned}$$

Osserviamo inoltre che la quantità $\|D^{1/2} \chi_i\|^2$ è uguale alla somma dei gradi dei vertici (volume) in \mathcal{G}_i che può anche essere espressa come $\mathbf{1}^T D \chi_i$. Perciò possiamo scrivere:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} D^{-1/2} \phi_i^* \phi_i^{*T} D^{1/2} \mathbf{x}^{(0)} &= D^{-1/2} \frac{D^{1/2} \chi_i}{\|D^{1/2} \chi_i\|} \frac{(D^{1/2} \chi_i)^T}{\|D^{1/2} \chi_i\|} D^{1/2} \mathbf{x}^{(0)} \\ &= \frac{\chi_i \chi_i^T D^{1/2} D^{1/2} \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \\ &= \frac{\chi_i^T D \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \chi_i \end{aligned}$$

Osserviamo inoltre che la quantità $\|D^{1/2} \chi_i\|^2$ è uguale alla somma dei gradi dei vertici (volume) in \mathcal{G}_i che può anche essere espressa come $\mathbf{1}^T D \chi_i$. Perciò possiamo scrivere:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} D^{-1/2} \phi_i^* \phi_i^{*T} D^{1/2} \mathbf{x}^{(0)} &= D^{-1/2} \frac{D^{1/2} \chi_i}{\|D^{1/2} \chi_i\|} \frac{(D^{1/2} \chi_i)^T}{\|D^{1/2} \chi_i\|} D^{1/2} \mathbf{x}^{(0)} \\ &= \frac{\chi_i \chi_i^T D^{1/2} D^{1/2} \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \\ &= \frac{\chi_i^T D \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \chi_i \end{aligned}$$

Osserviamo inoltre che la quantità $\|D^{1/2} \chi_i\|^2$ è uguale alla somma dei gradi dei vertici (volume) in \mathcal{G}_i che può anche essere espressa come $\mathbf{1}^T D \chi_i$. Perciò possiamo scrivere:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned} D^{-1/2} \phi_i^* \phi_i^{*T} D^{1/2} \mathbf{x}^{(0)} &= D^{-1/2} \frac{D^{1/2} \chi_i}{\|D^{1/2} \chi_i\|} \frac{(D^{1/2} \chi_i)^T}{\|D^{1/2} \chi_i\|} D^{1/2} \mathbf{x}^{(0)} \\ &= \frac{\chi_i \chi_i^T D^{1/2} D^{1/2} \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \\ &= \frac{\chi_i^T D \mathbf{x}^{(0)}}{\|D^{1/2} \chi_i\|^2} \chi_i \end{aligned}$$

Osserviamo inoltre che la quantità $\|D^{1/2} \chi_i\|^2$ è uguale alla somma dei gradi dei vertici (volume) in \mathcal{G}_i che può anche essere espressa come $\mathbf{1}^T D \chi_i$. Perciò possiamo scrivere:

Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned}
 \|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| &= \left\| \sum_{i=0}^m \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i} \chi_i - \sum_{i=0}^m c_i \chi_i + \right. \\
 &\quad \left. + D^{-1/2} \left(\sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \right\| \\
 &\leq \|D^{-1/2}\| \left\| \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right\| \|D^{1/2}\| \\
 &\leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}
 \end{aligned}$$



Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned}
 \|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| &= \left\| \sum_{i=0}^m \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i} \chi_i - \sum_{i=0}^m c_i \chi_i + \right. \\
 &\quad \left. + D^{-1/2} \left(\sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \right\| \\
 &\leq \|D^{-1/2}\| \left\| \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right\| \|D^{1/2}\| \\
 &\leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}
 \end{aligned}$$



Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned}
 \|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| &= \left\| \sum_{i=0}^m \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i} \chi_i - \sum_{i=0}^m c_i \chi_i + \right. \\
 &\quad \left. + D^{-1/2} \left(\sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \right\| \\
 &\leq \|D^{-1/2}\| \left\| \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right\| \|D^{1/2}\| \\
 &\leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}
 \end{aligned}$$



Teorema di convergenza (caso ideale)

Dimostrazione.

$$\begin{aligned}
 \|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| &= \left\| \sum_{i=0}^m \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i} \chi_i - \sum_{i=0}^m c_i \chi_i + \right. \\
 &\quad \left. + D^{-1/2} \left(\sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right) D^{1/2} \mathbf{x}^{(0)} \right\| \\
 &\leq \|D^{-1/2}\| \left\| \sum_{i=m+1}^n (1 - \alpha \lambda_i)^k \phi_i^* \phi_i^{*T} \right\| \|D^{1/2}\| \\
 &\leq \max_{i>m} |1 - \alpha \lambda_i|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}
 \end{aligned}$$



Teorema di convergenza (caso ideale)

Osservazione

Notiamo che si può sempre scegliere $\alpha \in [0, 1]$ in modo da avere $\lambda_{m+1} = \arg \max_{i>m} |1 - \alpha \lambda_i|$. In questo modo la diseuguaglianza del teorema diventa:

$$\|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| \leq |1 - \alpha \lambda_{m+1}|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}$$

Teorema di convergenza (caso ideale)

Osservazione

Notiamo che si può sempre scegliere $\alpha \in [0, 1]$ in modo da avere $\lambda_{m+1} = \arg \max_{i>m} |1 - \alpha \lambda_i|$. In questo modo la disuguaglianza del teorema diventa:

$$\|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| \leq |1 - \alpha \lambda_{m+1}|^k \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}$$

Analisi del caso generale

In generale i grafi da trattare avranno m componenti quasi connesse. Si può vedere il caso generale per un insieme di punti \mathcal{V} come una perturbazione del caso ideale:

- Sia $W = W^* + E$ la matrice di similarità, dove W^* è la matrice di similarità corrispondente ai cluster effettivi, che è diagonale a blocchi e simmetrica.
- W^* è ottenuta rimpiazzando con degli zeri gli elementi che non stanno nei blocchi diagonali di W e sommando gli elementi sostituiti in ogni riga all'elemento diagonale.

Osservazioni

- W e W^* hanno la stessa matrice dei gradi D .
- $E\mathbf{1} = \mathbf{0}$

Analisi del caso generale

In generale i grafi da trattare avranno m componenti quasi connesse. Si può vedere il caso generale per un insieme di punti \mathcal{V} come una perturbazione del caso ideale:

- Sia $W = W^* + E$ la matrice di similarità, dove W^* è la matrice di similarità corrispondente ai cluster effettivi, che è diagonale a blocchi e simmetrica.
- W^* è ottenuta rimpiazzando con degli zeri gli elementi che non stanno nei blocchi diagonali di W e sommando gli elementi sostituiti in ogni riga all'elemento diagonale.

Osservazioni

- W e W^* hanno la stessa matrice dei gradi D .
- $E\mathbf{1} = \mathbf{0}$

Analisi del caso generale

In generale i grafi da trattare avranno m componenti quasi connesse. Si può vedere il caso generale per un insieme di punti \mathcal{V} come una perturbazione del caso ideale:

- Sia $W = W^* + E$ la matrice di similarità, dove W^* è la matrice di similarità corrispondente ai cluster effettivi, che è diagonale a blocchi e simmetrica.
- W^* è ottenuta rimpiazzando con degli zeri gli elementi che non stanno nei blocchi diagonali di W e sommando gli elementi sostituiti in ogni riga all'elemento diagonale.

Osservazioni

- W e W^* hanno la stessa matrice dei gradi D .
- $E\mathbf{1} = \mathbf{0}$

Analisi del caso generale

In generale i grafi da trattare avranno m componenti quasi connesse. Si può vedere il caso generale per un insieme di punti \mathcal{V} come una perturbazione del caso ideale:

- Sia $W = W^* + E$ la matrice di similarità, dove W^* è la matrice di similarità corrispondente ai cluster effettivi, che è diagonale a blocchi e simmetrica.
- W^* è ottenuta rimpiazzando con degli zeri gli elementi che non stanno nei blocchi diagonali di W e sommando gli elementi sostituiti in ogni riga all'elemento diagonale.

Osservazioni

- W e W^* hanno la stessa matrice dei gradi D .
- $E\mathbf{1} = \mathbf{0}$

Analisi del caso generale

In generale i grafi da trattare avranno m componenti quasi connesse. Si può vedere il caso generale per un insieme di punti \mathcal{V} come una perturbazione del caso ideale:

- Sia $W = W^* + E$ la matrice di similarità, dove W^* è la matrice di similarità corrispondente ai cluster effettivi, che è diagonale a blocchi e simmetrica.
- W^* è ottenuta rimpiazzando con degli zeri gli elementi che non stanno nei blocchi diagonali di W e sommando gli elementi sostituiti in ogni riga all'elemento diagonale.

Osservazioni

- W e W^* hanno la stessa matrice dei gradi D .
- $E1 = 0$

Analisi del caso generale

In generale i grafi da trattare avranno m componenti quasi connesse. Si può vedere il caso generale per un insieme di punti \mathcal{V} come una perturbazione del caso ideale:

- Sia $W = W^* + E$ la matrice di similarità, dove W^* è la matrice di similarità corrispondente ai cluster effettivi, che è diagonale a blocchi e simmetrica.
- W^* è ottenuta rimpiazzando con degli zeri gli elementi che non stanno nei blocchi diagonali di W e sommando gli elementi sostituiti in ogni riga all'elemento diagonale.

Osservazioni

- W e W^* hanno la stessa matrice dei gradi D .
- $E\mathbf{1} = \mathbf{0}$

Esempio

$$W = \begin{bmatrix} 0 & 20 & 50 & 1 & 2 & 1 \\ 20 & 0 & 30 & 0 & 1 & 1 \\ 50 & 30 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 25 & 40 \\ 2 & 1 & 0 & 25 & 0 & 30 \\ 1 & 1 & 1 & 40 & 30 & 0 \end{bmatrix}$$

$$W^* = \begin{bmatrix} 4 & 20 & 50 & 0 & 0 & 0 \\ 20 & 2 & 30 & 0 & 0 & 0 \\ 50 & 30 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 25 & 40 \\ 0 & 0 & 0 & 25 & 3 & 30 \\ 0 & 0 & 0 & 40 & 30 & 3 \end{bmatrix}$$

$$E = \begin{bmatrix} -4 & 0 & 0 & 1 & 2 & 1 \\ 0 & -2 & 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 1 & 0 & 1 \\ 1 & 0 & 1 & -2 & 0 & 0 \\ 2 & 1 & 0 & 0 & -3 & 0 \\ 1 & 1 & 1 & 0 & 0 & -3 \end{bmatrix}$$

Esempio

$$W = \begin{bmatrix} 0 & 20 & 50 & 1 & 2 & 1 \\ 20 & 0 & 30 & 0 & 1 & 1 \\ 50 & 30 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 25 & 40 \\ 2 & 1 & 0 & 25 & 0 & 30 \\ 1 & 1 & 1 & 40 & 30 & 0 \end{bmatrix}$$

$$W^* = \begin{bmatrix} 4 & 20 & 50 & 0 & 0 & 0 \\ 20 & 2 & 30 & 0 & 0 & 0 \\ 50 & 30 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 25 & 40 \\ 0 & 0 & 0 & 25 & 3 & 30 \\ 0 & 0 & 0 & 40 & 30 & 3 \end{bmatrix}$$

$$E = \begin{bmatrix} -4 & 0 & 0 & 1 & 2 & 1 \\ 0 & -2 & 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 1 & 0 & 1 \\ 1 & 0 & 1 & -2 & 0 & 0 \\ 2 & 1 & 0 & 0 & -3 & 0 \\ 1 & 1 & 1 & 0 & 0 & -3 \end{bmatrix}$$

Esempio

$$W = \begin{bmatrix} 0 & 20 & 50 & 1 & 2 & 1 \\ 20 & 0 & 30 & 0 & 1 & 1 \\ 50 & 30 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 25 & 40 \\ 2 & 1 & 0 & 25 & 0 & 30 \\ 1 & 1 & 1 & 40 & 30 & 0 \end{bmatrix}$$

$$W^* = \begin{bmatrix} 4 & 20 & 50 & 0 & 0 & 0 \\ 20 & 2 & 30 & 0 & 0 & 0 \\ 50 & 30 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 25 & 40 \\ 0 & 0 & 0 & 25 & 3 & 30 \\ 0 & 0 & 0 & 40 & 30 & 3 \end{bmatrix}$$

$$E = \begin{bmatrix} -4 & 0 & 0 & 1 & 2 & 1 \\ 0 & -2 & 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 1 & 0 & 1 \\ 1 & 0 & 1 & -2 & 0 & 0 \\ 2 & 1 & 0 & 0 & -3 & 0 \\ 1 & 1 & 1 & 0 & 0 & -3 \end{bmatrix}$$

Teorema di convergenza (caso generale)

Lemma

Sia $W = W^* + E$ la matrice di similarità relativa all'insieme di punti \mathcal{V} . Il laplaciano normalizzato del grafo corrispondente è $L = L^* - \bar{E}$. Dove:

- L^* è il laplaciano normalizzato del grafo corrispondente a W^* ,
- $\bar{E} = D^{-1/2} E D^{-1/2}$.

Dimostrazione.

$$\begin{aligned} L &= I - D^{-1/2}(W^* + E)D^{-1/2} \\ &= L^* - D^{-1/2} E D^{-1/2} \end{aligned}$$



Teorema di convergenza (caso generale)

Lemma

Sia $W = W^* + E$ la matrice di similarità relativa all'insieme di punti \mathcal{V} . Il laplaciano normalizzato del grafo corrispondente è $L = L^* - \bar{E}$. Dove:

- L^* è il laplaciano normalizzato del grafo corrispondente a W^* ,
- $\bar{E} = D^{-1/2} E D^{-1/2}$.

Dimostrazione.

$$\begin{aligned} L &= I - D^{-1/2}(W^* + E)D^{-1/2} \\ &= L^* - D^{-1/2} E D^{-1/2} \end{aligned}$$



Teorema di convergenza (caso generale)

Lemma

Sia $W = W^* + E$ la matrice di similarità relativa all'insieme di punti \mathcal{V} . Il laplaciano normalizzato del grafo corrispondente è $L = L^* - \bar{E}$. Dove:

- L^* è il laplaciano normalizzato del grafo corrispondente a W^* ,
- $\bar{E} = D^{-1/2} E D^{-1/2}$.

Dimostrazione.

$$\begin{aligned} L &= I - D^{-1/2}(W^* + E)D^{-1/2} \\ &= L^* - D^{-1/2} E D^{-1/2} \end{aligned}$$



Teorema di convergenza (caso generale)

Lemma

Sia $W = W^* + E$ la matrice di similarità relativa all'insieme di punti \mathcal{V} . Il laplaciano normalizzato del grafo corrispondente è $L = L^* - \bar{E}$. Dove:

- L^* è il laplaciano normalizzato del grafo corrispondente a W^* ,
- $\bar{E} = D^{-1/2} E D^{-1/2}$.

Dimostrazione.

$$\begin{aligned} L &= I - D^{-1/2} (W^* + E) D^{-1/2} \\ &= L^* - D^{-1/2} E D^{-1/2} \end{aligned}$$



Teorema di convergenza (caso generale)

Lemma

Sia $W = W^* + E$ la matrice di similarità relativa all'insieme di punti \mathcal{V} . Il laplaciano normalizzato del grafo corrispondente è $L = L^* - \bar{E}$. Dove:

- L^* è il laplaciano normalizzato del grafo corrispondente a W^* ,
- $\bar{E} = D^{-1/2} E D^{-1/2}$.

Dimostrazione.

$$\begin{aligned} L &= I - D^{-1/2}(W^* + E)D^{-1/2} \\ &= L^* - D^{-1/2} E D^{-1/2} \end{aligned}$$



Teorema di convergenza (caso generale)

Lemma

Sia $W = W^* + E$ la matrice di similarità relativa all'insieme di punti \mathcal{V} . Il laplaciano normalizzato del grafo corrispondente è $L = L^* - \bar{E}$. Dove:

- L^* è il laplaciano normalizzato del grafo corrispondente a W^* ,
- $\bar{E} = D^{-1/2} E D^{-1/2}$.

Dimostrazione.

$$\begin{aligned} L &= I - D^{-1/2}(W^* + E)D^{-1/2} \\ &= L^* - D^{-1/2} E D^{-1/2} \end{aligned}$$



Teorema di convergenza (caso generale)

Gli autovalori e gli autovettori di L possono essere scritti come:

$$\lambda_i = \lambda_i^* + \tilde{\lambda}_i \qquad \phi_i = \phi_i^* + \tilde{\phi}_i$$

dove λ_i^* e ϕ_i^* sono autovalori e autovettori di L^* , $\tilde{\lambda}_i$ e $\tilde{\phi}_i$ dipendono in modo continuo dalle entrate della matrice \bar{E} .

Assumeremo che $\|E\|$ e di conseguenza $\|\bar{E}\|$ siano abbastanza piccoli in modo che $|\tilde{\lambda}_i|$ e $\|\tilde{\phi}_i\|$ siano piccoli anch'essi.

Teorema di convergenza (caso generale)

Gli autovalori e gli autovettori di L possono essere scritti come:

$$\lambda_i = \lambda_i^* + \tilde{\lambda}_i \qquad \phi_i = \phi_i^* + \tilde{\phi}_i$$

dove λ_i^* e ϕ_i^* sono autovalori e autovettori di L^* , $\tilde{\lambda}_i$ e $\tilde{\phi}_i$ dipendono in modo continuo dalle entrate della matrice \bar{E} .

Assumeremo che $\|E\|$ e di conseguenza $\|\bar{E}\|$ siano abbastanza piccoli in modo che $|\tilde{\lambda}_i|$ e $\|\tilde{\phi}_i\|$ siano piccoli anch'essi.

Teorema di convergenza (caso generale)

Teorema

Supponiamo di avere un insieme di punti composto da k cluster e sia $\mathbf{x}^{(0)}$ un qualsiasi vettore tale che $x_i^{(0)} > 0$ per ogni i e $(\mathbf{x}^{(0)})^T \mathbf{1} = 1$, allora

$$\|M^{*k} \mathbf{x}^{(0)} - \sum_{i=0}^m c_i \chi_i\| \leq \left(\sum_{i=1}^m (2\|\tilde{\phi}_i\| + \|\tilde{\phi}_i\|^2) + \max_{l>m} |1 - \alpha \lambda_l|^k \right) \frac{\max_j \sqrt{d_j}}{\min_j \sqrt{d_j}}$$

dove $c_i = (1 - \alpha \tilde{\lambda}_i) \frac{\chi_i D \mathbf{x}^{(0)}}{\mathbf{1}^T D \chi_i}$ e d_j indica il grado del j -esimo nodo.

Costruzione dell'algoritmo

- Simon e Ando hanno dimostrato (1961) che in un sistema dinamico discreto come quello da noi costruito, i sottosistemi raggiungono un equilibrio locale molto prima che l'intero sistema raggiunga un equilibrio globale.
- Per questo motivo un algoritmo di clustering efficiente dovrebbe arrestarsi quando viene rivelato un equilibrio locale, per poi riconoscere i cluster in base a come si sono aggregati i punti.
- A causa delle perturbazioni, al crescere di m , diventa più difficile distinguere i cluster dal vettore $M^k \mathbf{x}^{(0)}$ usando l'algoritmo *k-means*.
- L'idea è di dividere, ad ogni chiamata dell'algoritmo, i punti in due macrogruppi su cui poi agire ricorsivamente.

Costruzione dell'algoritmo

- Simon e Ando hanno dimostrato (1961) che in un sistema dinamico discreto come quello da noi costruito, i sottosistemi raggiungono un equilibrio locale molto prima che l'intero sistema raggiunga un equilibrio globale.
- Per questo motivo un algoritmo di clustering efficiente dovrebbe arrestarsi quando viene rivelato un equilibrio locale, per poi riconoscere i cluster in base a come si sono aggregati i punti.
- A causa delle perturbazioni, al crescere di m , diventa più difficile distinguere i cluster dal vettore $M^k \mathbf{x}^{(0)}$ usando l'algoritmo *k-means*.
- L'idea è di dividere, ad ogni chiamata dell'algoritmo, i punti in due macrogruppi su cui poi agire ricorsivamente.

Costruzione dell'algoritmo

- Simon e Ando hanno dimostrato (1961) che in un sistema dinamico discreto come quello da noi costruito, i sottosistemi raggiungono un equilibrio locale molto prima che l'intero sistema raggiunga un equilibrio globale.
- Per questo motivo un algoritmo di clustering efficiente dovrebbe arrestarsi quando viene rivelato un equilibrio locale, per poi riconoscere i cluster in base a come si sono aggregati i punti.
- A causa delle perturbazioni, al crescere di m , diventa più difficile distinguere i cluster dal vettore $M^k \mathbf{x}^{(0)}$ usando l'algoritmo *k-means*.
- L'idea è di dividere, ad ogni chiamata dell'algoritmo, i punti in due macrogruppi su cui poi agire ricorsivamente.

Costruzione dell'algoritmo

- Simon e Ando hanno dimostrato (1961) che in un sistema dinamico discreto come quello da noi costruito, i sottosistemi raggiungono un equilibrio locale molto prima che l'intero sistema raggiunga un equilibrio globale.
- Per questo motivo un algoritmo di clustering efficiente dovrebbe arrestarsi quando viene rivelato un equilibrio locale, per poi riconoscere i cluster in base a come si sono aggregati i punti.
- A causa delle perturbazioni, al crescere di m , diventa più difficile distinguere i cluster dal vettore $M^k \mathbf{x}^{(0)}$ usando l'algoritmo *k-means*.
- L'idea è di dividere, ad ogni chiamata dell'algoritmo, i punti in due macrogruppi su cui poi agire ricorsivamente.

Costruzione dell'algoritmo

- Simon e Ando hanno dimostrato (1961) che in un sistema dinamico discreto come quello da noi costruito, i sottosistemi raggiungono un equilibrio locale molto prima che l'intero sistema raggiunga un equilibrio globale.
- Per questo motivo un algoritmo di clustering efficiente dovrebbe arrestarsi quando viene rivelato un equilibrio locale, per poi riconoscere i cluster in base a come si sono aggregati i punti.
- A causa delle perturbazioni, al crescere di m , diventa più difficile distinguere i cluster dal vettore $M^k \mathbf{x}^{(0)}$ usando l'algoritmo *k-means*.
- L'idea è di dividere, ad ogni chiamata dell'algoritmo, i punti in due macrogruppi su cui poi agire ricorsivamente.

Come dividere i punti in due gruppi?

Ricordando che abbiamo inizializzato il vettore $\mathbf{x}^{(0)}$ scegliendo n punti random in $[0, b]$, immaginiamo di avere ordinato in maniera crescente il vettore $\mathbf{x}^{(k)}$ e definiamo il *gap* fra due componenti consecutive di $\mathbf{x}^{(k)}$ come

$$gap(i) = \begin{cases} x_{i+1}^{(k)} - x_i^{(k)} & \text{se } x_{i+1}^{(k)} - x_i^{(k)} > \frac{b}{2n}, \\ 0 & \text{altrimenti} \end{cases}$$

Osservazioni

- Il *gap* ha il compito di rilevare divari significativi fra i punti.
- Esso è direttamente proporzionale a b e inversamente a n .

La divisione dei punti sarà data dal primo *gap* non nullo trovato nel corso delle iterazioni.

Come dividere i punti in due gruppi?

Ricordando che abbiamo inizializzato il vettore $\mathbf{x}^{(0)}$ scegliendo n punti random in $[0, b]$, immaginiamo di avere ordinato in maniera crescente il vettore $\mathbf{x}^{(k)}$ e definiamo il *gap* fra due componenti consecutive di $\mathbf{x}^{(k)}$ come

$$gap(i) = \begin{cases} x_{i+1}^{(k)} - x_i^{(k)} & \text{se } x_{i+1}^{(k)} - x_i^{(k)} > \frac{b}{2n}, \\ 0 & \text{altrimenti} \end{cases}$$

Osservazioni

- Il *gap* ha il compito di rilevare divari significativi fra i punti.
- Esso è direttamente proporzionale a b e inversamente a n .

La divisione dei punti sarà data dal primo *gap* non nullo trovato nel corso delle iterazioni.

Come dividere i punti in due gruppi?

Ricordando che abbiamo inizializzato il vettore $\mathbf{x}^{(0)}$ scegliendo n punti random in $[0, b]$, immaginiamo di avere ordinato in maniera crescente il vettore $\mathbf{x}^{(k)}$ e definiamo il *gap* fra due componenti consecutive di $\mathbf{x}^{(k)}$ come

$$gap(i) = \begin{cases} x_{i+1}^{(k)} - x_i^{(k)} & \text{se } x_{i+1}^{(k)} - x_i^{(k)} > \frac{b}{2n}, \\ 0 & \text{altrimenti} \end{cases}$$

Osservazioni

- Il *gap* ha il compito di rilevare divari significativi fra i punti.
- Esso è direttamente proporzionale a b e inversamente a n .

La divisione dei punti sarà data dal primo *gap* non nullo trovato nel corso delle iterazioni.

Come dividere i punti in due gruppi?

Ricordando che abbiamo inizializzato il vettore $\mathbf{x}^{(0)}$ scegliendo n punti random in $[0, b]$, immaginiamo di avere ordinato in maniera crescente il vettore $\mathbf{x}^{(k)}$ e definiamo il *gap* fra due componenti consecutive di $\mathbf{x}^{(k)}$ come

$$gap(i) = \begin{cases} x_{i+1}^{(k)} - x_i^{(k)} & \text{se } x_{i+1}^{(k)} - x_i^{(k)} > \frac{b}{2n}, \\ 0 & \text{altrimenti} \end{cases}$$

Osservazioni

- Il *gap* ha il compito di rilevare divari significativi fra i punti.
- Esso è direttamente proporzionale a b e inversamente a n .

La divisione dei punti sarà data dal primo *gap* non nullo trovato nel corso delle iterazioni.

Come dividere i punti in due gruppi?

Ricordando che abbiamo inizializzato il vettore $\mathbf{x}^{(0)}$ scegliendo n punti random in $[0, b]$, immaginiamo di avere ordinato in maniera crescente il vettore $\mathbf{x}^{(k)}$ e definiamo il *gap* fra due componenti consecutive di $\mathbf{x}^{(k)}$ come

$$gap(i) = \begin{cases} x_{i+1}^{(k)} - x_i^{(k)} & \text{se } x_{i+1}^{(k)} - x_i^{(k)} > \frac{b}{2n}, \\ 0 & \text{altrimenti} \end{cases}$$

Osservazioni

- Il *gap* ha il compito di rilevare divari significativi fra i punti.
- Esso è direttamente proporzionale a b e inversamente a n .

La divisione dei punti sarà data dal primo *gap* non nullo trovato nel corso delle iterazioni.

Criterio d'arresto

Basandoci sul Teorema 2 un buon criterio per arrestare il processo di mescolamento e calcolare la *gap* sarebbe quello di stoppare le iterazioni quando il valore $(1 - \alpha\lambda_{m+1})^k$ è minore di una certa soglia di tolleranza.

Il nostro intento però è quello di evitare di calcolare autovalori, ed inoltre non è detto che m sia noto.

Nella pratica quindi la tolleranza sarà relativa al vettore del processo di mescolamento $\mathbf{x}^{(k)}$.

Criterio d'arresto

Basandoci sul Teorema 2 un buon criterio per arrestare il processo di mescolamento e calcolare la *gap* sarebbe quello di stoppare le iterazioni quando il valore $(1 - \alpha\lambda_{m+1})^k$ è minore di una certa soglia di tolleranza.

Il nostro intento però è quello di evitare di calcolare autovalori, ed inoltre non è detto che m sia noto.

Nella pratica quindi la tolleranza sarà relativa al vettore del processo di mescolamento $\mathbf{x}^{(k)}$.

Criterio d'arresto

Basandoci sul Teorema 2 un buon criterio per arrestare il processo di mescolamento e calcolare la *gap* sarebbe quello di stoppare le iterazioni quando il valore $(1 - \alpha\lambda_{m+1})^k$ è minore di una certa soglia di tolleranza.

Il nostro intento però è quello di evitare di calcolare autovalori, ed inoltre non è detto che m sia noto.

Nella pratica quindi la tolleranza sarà relativa al vettore del processo di mescolamento $\mathbf{x}^{(k)}$.

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}$, $y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- **if** $\epsilon < \epsilon_{min}$ **or** $k > k_{max}$ **then return** il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1))$, $C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- while il *gap* è nullo
- while $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}$, $y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- endwhile
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- if $\epsilon < \epsilon_{min}$ or $k > k_{max}$ then return il cluster C
- $\epsilon = \epsilon/2$
- endwhile
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1))$, $C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
 - **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
 - $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}, y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
 - **endwhile**
 - Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
 - **if** $\epsilon < \epsilon_{min}$ or $k > k_{max}$ **then return** il cluster C
 - $\epsilon = \epsilon/2$
 - **endwhile**
 - Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
 - $C(i_1) = \text{cluster}(M(i_1, i_1)), C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}, y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- **if** $\epsilon < \epsilon_{min}$ **or** $k > k_{max}$ **then return** il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1)), C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}$, $y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- if $\epsilon < \epsilon_{min}$ or $k > k_{max}$ then return il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1))$, $C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}, y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- **if** $\epsilon < \epsilon_{min}$ **or** $k > k_{max}$ **then return** il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1)), C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}, y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- **if** $\epsilon < \epsilon_{min}$ **or** $k > k_{max}$ **then return** il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1)), C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}, y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- **if** $\epsilon < \epsilon_{min}$ **or** $k > k_{max}$ **then return** il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1)), C(i_2) = \text{cluster}(M(i_2, i_2))$

Algoritmo (2)

$C = \text{cluster}(M)$

- Inizializziamo $\mathbf{x}^{(0)}$ scegliendo n punti in $[0, b]$ in modo casuale
- **while** il *gap* è nullo
- **while** $|y^{(k+1)} - y^{(k)}| > \epsilon$
- $\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)}$, $y^{(k+1)} = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$
- **endwhile**
- Ordiniamo $\mathbf{x}^{(k+1)}$ e troviamo il *gap* più grande
- **if** $\epsilon < \epsilon_{min}$ **or** $k > k_{max}$ **then return** il cluster C
- $\epsilon = \epsilon/2$
- **endwhile**
- Dividiamo gli indici di $\mathbf{x}^{(k+1)}$ in base al *gap* in i_1 e i_2
- $C(i_1) = \text{cluster}(M(i_1, i_1))$, $C(i_2) = \text{cluster}(M(i_2, i_2))$

Analisi del costo computazionale

Assumiamo per semplicità che $n = 2^j$ e che ogni cluster contenga $\frac{n}{m}$ punti, assumiamo inoltre che $m = 2^l$ e che in ogni chiamata ricorsiva i punti vengano divisi in due cluster con la stessa cardinalità.

In ogni chiamata ricorsiva l'algoritmo esegue al più k_{max} prodotti matrice-vettore, se le operazioni sono svolte su matrici sparse, questo richiede $O(\bar{n}k_{max})$.

Inoltre se non si trova nel caso base, l'algoritmo esegue un ordinamento di n numeri (costo $O(n \log n)$) e due chiamate ricorsive.

Sia $T(n, m)$ il tempo impiegato a trovare m cluster in un insieme di taglia n :

Analisi del costo computazionale

Assumiamo per semplicità che $n = 2^j$ e che ogni cluster contenga $\frac{n}{m}$ punti, assumiamo inoltre che $m = 2^l$ e che in ogni chiamata ricorsiva i punti vengano divisi in due cluster con la stessa cardinalità.

In ogni chiamata ricorsiva l'algoritmo esegue al più k_{max} prodotti matrice-vettore, se le operazioni sono svolte su matrici sparse, questo richiede $O(\bar{n}k_{max})$.

Inoltre se non si trova nel caso base, l'algoritmo esegue un ordinamento di n numeri (costo $O(n \log n)$) e due chiamate ricorsive.

Sia $T(n, m)$ il tempo impiegato a trovare m cluster in un insieme di taglia n :

Analisi del costo computazionale

Assumiamo per semplicità che $n = 2^j$ e che ogni cluster contenga $\frac{n}{m}$ punti, assumiamo inoltre che $m = 2^l$ e che in ogni chiamata ricorsiva i punti vengano divisi in due cluster con la stessa cardinalità.

In ogni chiamata ricorsiva l'algoritmo esegue al più k_{max} prodotti matrice-vettore, se le operazioni sono svolte su matrici sparse, questo richiede $O(\bar{n}k_{max})$.

Inoltre se non si trova nel caso base, l'algoritmo esegue un ordinamento di n numeri (costo $O(n \log n)$) e due chiamate ricorsive.

Sia $T(n, m)$ il tempo impiegato a trovare m cluster in un insieme di taglia n :

Analisi del costo computazionale

Assumiamo per semplicità che $n = 2^j$ e che ogni cluster contenga $\frac{n}{m}$ punti, assumiamo inoltre che $m = 2^l$ e che in ogni chiamata ricorsiva i punti vengano divisi in due cluster con la stessa cardinalità.

In ogni chiamata ricorsiva l'algoritmo esegue al più k_{max} prodotti matrice-vettore, se le operazioni sono svolte su matrici sparse, questo richiede $O(\bar{n}k_{max})$.

Inoltre se non si trova nel caso base, l'algoritmo esegue un ordinamento di n numeri (costo $O(n \log n)$) e due chiamate ricorsive.

Sia $T(n, m)$ il tempo impiegato a trovare m cluster in un insieme di taglia n :

Analisi del costo computazionale

$$\begin{aligned}
 T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\
 &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \\
 &\vdots \\
 &= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s} \\
 &= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log mc\bar{n}k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s \\
 &= O(\bar{n} \log mk_{max}) + O(n \log n \log m)
 \end{aligned}$$

Analisi del costo computazionale

$$\begin{aligned}
 T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\
 &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \\
 &\vdots \\
 &= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s} \\
 &= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log mc\bar{n}k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s \\
 &= O(\bar{n} \log mk_{max}) + O(n \log n \log m)
 \end{aligned}$$

Analisi del costo computazionale

$$\begin{aligned}
 T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\
 &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \\
 &\vdots \\
 &= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s} \\
 &= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log m c \bar{n} k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s \\
 &= O(\bar{n} \log m k_{max}) + O(n \log n \log m)
 \end{aligned}$$

Analisi del costo computazionale

$$\begin{aligned}
 T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\
 &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \\
 &\vdots \\
 &= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s} \\
 &= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log mc\bar{n}k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s \\
 &= O(\bar{n} \log mk_{max}) + O(n \log n \log m)
 \end{aligned}$$

Analisi del costo computazionale

$$\begin{aligned} T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\ &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \end{aligned}$$

⋮

$$= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s}$$

$$= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log mc\bar{n}k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s$$

$$= O(\bar{n} \log mk_{max}) + O(n \log n \log m)$$

Analisi del costo computazionale

$$\begin{aligned}
 T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\
 &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \\
 &\vdots \\
 &= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s} \\
 &= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log mc\bar{n}k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s \\
 &= O(\bar{n} \log mk_{max}) + O(n \log n \log m)
 \end{aligned}$$

Analisi del costo computazionale

$$\begin{aligned}
 T(n, m) &= 2T\left(\frac{n}{2}\right) + c\bar{n}k_{max} + n \log n \\
 &= 4T\left(\frac{n}{4}\right) + 2c\bar{n}k_{max} + n \log n + n \log \frac{n}{2} \\
 &\vdots \\
 &= 2^l T\left(\frac{n}{2^l}\right) + lc\bar{n}k_{max} + n \sum_{s=0}^{l-1} \log \frac{n}{2^s} \\
 &= 2^l c \frac{\bar{n}}{2^l} k_{max} + \log mc\bar{n}k_{max} + n \log n \log m - n \sum_{s=0}^{l-1} s \\
 &= O(\bar{n} \log mk_{max}) + O(n \log n \log m)
 \end{aligned}$$

Analisi del costo computazionale

In maniera simile si vede che, se assumiamo una divisione diversa dei dati in ogni chiamata ricorsiva, il tempo di calcolo rimane invariato.

Il caso peggiore si trova quando l'insieme dei punti è composto da un cluster "grande" di circa $\frac{n}{2}$ punti, e dai rimanenti $m - 1$ cluster che contengono gli altri $\frac{n}{2}$ punti.

In questo caso il tempo di calcolo dipende da quando il cluster grande viene individuato. Se questo viene individuato presto nella ricorsione, il tempo di calcolo rimane

$O(\bar{n} \log mk_{max}) + O(n \log n \log m)$, altrimenti, se ad esempio viene individuato per ultimo, il tempo di calcolo diventa

$O(\bar{n}mk_{max}) + O(nm \log n)$.

Analisi del costo computazionale

In maniera simile si vede che, se assumiamo una divisione diversa dei dati in ogni chiamata ricorsiva, il tempo di calcolo rimane invariato.

Il caso peggiore si trova quando l'insieme dei punti è composto da un cluster "grande" di circa $\frac{n}{2}$ punti, e dai rimanenti $m - 1$ cluster che contengono gli altri $\frac{n}{2}$ punti.

In questo caso il tempo di calcolo dipende da quando il cluster grande viene individuato. Se questo viene individuato presto nella ricorsione, il tempo di calcolo rimane

$O(\bar{n} \log mk_{max}) + O(n \log n \log m)$, altrimenti, se ad esempio viene individuato per ultimo, il tempo di calcolo diventa

$O(\bar{n}mk_{max}) + O(nm \log n)$.

Analisi del costo computazionale

In maniera simile si vede che, se assumiamo una divisione diversa dei dati in ogni chiamata ricorsiva, il tempo di calcolo rimane invariato.

Il caso peggiore si trova quando l'insieme dei punti è composto da un cluster "grande" di circa $\frac{n}{2}$ punti, e dai rimanenti $m - 1$ cluster che contengono gli altri $\frac{n}{2}$ punti.

In questo caso il tempo di calcolo dipende da quando il cluster grande viene individuato. Se questo viene individuato presto nella ricorsione, il tempo di calcolo rimane

$O(\bar{n} \log mk_{max}) + O(n \log n \log m)$, altrimenti, se ad esempio viene individuato per ultimo, il tempo di calcolo diventa

$O(\bar{n}mk_{max}) + O(nm \log n)$.

Connessioni con l'algoritmo NCut (Shi-Malik)

Definizione

$$\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} w_{ij}$$

$$a(\mathcal{A}) = \text{cut}(\mathcal{A}, \mathcal{V})$$

$$\text{Ncut}(\mathcal{V}_1, \dots, \mathcal{V}_k) = \sum_{i=0}^k \frac{\text{cut}(\mathcal{V}_i, \mathcal{V} - \mathcal{V}_i)}{a(\mathcal{V}_i)}$$

Quest'ultima quantità viene detta *k-NCut*

Connessioni con l'algoritmo NCut (Shi-Malik)

Definizione

$$\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} w_{ij}$$

$$a(\mathcal{A}) = \text{cut}(\mathcal{A}, \mathcal{V})$$

$$\text{Ncut}(\mathcal{V}_1, \dots, \mathcal{V}_k) = \sum_{i=0}^k \frac{\text{cut}(\mathcal{V}_i, \mathcal{V} - \mathcal{V}_i)}{a(\mathcal{V}_i)}$$

Quest'ultima quantità viene detta *k-NCut*

Connessioni con l'algoritmo NCut (Shi-Malik)

Definizione

$$\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} w_{ij}$$

$$a(\mathcal{A}) = \text{cut}(\mathcal{A}, \mathcal{V})$$

$$\text{Ncut}(\mathcal{V}_1, \dots, \mathcal{V}_k) = \sum_{i=0}^k \frac{\text{cut}(\mathcal{V}_i, \mathcal{V} - \mathcal{V}_i)}{a(\mathcal{V}_i)}$$

Quest'ultima quantità viene detta *k-NCut*

Connessioni con l'algoritmo NCut (Shi-Malik)

Shi e Malik hanno dimostrato che minimizzare il 2-NCut per ottenere una bipartizione del grafo è equivalente a minimizzare il seguente quoziente di Rayleigh:

$$\min_y \frac{y^T (D - W)y}{y^T D y}$$

A sua volta questo valore può essere approssimato calcolando l'autovettore relativo al secondo autovalore più piccolo della matrice $I - D^{-1}W$. Ricordando che il nostro processo converge ad una combinazione lineare di autovettori di $D^{-1}W$, osserviamo che gli autovettori delle due matrici coincidono.

Connessioni con l'algoritmo NCut (Shi-Malik)

Shi e Malik hanno dimostrato che minimizzare il 2-NCut per ottenere una bipartizione del grafo è equivalente a minimizzare il seguente quoziente di Rayleigh:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T (D - W) \mathbf{y}}{\mathbf{y}^T D \mathbf{y}}$$

A sua volta questo valore può essere approssimato calcolando l'autovettore relativo al secondo autovalore più piccolo della matrice $I - D^{-1}W$. Ricordando che il nostro processo converge ad una combinazione lineare di autovettori di $D^{-1}W$, osserviamo che gli autovettori delle due matrici coincidono.

Connessioni con l'algoritmo NCut (Shi-Malik)

Shi e Malik hanno dimostrato che minimizzare il 2-NCut per ottenere una bipartizione del grafo è equivalente a minimizzare il seguente quoziente di Rayleigh:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T (D - W) \mathbf{y}}{\mathbf{y}^T D \mathbf{y}}$$

A sua volta questo valore può essere approssimato calcolando l'autovettore relativo al secondo autovalore più piccolo della matrice $I - D^{-1}W$. Ricordando che il nostro processo converge ad una combinazione lineare di autovettori di $D^{-1}W$, osserviamo che gli autovettori delle due matrici coincidono.

Connessioni con l'algoritmo NCut (Shi-Malik)

Shi e Malik hanno dimostrato che minimizzare il 2-NCut per ottenere una bipartizione del grafo è equivalente a minimizzare il seguente quoziente di Rayleigh:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T (D - W) \mathbf{y}}{\mathbf{y}^T D \mathbf{y}}$$

A sua volta questo valore può essere approssimato calcolando l'autovettore relativo al secondo autovalore più piccolo della matrice $I - D^{-1}W$. Ricordando che il nostro processo converge ad una combinazione lineare di autovettori di $D^{-1}W$, osserviamo che gli autovettori delle due matrici coincidono.