Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Matematica

Alcuni metodi matriciali per lo Spectral Clustering

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.
VALERIA SIMONCINI

Presentata da:
SERENA MAROTTA

II Sessione
Anno Accademico 2016-2017

A chi conosce sorrisi e lacrime
di questi tre anni,
ma soprattutto
al Mio Papà
che con i suoi sacrifici ha contribuito alla
realizzazione del Mio Sogno
e alla Mia Mamma
colonna portante della mia vita ...

Indice

Introduzione

Introduzione al clustering

Il **clustering** è un processo di raggruppamento di elementi omogenei, rispetto a determinate caratteristiche, in un insieme di dati. Questa operazione, che letta così potrebbe sembrare molto astratta, nella vita di tutti i giorni ha un'infinità di applicazioni e viene messa in pratica inconsciamente ogni volta che si realizza un qualche raggruppamento: divisione fra maschi e femmine, raggruppamento di capoluoghi per regione, e così via. Però potrebbero esserci, in una popolazione o in un insieme dei dati, dei sottogruppi non così facilmente deducibili o oppure, a seconda della situazione che si sta analizzando, i raggruppamenti potrebbero variare: ad esempio le carte di un mazzo francese si raggruppano per seme se si gioca a Bridge, ma per valore se si gioca a Ramino. Poichè i risultati sono influenzati sia dall'obiettivo dell'indagine che dal contesto applicativo, occorre effettuare delle scelte che individuino la procedura più adatta.
Dunque, dato un insieme di dati, descritti da un insieme di attributi, dopo aver scelto **distanza** e **coefficienti di similarità** (descritti nella sezione "Distanze e misure di similarità"), trovare un insieme di **cluster**, vuol dire trovare dei raggruppamenti in cui oggetti appartenenti allo stesso cluster sono **simili** tra loro, ma **dissimili** da oggetti appartenenti a cluster differenti.

Il clustering dei dati è una disciplina scientifica giovane che sta attraversando un enorme sviluppo e oltre ad essere un'importante attività umana, è molto usata:

- in economia: può aiutare gli operatori a scoprire gruppi distinti di clienti caratterizzandoli in base ai loro acquisti;
- in biologia: può essere utilizzato per derivare le tassonomie delle piante e degli animali, per categorizzare i geni con funzionalità simili e per esaminare varie caratteristiche delle popolazioni;
- nell'identificazione di aree terrestri con uso simile in un database spaziale;
- nell'identificazione di gruppi di case in una città a seconda del tipo di casa, del suo valore e della sua locazione geografica;
- nella classificazione di documenti sul web;
- nel cercare di decifrare una cattiva calligrafia confrontando tra loro le lettere;
- come funzionalità del **data mining**, il clustering può essere utilizzato per esaminare le distribuzioni dei dati, per osservare le caratteristiche di ciascuna distribuzione e per focalizzarsi su quelle di maggiore interesse.

Distanze e coefficienti di similarità

Per $x, y \in \mathbb{R}^p$, abbiamo le seguenti misure di distanza:

- **Distanza euclidea**: $d(x,y) = \sqrt{(x-y)^T(x-y)}$;
- **Distanza statistica**: $d(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$;
- **Distanza di Minkowski**: $d_m(x,y) = (\sum_{i=1}^{p} |x_i - y_i|^m)^{\frac{1}{m}}$, con $m \in \mathbb{N}$;
- **distanza cityblock**: $d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$;
- **distanza di correlazione**: $d(x,y) = 1 - r(x,y)$, dove r indica la correlazione tra x e y;

e inoltre, denotando con:
a: frequenza di 1-1, **b**: frequenza di 1-0, **c**: frequenza di 0-1, **d**: frequenza di 0-0, utilizzando variabili binarie, abbiamo i seguenti coefficienti di similarità:

- $s_1(P,Q)=\frac{a}{p}$; ci dice che P e Q sono simili quando hanno entrambi 1;
- $s_2(P,Q)=\frac{a+d}{p}$; ci dice che P e Q sono simili solo quando hanno entrambi lo stesso peso;
- $s_3(P,Q)=\frac{a}{a+b+c}$; dà zero peso al termine con 0-0 ed è detto coefficiente di Jacard.

che si ricavano dalla seguente **tabella di contingenza**, dove p è il numero di variabili osservate:

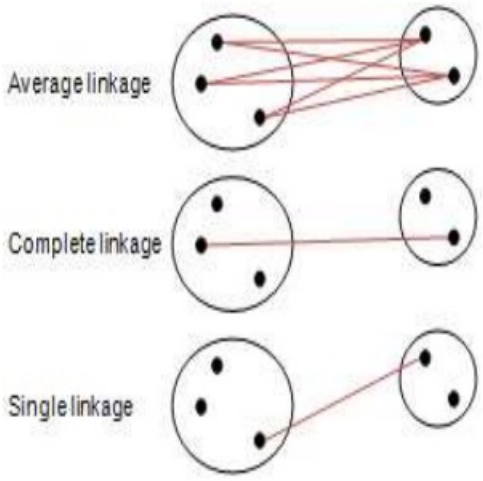
	1	0	Tot
1	a	b	$a+b$
0	c	d	$c+d$
Tot	$a+c$	$b+d$	$a+b+c+d=p$

Complete Linkage, K-means e Spectral Clustering

Tra i vari metodi di clustering, troviamo:

1. **Metodi di connessione**, come il Linkage che è un metodo gerarchico adatto per raggruppare sia variabili che osservazioni.

   - **Single linkage**, basati sulla minima distanza (vedremo nel Capitolo 4 con degli esempi, che questo metodo non è molto efficace, perchè non crea dei gruppi netti ma unisce pian piano tutti i cluster);
   - **Complete linkage**, basati sulla massima distanza;
   - **Average linkage**, basati sulla distanza media.



La tipica **procedura in un metodo gerarchico agglomerativo** è la seguente:

   ° Inizia con n gruppi ed una matrice $n \times n$ simmetrica di distanze (o similarità) D;

- Determina la coppia di elementi u e v più **vicini** (guardando la matrice D);
- Forma il gruppo (UV);
- Aggiorna D sostituendo alle due righe di U e V una sola riga della distanza del gruppo (YV) dagli altri oggetti. D sarà quindi $(n-1)\times(n-1)$. (Questo step individua il metodo di tipo gerarchico).
- Si ripetono i passi 2 e 4 per $n-1$ volte.

2. **Metodo delle k-medie**, che è un metodo non gerarchico e procede così:

   - Suddivide gli oggetti in k clusters, dove il k è dato in input, e calcola il centroide di ogni cluster (che può essere il primo elemento di ogni gruppo, o la media, o altro...)
   - Presi n oggetti, per ogni oggetto calcola la distanza dal centroide di ogni classe, ma senza usare matrice di distanza, quindi risulta utile per grandi moli di dati.
   - Riposiziona l'oggetto nel cluster con centroide più vicino;
   - Ricalcola poi il centroide del cluster che ha ricevuto il nuovo oggetto e per quello che ha perso l'oggetto;
   - Tutto questo finchè nessun oggetto cambia più cluster.

   Per verificare la stabilità del risultato, come vedremo meglio nel Capitolo 4, bisognerà far "girare" più volte l'algoritmo, cambiando l'inizializzazione, ovvero il parametro k oppure il tipo di distanza usata.

3. **Spectral Clustering**, argomento centrale della mia tesi, che è la teoria più diffusa negli ultimi anni. I motivi sono vari, innanzitutto la semplicità dell'implementazione, per la quale è sufficiente una libreria di

algebra lineare, e in secondo luogo l'ottenimento di risultati sbalorditivi che superano molte difficoltà che sembravano insormontabili. La semplicità è solo apparente, la teoria sottostante è la Teoria dei Grafi, una materia vasta e profonda che coinvolge aree scientifiche completamente differenti. La visione che esporrò sui grafi è quindi solo una piccola finestra per capire il funzionamento di questa tecnica, dettagliando diverse varianti, spiegandone differenze e similarità. L'esposizione, come per i diversi metodi, avverrà a livelli successivi di dettaglio: da una versione intuitiva a dimostrazioni di risultati (in particolare nel Capitolo 4).

Rappresentazione grafica: il dendrogramma

Una rappresentazione grafica del processo di clustering è fornita dal **dendrogramma** (vedi Figura 1), che gode di alcune proprietà:

- Il livello a cui avviene il raggruppamento è importante perchè evidenzia l'effettiva distanza;
- Se D ha minimi uguali con indici diversi, si raggruppano i clusters separatamente;
- Se D ha minimi uguali con indici in comune, si raggruppano solo gli oggetti con la stessa distanza;
- I clusters rimangono inalterati se si usano distanze che mantengono lo stesso rdine.

Il nostro interesse si focalizzerà sui raggruppamenti intermedi.

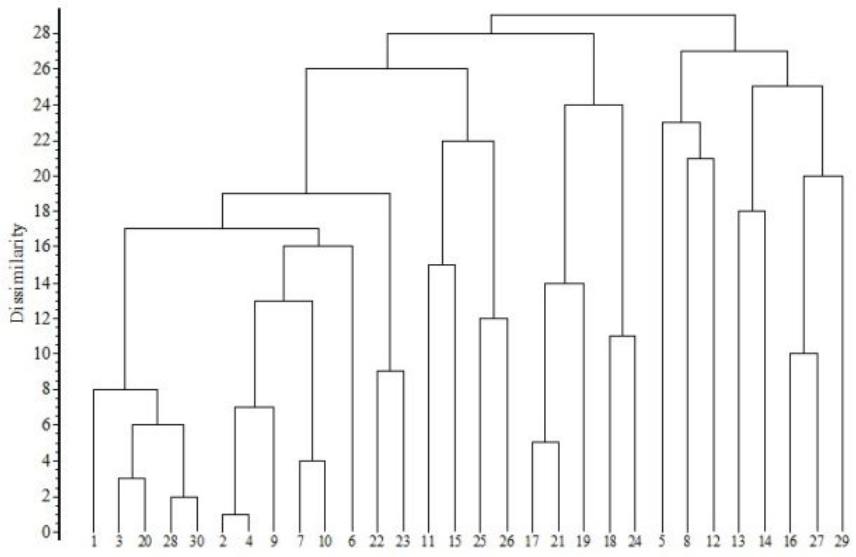


Figura 1: dendrogramma

Notazioni

- Preso un sottinsieme di vertici $A \subset V$, denoteremo con $\overline{A}$ il suo **complementare**, cioè $A \backslash V$;

- Definiremo il vettore indicatore $\mathbb{I}_A=(f_1, \cdots, f_n)^T \in \mathbb{R}^n$ con:

$$f_i = \begin{cases} 1 & se \ \ v_i \in A \\ 0 & se \ \ v_i \notin A \end{cases} \tag{1}$$

  Per convenienza però, introdurremo la notazione sintetica $i \in A$ per indicare l'insieme di indici $\{i | v_i \in A\}$, in particolare quando si tratta di una somma $\sum_{i \in A} w_{ij}$.

Capitolo 1

I Grafi

1.1 Richiami ed approfondimenti sui grafi

Consideriamo un insieme di punti $x_1, ..., x_n$ e i coefficienti di similarità $s_{ij} \geq 0$ tra tutte le coppie di punti x_i e x_j, lo scopo intuitivo del cluster è dividere i punti in diversi gruppi contenenti ognuno caratteristiche simili fra loro e dissimili da altri.
Se non abbiamo informazioni sulle similarità tra i punti, consideriamo il **grafo della similarità** $G = (V, E)$, dove ogni vertice v_i rappresenta un punto x_i.
Due vertici sono connessi se la similarità tra i punti è positiva o supera una certa soglia e il loro lato è pesato da s_{ij}.
Dunque il problema del clustering può essere riformulato usando questo grafo: vogliamo trovare una partizione del grafo, distinguendo lati in gruppi diversi con pesi bassi e lati all'interno di uno stesso gruppo con peso elevato. Sia $G = (V, E)$ un grafo non diretto, cioè simmetrico e non orientato, composto da un insieme finito di vertici $V = \{v_1, \cdots, v_n\}$ e da un insieme di archi $E \subset V \times V$, che connettono coppie di nodi. Due nodi connessi da un arco sono detti adiacenti.

Dati V ed E, sia $w\vdots E \longmapsto \mathbb{R}^+$ una funzione che associa un peso ad un arco, allora il grafo $G = (V, E, w)$ si dice **grafo pesato**.
$W = (w_{ij})$ con $i, j = 1, \cdots, n$ è detta **matrice di adiacenza**.
Chiaramente, $w_{ij} = 0$ se v_i non è connesso con v_j.
Ma W è matrice simmetrica, cioè $w_{ij} = w_{ji}$ e questo è importante perchè tutte le quantità spettrali saranno reali.

Il **grado** di un vertice $v_i \in V$ è definito come $d_i = \sum_{j=1}^{n} w_{ij}$,
indica il numero di nodi ad esso adiacente e corrisponde alla somma dei pesi della riga i-esima della matrice di adiacenza W.

$D = diag(d_1, \cdots, d_n)$ è detta **matrice dei gradi**.

Un **taglio** è una partizione del grafo in due insiemi disgiunti, ha un valore chiamato **cut** ed è definito come la somma delle discrepanze, cioè tutto ciò che disturba fuori dal gruppo e che andremo appunto a minimizzare, perchè più diventano piccoli e più ci avviciniamo ad una componente connessa.

$$cut(A_1, \cdots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i})$$

.

Un sottinsieme $A \subset V$ si dice **connesso** se ogni coppia di vertici di A è collegata da un percorso tutto di vertici in A, cioè riesco a muovermi ovunque all'interno, ma non è negata la possibilità di uscire.

$A \subset V$ si dice invece **componente connessa** se A è connesso e non ci sono connessioni con il complementare di A, quindi in questo caso non possono esserci collegamenti con l'esterno.

$\{A_1, \cdots, A_k\}$ forma una **partizione** di un grafo se $A_i \cap A_j = \emptyset \qquad \forall i \neq j$
e inoltre se $\bigcup_{i=1}^{k} A_i = V$.

1.2 Differenti grafi di similarità

Come abbiamo accennato prima, ci sono diversi modi per raggruppare le variabili $x_1, \cdots, x_n$, in modo che gli oggetti nello stesso cluster mostrino un alto grado di similarità e gli oggetti in cluster differenti invece un alto grado di dissimilarità. Per questo motivo possiamo trovare diversi grafi:

* **Grafo delle ε-vicinanze**: connettiamo i punti la cui distanza è più piccola di un certo ε. Poiché le distanze tra tutti i punti collegati sono approssimativamente della stessa scala, ponderando i lati non vengono più incluse le informazioni sui dati sul grafico. In tal caso, la connessione vale 1, altrimenti 0.

* **Grafo dei k-vicini più prossimi**: connettiamo ogni vertice solo con i propri k primi vicini. Questo tipo di relazione porta ad un grafo diretto, visto che non è simmetrica. Ci sono due modi per rendere questo grafo non diretto:

  ◦ **grafo dei k più vicini**: ignoriamo le direzioni dei lati e connettiamo v_i e v_j con un lato non diretto se v_i è tra i più vicini di v_j o viceversa;

  ◦ **grafo dei reciproci k più vicini**: connettiamo i vertici sia se v_i è tra i k più vicini di v_j, che se v_j è tra i k più vicini di v_i.

* **Grafo completamente connesso**: connettiamo semplicemente tutti i punti con similarità positiva e pesiamo tutti i lati con s_{ij}. Poiché il grafo dovrebbe rappresentare le relazioni locali di vicinanza, questa costruzione è utile solo se la funzione di similarità modella le vicinanze locali. Un esempio è la funzione Gaussiana di similarità, dove σ è un parametro fissato a priori che controlla l'ampiezza della vicinanza:

$$s(v_i, v_j) = exp(-\|(v_i, v_j)\|^2/(2\sigma^2))$$

Tutti questi grafici menzionati sono regolarmente usati nello spectral clustering. Per una discussione sul comportamento dei diversi grafi, si fa riferimento al Capitolo 3.

1.3 Matrici Laplaciane sui grafi

Il principio chiave delle tecniche di Spectral Clustering è considerare i dati come se fossero i vertici di un grafo e pesare le connessioni in base alla similarità tra due vertici. Questa interpretazione porta nel framework della "Teoria Spettrale dei Grafi", una teoria in cui i dati del training set possono essere considerati come l'approssimazione di uno spazio topologico (una varietà) le cui proprietà possono essere studiate attraverso le **proprietà spettrali** di una matrice chiamata **Laplaciano**. Queste proprietà, da cui il nome "Spectral", servono per caratterizzare i grafi, in modo da procedere a partizionamenti opportuni.

1.3.1 Matrici non normalizzate

La matrice Laplaciana **non normalizzata** è definita come $L = D - W$.

Proposizione 1.3.1. *La matrice L soddisfa le seguenti proprietà:*

1. $\forall v \in \mathbb{R}^n$ *si ha:*
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2;$$

2. *L è simmetrica e semidefinita positiva;*

3. *Il più piccolo autovalore di L è 0, il corrispondente autovettore è il vettore costante $\mathbb{I}$;*

4. *L ha n autovalori non negativi:* $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$.

Dimostrazione. 1. Dalla definizione di d_i,

$$f^T L f = f^T D f - f^T W f = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij} =$$

$$\frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

2. La simmetria di L segue dalla simmetria di W e D. Mentre il fatto che L è semidefinita positiva segue dal primo punto, perchè

$$f^T L f = \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 \geqslant 0, \forall f \in \mathbb{R}^n.$$

3.
$$L\underline{1} = D\underline{1} - W\underline{1} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} - d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} - \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \underline{0}. \qquad (1.1)$$

4. E' una conseguenza dei punti 1. e 3.

□

Proposizione 1.3.2. *Sia G un grafo non diretto con pesi non negativi. Allora:*

1. *La **molteplicità** k **dell'autovalore** 0 di L corrisponde al **numero di componenti connesse** $\{A_1, \cdots, A_k\}$ del grafo;*
   Nota: Se $k = 1$, non ci sono componenti connesse non banali, cioè il grafo è connesso.

2. *L'autospazio associato è generato dai **vettori indicatori** $\mathbb{I}_{A_i}$.*

Dimostrazione. 1. Per $k = 1$, il grafo è connesso. Sia x un autovettore associato all'autovalore 0, allora:

$$0 = f^T L f = \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

Poichè per ipotesi di connessione $w_{ij} > 0$ perchè abbiamo pesi non negativi, allora necessariamente $f_i = f_j$. Quindi gli elementi di f devono essere uguali se corrispondono a vertici connessi attraverso un percorso nel grafo, e quindi $f = 1 \equiv \mathbb{I}$.

2. Supponiamo ora che ci siano k componenti connesse. E' possibile ordinare i vertici in modo che quelli consecutivi appartengano alla stessa componente connessa. Cosicchè $W = blkdiag(W_1, \cdots, W_k)$, cioè siccome non c'è connessione, fuori dai blocchi è tutto zero, ovvero non c'è una strada che porti i vertici di un blocco nell'altro.
$L = blkdiag(L_1, \cdots, L_k)$ chiaramente è ancora a blocchi perchè da W tolgo D che è chiaramente a blocchi.
Ogni L_i è una matrice laplaciana di un sottografo, cioè della componente connessa, quindi L_i ha un autovalore 0 con autovettore $\underline{1}$ con dimensione la dimensione di L_i, cioè si ricade per ogni pezzetto i-esimo nel caso precedente, ovvero $\mathbb{I} = blkdiag(\underline{1}, \cdots, \underline{1})$.

□

1.3.2 Matrici normalizzate

A seconda di come scaliamo la matrice L, otteniamo due diverse matrici normalizzate, strettamente correlate tra loro:

$$L_{sym} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$$L_{nonsym} = D^{-1} L = I - D^{-1} W.$$

Nella prima, togliamo la matrice diagonale, moltiplicando a destra e a sinistra per la sua radice. Mentre nella seconda si moltiplica solo a sinistra, cioè vengono scalate le righe, per quello è non simmetrica.

Proposizione 1.3.3 (Proprietà di L_{sym} e di L_{nonsym})**.**

1. *$\forall f \in \mathbb{R}^n$ si ha:*
$$f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 ;$$

2. *λ è un autovalore di L_{nonsym} con autovettore $u \iff \lambda$ è un autovalore di L_{sym} con autovettore $w = D^{\frac{1}{2}} u$;*

3. *λ è un autovalore di L_{nonsym} con autovettore $u \iff \lambda$ e u risolvono il* ***problema generalizzato agli autovettori*** *$Lu = \lambda Du$;*

4. *0 è un autovalore di L_{nonsym} con autovettore costante $\mathbf{1}$; 0 un autovalore di L_{sym} con autovettore $D^{\frac{1}{2}}\mathbf{1}$;*

5. *L_{sym} e L_{nonsym} sono matrici semidefinite positive e hanno n autovalori reali non negativi $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$.*

Dimostrazione. 1. Si prova come la 1. della Proposizione (1.3.1);

2. Poichè $L_{nonsym} = D^{-1}L$,
(λ, u) sua autocoppia, vuol dire che vale
$$L_{nonsym} = \lambda u \iff D^{-1}Lu = \lambda u \iff Lu = \lambda Du \iff Lu = \lambda D^{\frac{1}{2}} D^{\frac{1}{2}} u$$
e ora moltiplicando entrambi i membri per $D^{-\frac{1}{2}}$, ottengo
$$D^{-\frac{1}{2}} Lu = \lambda D^{\frac{1}{2}} u$$
che è come dire
$$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} D^{\frac{1}{2}} u = \lambda D^{\frac{1}{2}} u$$
ovvero
$$L_{sym} D^{\frac{1}{2}} u = \lambda D^{\frac{1}{2}} u$$
e chiamando $D^{\frac{1}{2}} u := w$, otteniamo $(\lambda, D^{\frac{1}{2}} u)$ è autocoppia di L_{sym}.

3. Segue banalmente dalla definizione di autocoppia e di L_{nonsym}.

4. La prima affermazione segue da $L_{nonsym}\mathbf{1} = 0$, mentre la seconda segue dalla 2.;

5. L'affermazione su L_{sym} segue da 1., mentre quella su L_{nonsym} dalla 2.

□

Come nel caso della matrice non normalizzata, la molteplicità dell'autovalore 0 è collegata al numero di componenti connesse. La dimostrazione è analoga.

1.4 Algoritmo di Spectral Clustering

Indichiamo ora i più comuni algoritmi di cluster, distinguendo grafo non normalizzato e grafo normalizzato, in cui in particolare analizziamo due diversi modi di normalizzare.
Supponiamo che il nostro data set consista di un insieme di punti $x_1, \cdots, x_n$, corrispondenti ad oggetti arbitrari. Misuriamo le loro similarità a coppie, attraverso i coefficienti di similarità $s_{ij} = s(x_i, x_j)$ e definiamo così la matrice di similarità $S = (s_{ij})_{i,j=1,\cdots,n}$.

◦ Spectral clustering non normalizzato

INPUT: $S \in \mathbb{R}^{n*n}$ matrice di similarità e k che è il numero di cluster da costruire.

* Costruiamo un grafo di similarità con matrice di adiacenza W;
* Calcoliamo $L = D - W$ (non normalizzata);
* Troviamo i primi k autovettori $u_1, \cdots, u_k$ di L;
* Prendiamo $U \in \mathbb{R}^{n \times k}$ come la matrice contenente i vettori $u_1, \cdots, u_k$ come colonne;
* Per $i = 1, \cdots, n$, sia $y_i \in \mathbb{R}^k$ il vettore corrispondente all'i-esima riga di U;
* Raggruppiamo $(y_i)_{i=1,\cdots,n}$ in $\mathbb{R}^k$ con l'algoritmo delle k-medie nei cluster $C_1, \cdots, C_k$.

OUTPUT: **Clusters** $A_1, \cdots, A_k$ con $A_i = \{j | y_j \in C_i\}$.

◦ Spectral clustering normalizzato in accordo con Shi e Malik (2000)

INPUT: $S \in \mathbb{R}^{n \times n}$ matrice di similarità e k che è il numero di cluster da costruire.

* Costruiamo un grafo di similarità con matrice di adiacenza W;
* Calcoliamo $L = D - W$ (non normalizzata);
* Troviamo i primi k autovettori $u_1, \cdots, u_k$ del **problema generalizzato agli autovettori** $Lu = \lambda Du$;
* Prendiamo $U \in \mathbb{R}^{n \times k}$ come la matrice contenente i vettori $u_1, \cdots, u_k$ come colonne;
* Per $i = 1, \cdots, n$, sia $y_i \in \mathbb{R}^k$ il vettore corrispondente all'i-esima riga di U;
* Raggruppiamo $(y_i)_{i=1,\cdots,n}$ in $\mathbb{R}^k$ con l'algoritmo delle k-medie nei cluster $C_1, \cdots, C_k$.

OUTPUT: **Clusters** $A_1, \cdots, A_k$ con $A_i = \{j | y_j \in C_i\}$.

Notiamo che questo algoritmo usa gli autovettori generalizzati di L, che in accordo con la Proposizione (1.3.3) corrispondono agli autovettori della matrice L_{nonsym}.

L'algoritmo che segue invece usa gli autovettori della matrice L_{sym} e come vedremo, verrà introdotto un ulteriore passo di normalizzazione delle righe, che non è necessario nell'algoritmo precedente. Il perchè sarà spiegato nel Capitolo 2.

◦ Spectral clustering normalizzato in accordo con Ng, Jordan e Weiss (2002)

INPUT: $S \in \mathbb{R}^{n\times n}$ matrice di similarità e k che è il numero di cluster da costruire.

* Costruiamo un grafo di similarità con matrice di adiacenza W;
* Calcoliamo $L = D - W$(non normalizzata);
* Troviamo i primi k autovettori $u_1, \cdots, u_k$ di L_{sym} ;
* Prendiamo $U \in \mathbb{R}^{n\times k}$ come la matrice contenente i vettori $u_1, \cdots, u_k$ come colonne;
* Creiamo la matrice $T \in \mathbb{R}^{n\times k}$ da U, normalizzando le righe con la norma 1, cioè $t_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{\frac{1}{2}}$;
* Per $i = 1, \cdots, n$, sia $y_i \in \mathbb{R}^k$ il vettore corrispondente all'i-esima riga di T;
* Raggruppiamo $(y_i)_{i=1,\cdots,n}$ in $\mathbb{R}^k$ con l'algoritmo delle k-medie nei cluster $C_1, \cdots, C_k$.

OUTPUT: **Clusters** $A_1, \cdots, A_k$ con $A_i = \{j | y_j \in C_i\}$.

Vedremo nelle sezioni successive che le varie modifiche portano ad una rappresentazione migliore.

1.5 Partizionamento dei grafi

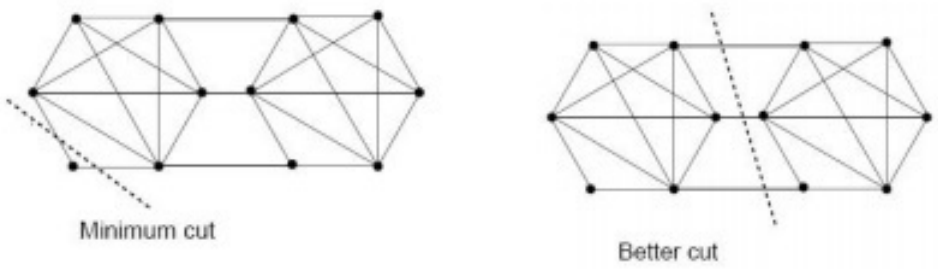
Considerato il training set come se fosse un grafo pesato G, lo **scopo** dello Spectral Clustering è quello di tagliarlo nel miglior modo possibile, secondo le similarità dei suoi elementi, cioè di trovare una partizione del grafo, distinguendo i gruppi con peso molto basso (il che vuol dire che i punti sono differenti l'uno dall'altro) e i gruppi con peso molto alto (il che vuol dire che i punti all'interno dello stesso cluster sono simili tra loro).

Preso un grafo di similarità con matrice di adiacenza W, il modo più semplice e diretto per creare una partizione è risolvere il problema di minimizzazione.

Ricordiamo che un **taglio** è una partizione del grafo in due insiemi disgiunti, ha un valore chiamato **cut** ed è definito come la somma delle discrepanze, cioè tutto ciò che disturba fuori dal gruppo e che andremo appunto a minimizzare, perchè più diventano piccoli e più ci avviciniamo ad una componente connessa.

$$cut(A_1, \cdots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i})$$

Introduciamo il fattore $\frac{1}{2}$ altrimenti avremmo contato ogni lato due volte. In particolare per $k = 2$, questo problema è relativamente semplice, ma porta con sè anche degli svantaggi, in quanto conduce a scarsi risultati perchè spesso taglia via solo un vertice.



Un modo per risolvere questo problema è richiedere che gli insiemi $(A_1, \cdots, A_k)$ siano **ragionevolmente grandi**.

* PROBLEMA DI OTTIMO PARTIZIONAMENTO

Definiamo allora le due funzioni più comuni da minimizzare:

$$RatioCut(A_1, \cdots, A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \overline{A_i})}{|A_i|} = \sum_{i=1}^{k}\frac{cut(A_i, \overline{A_i})}{|A_i|}$$

$$Ncut(A_1, \cdots, A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W(A_i, \overline{A_i})}{vol(A_i)} = \sum_{i=1}^{k}\frac{cut(A_i, \overline{A_i})}{vol(A_i)}$$

dove:
$|A_i|$ è il numero di vertici in A;

$vol(A_i) := \sum_{i \in A} d_i$ misura la dimensione di A sommando i pesi di tutti i lati collegati ai vertici in A.

Con questa formulazione non è più possibile tagliare via un solo vertice, altrimenti i volumi non sono massimizzati; la presenza del denominatore fa evitare di prendere sottografi con pochi vertici, ed avere così una partizione bilanciata.

Osservazione 1. Ncut minimizza anche la similarità all'interno di ogni cluster.

Posto in questi termini il problema è più semplice, ma la sua risoluzione con un algoritmo presenta delle difficoltà perchè se si pensa di generare tutte le partizioni possibili, si ha di fronte una complessità esponenziale $O(2^n)$. Si dice allora che il problema è **NP-hard**, da **"nondetermistic polynomial-time hard problem"**, ovvero "problema difficile non deterministico in tempo polinomiale" ma in tempo esponenziale e per questo motivo non siamo in grado di trovare una soluzione in tempi ragionevoli. Cercheremo allora

delle **condizioni semplificative** che ci porteranno proprio allo **spectral clustering**.

1.5.1 Approssimazione di RatioCut con $k = 2$

Cominciamo con il caso di Ratiocut per $k = 2$, in quanto il rilassamento è più facile da capire in questo ambito. Il nostro obiettivo è quello di risolvere il problema di ottimizzazione

$$\min_{A \subset V} RatioCut(A, \overline{A}).$$

Riscriviamo il problema in una forma più conveniente. Prendiamo il sottinsieme $A \subset V$ e definiamo il vettore $f = (f_1, \cdots, f_n)^T \in \mathbb{R}^n$ in questo modo:

$$f_i = \begin{cases} \sqrt{\frac{|\overline{A}|}{|A|}} & se \ \ v_i \in A \\ -\sqrt{\frac{|A|}{|\overline{A}|}} & se \ \ v_i \in \overline{A} \end{cases} \tag{1.2}$$

Ora la funzione *RatioCut* può essere riscritta utilizzando il grafo normalizzato laplaciano, che sarà più facile da minimizzare essendo un problema di algebra lineare e non un funzionale. Vediamo che:

$$\begin{aligned} f^T L f &= \frac{1}{2} \sum_{i,j=1}^{n} \omega_{ij} (f_i - f_j)^2 = \\ &= \frac{1}{2} \sum_{i \in A, j \in \overline{A}} \omega_{ij} \left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}} \right)^2 + \frac{1}{2} \sum_{i \in \overline{A}, j \in A} \omega_{ij} \left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}} \right)^2 = \\ &= cut(A, \overline{A}) \left(\frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} + 2 \right) = \\ &= cut(A, \overline{A}) \left(\frac{|A| + |\overline{A}|}{|A|} + \frac{|A| + |\overline{A}|}{|\overline{A}|} \right) = \\ &= |V| \cdot RatioCut(A, \overline{A}). \end{aligned} \tag{1.3}$$

Inoltre, abbiamo:

$$\sum_{i=1}^{n} f_i = \sum_{i\in A}\sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i\in\overline{A}}\sqrt{\frac{|A|}{|\overline{A}|}} = |A|\sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}|\sqrt{\frac{|A|}{|\overline{A}|}} = 0,$$

cioè, il vettore f, come definito nell'equazione (1.2) è ortogonale al vettore costante $\mathbf{1}$ e soddisfa:

$$||f||^2 = \sum_{i=1}^{n} f_i^2 = |A|\frac{|\overline{A}|}{|A|} + |\overline{A}|\frac{|A|}{|\overline{A}|} = |\overline{A}| + |A| = n.$$

Complessivamente, possiamo vedere che il problema di minimizzazione può essere riscritto come:

$$\min_{A\subset V} f^T L f, \qquad \text{con } f\perp\mathbf{1}\ , \qquad ||f|| = \sqrt{n}$$

Questo è un problema di ottimizzazione discreta poichè le componenti di f_i, definite nell'eq (1.2), sono autorizzate a prevedere solo due particolari valori e quindi è certamente ancora **NP hard**.
Il rilassamento più evidente in questa impostazione è scartare la condizione discreta e permettere invece che f_i assuma valori arbitrari in $\mathbb{R}$.
Questo porta al **problema di ottimizzazione rilassato**.

$$\min_{f\in\mathbf{R}^n} f^T L f, \qquad \text{con } f\perp\mathbf{1}, \qquad ||f|| = \sqrt{n}$$

Per il **teorema di Rayleigh-Ritz**, si può vedere immediatamente che la soluzione di questo problema è dato dal vettore f, che è l'autovettore corrispondente al **secondo più piccolo autovalore di** L. In questo modo possiamo approssimare un minimizzante di RatioCut dal secondo autovettore di L.
Tuttavia, al fine di ottenere una partizione del grafo, abbiamo bisogno di ritrasformare i valori reali di f del problema rilassato in un indicatore vettore discreto.

Il modo più semplice per farlo è quello di utilizzare il segno di f come **funzione indicatrice**, cioè scegliere:

$$\begin{cases} v_i \in A & se \;\; f_i \geq 0 \\ v_i \in \overline{A} & se \;\; f_i < 0 \end{cases} \tag{1.4}$$

Tuttavia, nel caso particolare di $k > 2$ che tratteremo in seguito, questa euristica è troppo semplice.
Ciò che la maggior parte degli algoritmi di clustering spettrale fa invece è quello di **considerare le coordinate di f_i come punti in $\mathbb{R}$ e raggruppare in due gruppi** C e $\bar{C}$ attraverso l'algoritmo di k-means. Scegliendo:

$$\begin{cases} v_i \in A & se \;\; f_i \in C \\ v_i \in \overline{A} & se \;\; f_i \in \bar{C} \end{cases} \tag{1.5}$$

Questo è esattamente **l'algoritmo di spectral clustering non normalizzato** per il caso $k = 2$.

1.5.2 Approssimazione di RatioCut con k arbitrario

Nel caso di k arbitrario, la procedura segue un principio simile a quello sopra descritto. Data una partizione di V in k insiemi $A_1, \cdots, A_k$, definiamo k vettori indicatori $h_j = (h_{1,j}, \cdots, h_{n,j})^T$ in questo modo:

$$h_{i,j} \begin{cases} \frac{1}{\sqrt{|A_j|}} & se \;\; v_i \in A_j \\ \\ 0 & altrimenti \end{cases} \qquad (i = 1, \cdots, n; j = 1, \cdots, k) \tag{1.6}$$

Poi costruiamo la matrice $H \in \mathbb{R}^{n\times k}$ come la matrice contenente questi k vettori indicatori come colonne. Osserviamo che le colonne di H sono **ortonormali** tra loro, cioè $H^T H = I$, e con dei calcoli simili a quelli svolti prima, si vede che:

$$h_i^T L h_i = \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

e inoltre si può verificare che:

$$h_i^T L h_i = (H^T L H)_{ii}$$

e la combinazione di questi fatti porta a:

$$RatioCut(A_1, \cdots, A_k) = \sum_{i=1}^{k} h_i^T L h_i = \sum_{i=1}^{k} (H^T L H)_{ii} = Tr(H^T L H),$$

dove Tr denota la **traccia** della matrice, cioè la somma degli elementi sulla diagonale. Così il problema di minimizzazione di $RatioCut(A_1, \cdots, A_k)$ può essere riscritto come:

$$\min_{A_1,\cdots,A_k} Tr(H^T L H) \quad \text{con } H^T H = I, \quad H \text{ come in (1.6)}$$

Rilassiamo nuovamente il problema, consentendo agli elementi della matrice H di assumere valori reali, e così il problema diventa:

$$\min_{H\in\mathbb{R}^{n\times k}} Tr(H^T L H) \qquad \text{con } H^T H = I.$$

Avendo effettuato molte semplificazioni, la soluzione del problema finale non è quella del problema iniziale, ma è **accettabile**.
La soluzione di cui parliamo si ottiene prendendo $H = X_{:,1:k}$, ovvero come la matrice contenente i primi k autovettori, cioè relativi ai più piccoli k autovalori di L come colonne. L'uso successivo dell'algoritmo delle k-medie permette di recuperare l'indicatore di clusters.
Questo porta all'**algoritmo generale di spectral clustering non normalizzato**, presentato nella Sezione 1.4.

1.5.3 Approssimazione di Ncut

Tecniche molto simili a quelle usate per *RatioCut* possono essere usate per minimizzare $Ncut$.

Nel caso $k = 2$ definiamo:

$$f_i \begin{cases} \sqrt{\frac{vol(\overline{A})}{vol(A)}} & se \;\; v_i \in A \\ -\sqrt{\frac{vol(A)}{vol(\overline{A})}} & se \;\; v_i \in \overline{A} \end{cases} \tag{1.7}$$

Si può verificare che $(Df)^T\mathbf{1} = 0, f^TDf = vol(V), f^TLf = vol(V)Ncut(A, \overline{A})$. Così possiamo riscrivere il problema di minimizzazione $Ncut$ con il problema equivalente:

$$\min_A f^TLf \quad \text{con } f \text{ come in (1.7)}, \quad Df \perp \mathbf{1}, \;\; f^TDf = vol(V)$$

Di nuovo rilassiamo il problema permettendo ad f di prendere valori reali:

$$\min_{f \in \mathbb{R}^n} f^TLf \quad \text{con } Df \perp \mathbf{1}, \;\; f^TDf = vol(V)$$

Adesso sostituiamo $g := D^{\frac{1}{2}}f$ e il problema diventa:

$$\min_{g \in \mathbb{R}^n} g^TD^{-\frac{1}{2}}LD^{-\frac{1}{2}}g \quad \text{con } g \perp D^{\frac{1}{2}}\mathbf{1}, \quad ||g||^2 = vol(V)$$

Osserviamo che $D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = L_{sym}$, dove $D^{\frac{1}{2}}\mathbf{1}$ è il primo autovettore di L_{sym} e $vol(V)$ è una costante. Quindi il problema è ora nella forma del teorema di Rayleigh-Ritz e la sua soluzione g è data dal secondo autovettore di L_{sym}. Sostituendo $f = D^{-\frac{1}{2}}g$ e usando la Proposizione (1.3.3) vediamo che f è il secondo autovettore di L_{nonsym} o equivalentemente l'autovettore generalizzato di $Lu = \lambda Du$.

Nel caso $k > 2$, definiamo il vettore indicatore $h_j = (h_{1,j}, \cdots, h_{n,j})$ così:

$$h_{i,j} \begin{cases} \frac{1}{\sqrt{vol(A_j)}} & se\ \ v_i \in A_j \\ & (i = 1, \cdots, n; j = 1, \cdots, k). \\ 0 & altrimenti \end{cases} \tag{1.8}$$

Poi costruiamo la matrice H mettendo in colonna i k vettori indicatori.

Osserviamo che, come nel caso di *RatioCut*, le colonne di H sono **ortonormali** tra loro, cioè $H^T H = I$, e con dei calcoli simili a quelli svolti prima si vede che

$$h_i^T D h_i = 1, \quad h_i^T L h_i = \frac{cut(A_i, \overline{A_i})}{vol(A_i)}.$$

Così possiamo riscrivere il problema di minimizzazione $Ncut$ come:

$$\min_{A_1, \cdots, A_k} Tr(H^T L H), \quad \text{con } H^T D H = I, \quad H \text{ come in (1.8)}$$

Rilassando la condizione discreta e sostituendo $T = D^{\frac{1}{2}} H$, otteniamo il problema rilassato:

$$\min_{T \in \mathbb{R}^{n*k}} Tr(T^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} T), \qquad \text{con } T^T T = I.$$

Anche questo è un problema standard di **minimizzazione della traccia** che è risolto dalla matrice T contenente i primi k autovettori di L_{sym} come colonne. Risostituendo $H = D^{-\frac{1}{2}} T$ e usando la Proposizione (1.3.3) vediamo che la soluzione H consiste nei primi k autovettori della matrice L_{nonsym} o i primi k autovettori generalizzati di $Lu = \lambda D u$.

Questo porta all'**algoritmo di spectral clustering normalizzato** in accordo con Shi e Malik (2000).

1.5.4 Commenti sull'approccio del rilassamento

Ci sono diversi commenti che dovremmo fare sulla derivazione dello spectral clustering. La più importante è che **non vi è alcuna garanzia sulla qualità della soluzione del problema rilassato rispetto alla soluzione esatta.** Cioè, se $A_1, \cdots, A_k$ è la soluzione esatta della minimizzazione di $RatioCut$, mentre $B_1, \cdots, B_k$ è la soluzione ottenuta dall'algoritmo di spectral clustering non normalizzato, la differenza

$$RatioCut(B_1, \cdots, B_k) - RatioCut(A_1, \cdots, A_k)$$

può essere arbitrariamente grande.

Dunque come avevamo già accennato prima, il motivo per cui il rilassamento spettrale è così attraente, non è che esso porta a soluzioni particolarmente buone. Ma la sua popolarità è dovuta principalmente al fatto che si traduce in un problema standard di algebra lineare, semplice da risolvere.

Capitolo 2

Teoria della perturbazione

La teoria della perturbazione studia come autovalori e autovettori di una matrice A cambiano se aggiungiamo una **piccola perturbazione** H, cioè consideriamo la matrice $\widetilde{A} = A + H$. La maggiorparte dei teoremi sulle perturbazioni affermano che una certa distanza tra autovalori e autovettori di A e $\widetilde{A}$ è delimitata da una costante che di solito dipende da quale autovalore guardiamo e da quanto questo è separato dal resto dello spettro.

La giustificazione dello spectral clustering è quindi la seguente:

- Consideriamo prima **"il caso ideale"**, dove la similarità **intracluster** è esattamente 0. Abbiamo già visto nella Sezione 1.3 che allora i primi k autovettori di L o L_{nonsym} sono i vettori indicatori dei cluster. In questo caso i punti $y_i \in \mathbb{R}^k$ costruiti con l'algoritmo di spectral clustering hanno la forma $(0, \cdots, 0, 1, 0, \cdots, 0)^T$ dove la posizione dell'1 indica la componente connessa a cui il punto appartiene. In particolare, tutti i punti y_i che coincidono, appartengono alla stessa componente connessa. L'algoritmo di k-medie troverà banalmente la corretta partizione posizionando un punto centrale su ciascuno dei punti $(0, \cdots, 0, 1, 0, \cdots, 0)^T \in \mathbb{R}^k$.

- In un **"caso quasi ideale"**, dove troviamo dei gruppi distinti, ma la similarità **intracluster** non è esattamente 0, consideriamo la matrice

Laplaciana come la versione perturbata di quella del caso ideale. La teoria della perturbazione ci dice che gli autovettori saranno molto vicini ai vettori indicatori ideali. I punti y_i non possono completamente coincidere con $(0, \cdots, 0, 1, 0, \cdots, 0)^T$, ma sono coincidenti a meno di un piccolo termine di errore. Quindi, se le perturbazioni non sono troppo grandi, l'algoritmo delle k-medie è ancora in grado di separare i gruppi l'uno dall'altro.

2.1 Argomento formale della perturbazione

Ciò che sta alla base della perturbazione nello spectral clustering è il **teorema di Davis-Kahan** dalla teoria della perturbazione matriciale. Questo teorema misura la differenza tra gli autospazi di una matrice simmetrica perturbata.

Le distanze tra i sottospazi sono solitamente misurate usando gli **"angoli canonici"** (chiamati anche "angoli principali"). Per definirli, prendiamo ν_1 e ν_2, due sottospazi p-dimensionali di $\mathbb{R}^d$, e siano poi V_1 e V_2 due matrici contenenti colonne ortonormali a ν_1 e ν_2, rispettivamente. (Notiamo che gli angoli canonici possono essere definiti se ν_1 e ν_2 non hanno la stessa dimensione).

Allora i $\cos\phi_i$ degli angoli principali ϕ_i sono i valori singolari di $V_1^T V_2$. Per $p = 1$, coincidono con la definizione normale di angolo. La matrice $\sin\phi(V_1, V_2)$ invece sarà la matrice diagonale, avente sulla diagonale i seni degli angoli canonici. Vediamo più precisamente:

Teorema 2.1.1. *(Davis-Kahan)*
Siano $A, H \in \mathbb{R}^{n\times n}$ matrici simmetriche, e sia $\|\cdot\|$ la norma di Frobenius o la norma-2 per matrici. Sia $\widetilde{A} := A + H$ una perturbazione di A e sia $S_1 \subset \mathbb{R}$ un intervallo. Denotiamo con $\sigma_{S_1}(A)$ l'insieme di autovalori di A che sono contenuti in S_1 e con V_1 l'autospazio corrispondente a tutti questi autovalori (più formalmente, V_1 è l'immagine della proiezione spettrale indotta da

$\sigma_{S_1}(A)$. Denotiamo poi con $\sigma_{S_1}(\widetilde{A})$ e $\widetilde{V_1}$ le analoghe quantità per $\widetilde{A}$.
Definiamo poi la distanza tra S_1 e lo spettro di A, fuori da S_1 come:

$$\delta = \min\{|\lambda - s|; \lambda \text{ autovalore di } A,\ \lambda \notin S_1, s \in S_1\}.$$

Allora, la distanza $d(V_1, \widetilde{V_1}) := \|\sin\phi(V_1, \widetilde{V_1})\|$ tra i due sottospazi V_1 e $\widetilde{V_1}$ è limitata da:

$$d(V_1, \widetilde{V_1}) \leq \frac{\|H\|}{\delta}.$$

Vediamo un'interpretazione di questo risultato in termini del laplaciano non normalizzato (si lavora analogamente nel caso normalizzato).
La matrice A corrisponde al grafo Laplaciano L nel caso ideale, dove il grafo ha k componenti connesse, mentre la matrice $\widetilde{A}$ corrisponde al caso perturbato, dove a causa delle discrepanze, le k componenti nel grafo non sono più completamente disconnesse, ma connesse da qualche lato con peso leggero. Denotiamo il grafo Laplaciano corrispondente $\widetilde{L}$.
Per lo spectral clustering dobbiamo considerare i primi k autovalori e autovettori di $\widetilde{L}$. Denotiamo gli autovalori di L con $\lambda_1, \cdots, \lambda_n$, mentre quelli di $\widetilde{L}$ con $\widetilde{\lambda_1}, \cdots, \widetilde{\lambda_n}$. Scegliere ora l'intervallo S_1 è un punto **cruciale**, perchè deve esser tale da contenere gli autovalori di L e di $\widetilde{L}$. Tale scelta è **più facile** se $||H||$ è piccola e se $|\lambda_k - \lambda_{k+1}|$ è grande.
Se riusciamo a trovare tale insieme, allora il **teorema di Davis-Kahan** ci dice che gli autospazi corrispondenti ai primi k autovalori di L e $\widetilde{L}$, sono vicini l'uno con l'altro, in particolare di una distanza pari a $\|H\|/\delta$. Inoltre, mentre gli autovettori nel caso ideale hanno elementi costanti sulle componenti connesse, lo stesso sarà approssimativamente vero nel caso perturbato. Quanto "è accurato" dipende da $\|H\|$ e dalla distanza δ tra S_1 e il $(k+1)$-esimo autovettore di L.
Se l'insieme S_1 è stato scelto come l'intervallo $[0, \lambda_k]$, allora δ coincide con l'intervallo spettrale $|\lambda_{k+1} - \lambda_k|$. Possiamo vedere dal teorema che più è grande questo intervallo e più saranno vicini gli autovettori del caso ideale e del caso perturbato; di conseguenza, lo spectral clustering lavorerà meglio.

Di seguito vedremo che la dimensione di questo intervallo può anche essere utilizzata in un contesto differente come criterio di qualità per lo spectral clustering, vale a dire al momento di scegliere il numero k di clusters da costruire.
Se la perturbazione H è troppo grande o l'intervallo è troppo piccolo, potremmo non trovare un insieme S_1 opportuno. Dunque c'è bisogno di un compromesso: l'affermazione del teorema allora diventa più debole nel senso che o non confrontiamo gli autospazi corrispondenti ai primi k autovettori di L e di $\widetilde{L}$ o può accadere che δ sia troppo piccolo cosicché il limite sulla distanza tra $d(V_1, \widetilde{V}_1)$ cresca così tanto da diventare inutile.

2.2 Commenti sull'approccio alla perturbazione

L'approccio perturbativo necessita un po' di cautela in quanto si utilizzano argomenti di teoria della perturbazione per giustificare gli algoritmi di clustering basati sugli autovettori. In generale, qualsiasi matrice simmetrica diagonale a blocchi ammette base di autovettori che sono zero fuori dai blocchi individuali, mentre a valori reali all'interno. Per esempio, sulla base di questi argomenti, molti autori usano gli autovettori della matrice di similarità S o di adiacenza W per scoprire i cluster. Tuttavia, essendo questa matrice diagonale a blocchi, nel caso ideale di completa separazione dei cluster, può essere considerata come una condizione **necessaria** per un corretto uso di autovettori, ma **non sufficiente**. Almeno altre due proprietà devono essere soddisfatte:

1. Prima di tutto, è necessario che l'ordine degli autovalori e degli autovettori sia significativo. Nel caso del Laplaciano, questo è sempre vero, cioè come sappiamo già, ogni componente connessa possiede un autovettore che ha autovalore 0. Quindi il grafo ha k componenti connesse e prendiamo i primi k autovettori del Laplaciano. Allora sappiamo che

ha esattamente un autovettore per componente.
Tuttavia potrebbe non essere così nel caso di altre matrici come S o W. Per esempio, potrebbe essere che i due più grandi autovalori della matrice diagonale a blocchi S, vengano dallo stesso blocco. In tale situazione, prendiamo i primi k autovettori di S, alcuni blocchi saranno rappresentati più volte, mentre altri mancheranno completamente (a meno che non prendiamo le appropriate precauzioni). Questa è la ragione per cui non dovremmo utilizzare gli autovettori di S o di W per il clustering.

2. La seconda proprietà è che nel caso ideale, gli elementi degli autovettori sulle componenti dovrebbero essere **sufficientemente lontane da zero**. Assumiamo che l'autovettore sulla prima componente connessa abbia un'entrata $u_{1,i} > 0$ in posizione i. Nel caso ideale, il fatto che questo elemento sia $non - zero$, indica che il corrispondente punto i appartiene al primo cluster. Viceversa, se un punto i non appartiene al primo cluster, allora nel caso ideale dovrebbe essere $u_{1,i} = 0$.
Adesso consideriamo la stessa situazione, ma nel caso pertubato. L'autovettore perturbato $\widetilde{u}$ non avrà nessun elemento $non - zero$, ma se il rumore non è troppo grande, allora la teoria della perturbazione ci dice che le componenti $\widetilde{u_{1,i}}$ e $\widetilde{u_{1,j}}$, sono ancora **abbastanza vicine** ai valori originali $u_{1,i}$ e $u_{1,j}$, cioè prenderanno qualche piccolo valore, come ε_1 e ε_2. In pratica, se questi valori sono molto piccoli non è chiaro come interpretare la situazione: o crediamo che i piccoli elementi indichino che i punti non appartengono al primo cluster o che indichino già la classe di appartenenza e classifichino entrambi i punti al primo cluster.

Per entrambe le matrici, L e L_{nonsym}, gli autovettori nella situazione ideale sono i vettori indicatori, così il secondo problema non si verifica. Invece questo non è vero nel caso della matrice L_{sym}, in cui anche nel caso ideale, gli autovettori corrispondono a $D^{\frac{1}{2}}\mathbf{1}_{\mathbf{A_i}}$.
Se i gradi dei vertici differiscono molto, e in particolare se i vertici hanno gradi molto bassi, allora gli elementi degli autovettori sono molto piccoli. Per

contrastare il problema sopra descritto, entra in gioco la **normalizzazione delle righe**. Nel caso ideale, la matrice U nell'algoritmo ha esattamente un elemento $non-zero$ per riga, e dopo aver effettuato questa normalizzazione, troviamo la matrice T che è quindi costituita dai vettori indicatori. Si noti tuttavia che questo potrebbe comunque non funzionare correttamente nella pratica. Assumiamo di avere $\widetilde{u_{i,1}} = \varepsilon_1$ e $\widetilde{u_{i,2}} = \varepsilon_2$, se adesso normalizziamo la i-esima riga di U, ε_1 e ε_2 saranno moltiplicati del fattore $\frac{1}{\sqrt{\varepsilon_1^2+\varepsilon_2^2}}$ e diventeranno così piuttosto grandi.
Ora ci imbattiamo in un problema simile a quello appena descritto: entrambi i punti sono verosimilmente classificati nello stesso cluster, anche se appartengono a gruppi differenti. Questo fatto mostra che lo spectral clustering che usa la matrice L_{sym} può essere problematico se gli autovettori contengono elementi particolarmente piccoli. D'altronde si noti che, questi piccoli elementi negli autovettori si verificano solo se qualche vertice ha peso particolarmente basso. Si potrebbe obiettare che in tal caso il punto dato dovrebbe essere considerato un'anomalia e allora non è un vero problema in quale cluster il punto finirà.

Per riassumere, la conclusione è che lo spectral clustering normalizzato con L_{sym} può essere giustificato dalla teoria della perturbazione, ma dovrebbe essere trattato con molta attenzione se il grafo contiene vertici con pesi molto bassi.

Capitolo 3

Dettagli pratici

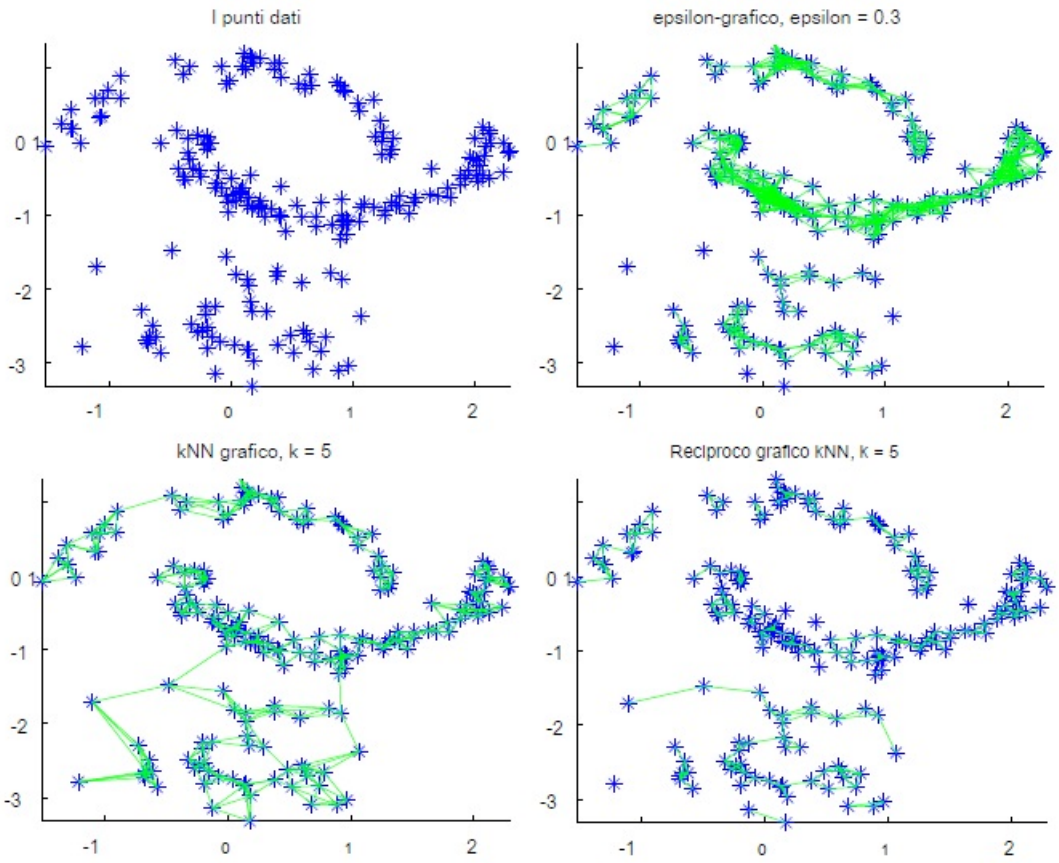
In questa sezione discuteremo brevemente di alcuni problemi che sorgono quando in realtà implementiamo lo spectral clustering. Ci sono molte scelte da fare e parametri da impostare.

3.1 Costruzione del grafo di similarità

Costruire il grafo di similarità per lo spectral clustering non è un compito banale e si sa poco sulla teoria delle varie costruzioni.

Prima di poter pensare a costruire il grafo di similarità, abbiamo bisogno di definire una **funzione di similarità** sui dati. Per esempio, quando costruiamo una funzione di similarità tra documenti, ha senso verificare che i documenti con alto punteggio di similarità appartengano allo stesso cluster. Il comportamento globale **"a lungo termine"** non è importante per lo spectral clustering perchè comunque non connettiamo questi due punti nel grafo di similarità. Nel caso comune, dove i punti vivono nello spazio Euclideo $\mathbb{R}^d$, un candidato ragionevole di default è la funzione di similarità Gaussiana $s(x_i, x_j) = exp(-\|x_i - x_j\|^2/(2\sigma^2))$, dove abbiamo bisogno di scegliere il parametro σ.

La prossima scelta riguarda il tipo di grafo che vogliamo usare; illustriamo brevemente il comportamento dei differenti grafi usati nel seguente semplice esempio presentato in figura:



Scegliamo una distribuzione in $\mathbb{R}^2$ con tre cluster: due "lune" e una Gaussiana. La densità della luna inferiore è scelta maggiore della luna superiore. Il grafico in alto a sinistra mostra un campione prelevato da questa distribuzione mentre gli altri tre riquadri mostrano differenti grafi di similarità su questo campione.

* Nel **grafo delle ε-vicinanze**, possiamo vedere che è difficile scegliere un utile parametro ε. Con ε=0.3 come in figura, i punti sulla luna cen-

trale sono già molto strettamente connessi, mentre i punti sulla Gaussiana lo sono leggermente. Il problema si verifica sempre se abbiamo i dati su "differenti scale", cioè distanze differenti in regioni differenti dello spazio.

* Nel **grafo dei k più vicini**, possiamo connettere i punti su "differenti scale": vediamo infatti che i punti sulla Gaussiana sono connessi con quelli sulla luna di densità elevata. Questa è una proprietà generale che può risultare molto utile.

* Il **grafo dei reciproci k più vicini** invece tende a connettere i punti all'interno di regioni a densità costanti, ma non connette regioni a densità differenti. Così questo può esser considerato come **"nel mezzo"** tra i due grafi sopra descritti.

* Il **grafo completamente connesso** è spesso usato nella connessione con la funzione di similarità Gaussiana. I punti vicini sono connessi con pesi alti, mentre quelli lontani con pesi trascurabili. Tuttavia, la risultante matrice di similarità, non è una matrice sparsa.

Come raccomandazione generale, si suggerisce di lavorare con il **grafo dei k più vicini** come prima scelta, perchè è più semplice: si traduce in una matrice di adiacenza W, sparsa, ed è meno vulnerabile alle scelte inadatte dei parametri, rispetto agli altri grafi.

Una volta deciso il tipo di grafo da usare, dobbiamo scegliere i parametri. Si deve fare in modo che tali componenti connesse siano i corretti cluster, ovvero che il grafo sia collegato, o che consista solo di "poche" componenti connesse rispetto al numero di cluster da rilevare e nessun vertice isolato. Ci sono molti risultati teorici su come connettere un grafo random, ma tutti questi valgono solo nel caso in cui si tende al limite per $n \to \infty$.
Per esempio, se si sa che per n punti **i.i.d** (=indipendenti identicamente distribuiti) a supporto connesso in $\mathbb{R}^d$, il grafo dei k più vicini e il grafo dei reciproci k più vicini saranno connessi se scegliamo k dell'ordine di $log(n)$.

Argomenti simili mostrano che il parametro ε nel grafo delle ε -vicinanze deve essere scelto come $(log(n)/n)^d$ per garantire la connettività al limite. Un altro modo semplice consiste nello sceglierlo come la lunghezza del lato più lungo in un albero di minima copertura completamente connesso sui punti dati.

3.2 Importanza degli autovettori

Per implementare lo spectral clustering si devono calcolare i primi k autovettori della matrice Laplaciana. Fortunatamente, se si usa il grafo dei k più vicini o delle ε-vicinanze, allora tutte queste matrici sono sparse ed esistono algoritmi efficienti per calcolare i primi k autovettori, quali il **metodo delle potenze** o il **metodo dei sottospazi di Krylov**, come per esempio il **metodo di Lanczos**. La velocità di convergenza di tali algoritmi dipende dalla dimensione di $\gamma_k = |\lambda_k - \lambda_{k+1}|$, cioè la differenza tra autovalori successivi.
Un problema generale si verifica se uno degli autovalori ha molteplicità maggiore di uno. Per esempio, nella situazione ideale di k cluster disconnessi, l'autovalore 0 ha molteplicità k. Come abbiamo visto, in questo caso, l'autospazio è dato dallo **span** dei k vettori indicatori. Ma sfortunatamente, i vettori calcolati dagli algoritmi numerici, non necessariamente convergono a questi particolari vettori, ma alle basi ortonormali degli autospazio e di solito questo dipende dai dettagli dell'implementazione. Questo non è però poi così male... si può vedere infatti che tutti i vettori nello spazio dato dallo span dei vettori indicatori $\mathbf{1}_{\mathbf{A_i}}$ sono della forma $u = \sum_{i=1}^{k} a_i \mathbf{1}_{\mathbf{A_i}}$, per qualche coefficiente a_i, cioè sono a valori costanti sui cluster e dunque i vettori restituiti dagli algoritmi possono essere usati per decodificare informazioni importanti sui cluster.

3.3 Il numero di clusters

Scegliere il numero k di cluster è un problema generale per tutti gli algoritmi di cluster e sono stati ideati numerosi metodi per questo problema: un metodo molto usato è quello di scegliere il numero k in modo che gli autovalori $\lambda_1, \cdots, \lambda_k$ siano molto piccoli, ma λ_{k+1} sia relativamente grande. Ci sono molte giustificazioni per questa procedura: la prima è basata sulla teoria della perturbazione, in cui osserviamo che nel caso ideale di k cluster completamente disconnessi, l'autovalore 0 ha molteplicità k e c'è un divario con il $(k+1)$-esimo autovalore $\lambda_{k+1} > 0$. Un'altra spiegazione può esser data dalla teoria spettrale dei grafi: in particolare, la dimensione dei tagli è strettamente correlata alla dimensione dei primi autovalori.
Nonostante questo però, il metodo migliore è quello di scegliere il numero di cluster in modo euristico, ovvero vedere per quale k i gruppi si stabilizzano, cioè non variano (troppo) al variare dell'inizializzazione e soprattutto, sono significativi.

3.4 Il metodo delle k-medie

I tre algoritmi di spectral clustering che abbiamo presentato nella Sezione 1.4 usano l'algoritmo delle k-medie nell'ultimo passaggio per estrarre la partizione finale dalla matrice di autovettori a valori reali e abbiamo visto dalle varie spiegazioni che questo passaggio è molto semplice se i dati contengono cluster ben evidenti. Per esempio, nel caso di cluster completamente separati, sappiamo che gli autovettori di L e di L_{nonsym} sono costanti a tratti e dunque tutti i punti x_i che appartengono allo stesso cluster C_s, saranno mappati esattamente al punto di campionamento y_i. La distanza euclidea tra i punti y_i è una quantità significativa da guardare, poichè è legata alla distanza sul grafo.

Ci sono però altre tecniche per costruire la soluzione finale: per esempio nei testi di **Lang** o di **Bach and Jordan**, dove gli autori utilizzano gli iper-

piani nel primo caso, mentre il sottospazio generato dai primi k autovettori e provano ad approssimarlo usando i vettori costanti a tratti nel secondo.

3.5 Quale grafo dev'essere usato?

Prima di poter rispondere a questa domanda, bisogna guardare il grado della distribuzione del grafo di similarità: se il grafo è molto regolare e molti vertici hanno approssimativamente lo stesso grado, allora tutti i laplaciani sono simili tra loro e funzionerà altrettanto bene per il clustering; se invece i gradi sono distribuiti a grandi linee, allora i laplaciani saranno notevolmente differenti.
Ci sono diverse discussioni che sostengono l'uso dello spectral clustering normalizzato piuttosto che il non normalizzato e inoltre che nel caso normalizzato bisogna usare gli autovettori di L_{nonsym} piuttosto che quelli di L_{sym}.

Come abbiamo già visto nelle sezioni precedenti, la prima giustificazione a favore dello spectral clustering normalizzato viene dal punto di vista del **partizionamento** del grafo. Per semplicità discutiamo del caso $k = 2$. In generale, il clustering ha due **obiettivi** differenti:

1. Trovare una partizione in modo che i punti in cluster differenti siano dissimili, cioè minimizzare la differenza fra i cluster, ovvero minimizzare $cut(A, \overline{A})$;

2. Trovare una partizione in modo che i punti nello stesso cluster siano simili, cioè massimizzare la differenza intra-cluster $W(A, A)$ e $W(\overline{A}, \overline{A})$, dove questi ultimi, come abbiamo visto nel Capitolo 1, indicano la somma delle discrepanze, cioè tutto ciò che disturba fuori dal gruppo e che andremo appunto a minimizzare, perchè più diventano piccoli e più ci avviciniamo ad una componente connessa.

$$cut(A_1, \cdots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i})$$

.

Sia *RatioCut* sia $Ncut$ implementano il primo obiettivo, inserendo esplicitamente $cut(A, \overline{A})$ nella funzione. Ma riguardo al secondo punto, le due funzioni agiscono diversamente.

Notiamo innanzitutto che

$$W(A, A) = W(A, V) - W(A, \overline{A}) = vol(A) - cut(A, \overline{A})$$

. Quindi, la similarità intra cluster è massimizzata se $cut(A, \overline{A})$ è piccolo e se $vol(A)$ è grande. Poichè questo è esattamente ciò che avviene minimizzando $Ncut$, allora $Ncut$ implementa tranquillamente anche il secondo obiettivo.

Questo è ancora più esplicito se consideriamo il criterio di **MinMaxCut** introdotto da Ding, He,Zha, Gu e Simon, nel 2001:

$$MinMaxCut(A_1, \cdots, A_k) := \sum_{i=1}^{k} \frac{cut(A_i, \overline{A_i})}{W(A_i, A_i)}$$

con denominatore differente rispetto a Ncut, nel quale compare $vol(A) = cut(A, \overline{A}) + W(A, A)$.
In pratica, Ncut e MinMaxCut sono spesso minimizzate da tagli così fatti, infatti rilassare la seconda equivale allo stesso problema di ottimizzazione ottenuto rilassando la prima, cioè alla normalizzazione dello spectral clustering con gli autovettori di L_{nonsym}.

Nel caso di *RatioCut* invece, l'obiettivo è massimizzare $|A|$ e $|\overline{A}|$ invece dei rispettivi volumi. Ma $|A|$ e $|\overline{A}|$ non sono necessariamente legati alla similarità intracluster, che dipende dai lati e non dal numero di vertici in A. Per esempio, basti pensare ad un insieme A con molti vertici, saranno tutti collegati tra loro da lati con pesi leggeri.
Quindi la minimizzazione di *RatioCut* **non** porta alla massimizzazione della similarità intracluster e allora il primo importante punto è tenere in mente che **lo spectral clustering normalizzato implementa entrambi gli**

obiettivi, mentre quello non normalizzato implementa solo il primo.

3.6 Problemi di coerenza

Una discussione completamente differente sulla superiorità dello spectral clustering normalizzato viene da un'analisi statistica su entrambi gli algoritmi di spectral clustering normalizzato descritti nella Sezione 1.4.
Assumiamo di avere i punti $x_1, \cdots, x_n$ i.i.d in accordo con la distribuzione di probabilità P e sottostanti lo spazio X. La domanda fondamentale riguarda il **problema di coerenza**: se tracciamo sempre più punti, i risultati di clustering convengono a una partizione utile sottostante lo spazio X?
Matematicamente si dimostra che, quando prendiamo il limite $n \to \infty$, la matrice L_{sym} converge in senso **forte** a un operatore U sullo spazio $C(X)$ delle funzioni continue su X. Questa convergenza implica che gli autovalori e gli autovettori di L_{sym} convergono a quelli di U e si può dimostrare che la partizione indotta su X dagli autovettori di U, può essere interpretata come la passeggiata random dello spectral clustering. Questo vuol dire che, se consideriamo un processo di diffusione sullo spazio dato X, allora la partizione indotta dagli autovettori di U è tale che la diffusione non transiti molto spesso nei diversi cluster.

In contrasto con i risultati chiari sulla convergenza dello spectral clustering normalizzato, la situazione per quello non normalizzato è molto più sgradevole. Si può dimostrare che non sempre converge, oppure che converge a soluzioni banali che portano alla creazione di cluster contenenti un solo punto o ancora che la matrice $(1\backslash n)L$ converge a qualche operatore limite T su $C(X)$ quando $n \to \infty$, il quale però ha delle proprietà così ostili da impedire la convergenza dello spectral clustering.

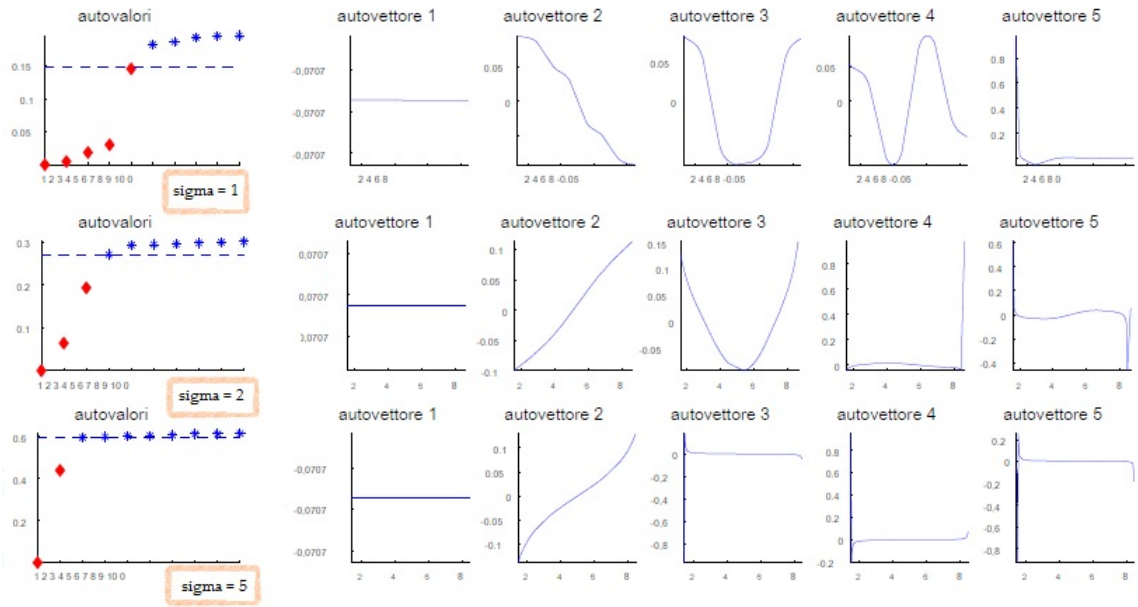
Vedremo tra poco con un semplice esempio che questo problema non riguarda solo le grandi dimensioni. Intanto però, è possibile caratterizzare le condizioni in cui questo problema non si verifica:

* Bisogna fare in modo che gli autovalori di L, corrispondenti agli autovettori usati nello spectral clustering non normalizzato, siano **significativamente inferiori al minimo grado del grafo**. Questo significa che se usiamo i primi k autovettori, allora deve essere

$$\lambda_i \ll \min_{j=1,\cdots,n} d_j$$

per tutti gli $i = 1, \cdots, k$.

La ragione è che gli autovettori relativi agli autovalori che non rispettano questa condizione **approssimano le funzioni di Dirac**, cioè sono *circa* 0 in tutte le direzioni tranne una. E allora se andiamo ad utilizzare questi autovettori nel clustering, l'unico vertice diverso da 0 si separerà da tutti gli altri e questo non è sicuramente quello che vogliamo.
Per un esempio di questo fenomeno, analizziamo il seguente semplice esempio, per scelte differenti del parametro σ.



La soglia $\min d_j$ è indicata da una linea tratteggiata; gli autovalori sopra questa soglia li rappresentiamo con stelle blu, mentre quelli sotto con diamanti rossi.

In generale possiamo vedere che gli autovettori corrispondenti agli autovalori che sono al di sotto della soglia tratteggiata sono **"utili"**.

- Nel caso $\sigma = 1$, gli autovalori $1, 2, 3$ e 4 sono significativamente sotto, e allora i rispettivi autovettori sono importanti;
- Osserviamo che, incrementando il parametro, per esempio $\sigma = 2$, gli autovalori tendono a muoversi verso la soglia;
- E infine nel caso $\sigma = 5$, solo due autovalori sono al di sotto.

Possiamo allora vedere che, non appena l'autovalore si avvicina o supera $\min d_j$, il corrispondente autovettore approssima una funzione di Dirac. Naturalmente, questi autovettori non sono efficaci per il clustering. Questi problemi riguardano solo gli autovettori della matrice L, non quelli di L_{sym} e L_{nonsym}, perchè queste matrici non vengono utilizzate negli algoritmi non normalizzati. Dunque, da un punto di vista statistico, è preferibile **evitare lo spectral clustering non normalizzato e usare invece gli algoritmi normalizzati**.

Guardando le differenze tra i due algoritmi di spectral clustering normalizzati nella Sezione 1.4, è preferibile invece il primo. La ragione è che il primo usa gli autovettori generalizzati di L, che in accordo con la Proposizione (1.3.3), corrispondono agli autovettori della matrice L_{nonsym}, dove nella situazione ideale gli autovettori corrispondono ai vettori indicatori; mentre il secondo algoritmo usa gli autovettori della matrice L_{sym} e come abbiamo visto nella Sezione 1.4, anche nella situazione ideale, gli autovettori corrispondono a $D^{\frac{1}{2}}\mathbf{1}_{A_i}$ e dunque se i vertici hanno gradi molto bassi, gli elementi degli autovettori sono molto piccoli. Per questa ragione si effettua un'altra

normalizzazione delle righe e si costruisce così T, contenente i vettori indicatori. Ciò nonostante, questo potrebbe non funzionare nella pratica, e questo è il motivo per il quale è preferito l'algoritmo con L_{nonsym}.

Capitolo 4

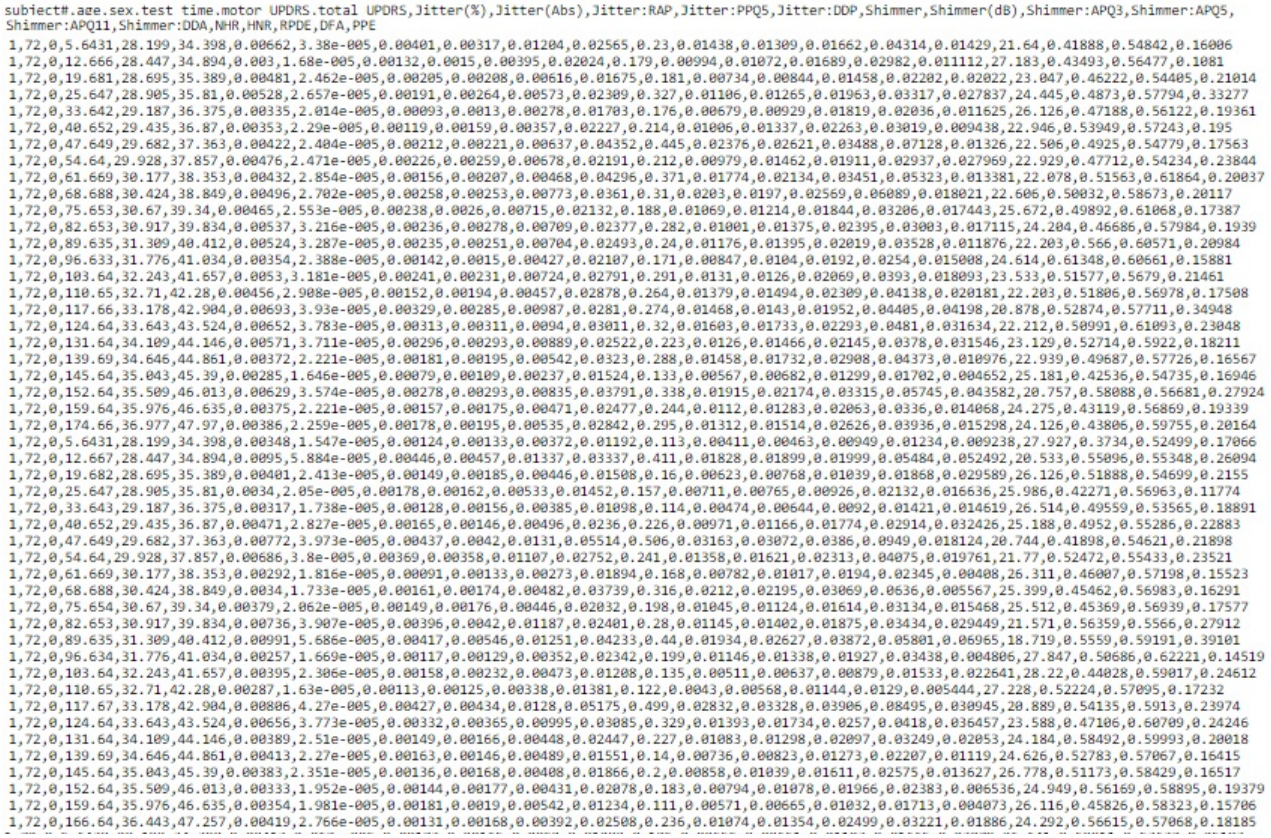
Problema reale: Analisi di un Data Set sul Parkinson

4.1 Descrizione del data set

Come abbiamo visto nell'Introduzione al clustering, il raggruppamento può esser visto come una procedura che cerca divisioni interne, plausibili, di un data set ritenuto troppo grande per esser trattato come unico.

In questo Capitolo riporto un'analisi del set di dati **"Parkinson Telemonitoring Data Set"**, creato da Athanasios Tsanas e Max Little, professori dell'Università di Oxford, in collaborazione con dieci centri medici degli Stati Uniti ed Intel Corporation che hanno sviluppato un dispositivo che registra suoni vocali. Questo data set è composto da una matrice $\mathbb{P} \in \mathbb{R}^{5875\times22}$, dove le righe indicano le **osservazioni**, cioè i 42 malati di Parkinson, di cui per ognuno sono state fatte circa 150 osservazioni in sei mesi, mentre le colonne rappresentano le **variabili**, ovvero informazioni sul paziente e misure biomediche. Il nostro **scopo** è cercare di capire come le variabili possono essere raggruppate in **cluster**, cioè quali variabili sono correlate tra loro, e quali invece, non hanno niente a che fare l'una con l'altra, nonostante nella pratica si possa pensare il contrario (come per esempio non è vero che un malato di

80 anni ha sintomi diversi da quelli di 50, se sono malati dallo stesso tempo).



```
subject#,age,sex,test_time,motor_UPDRS,total_UPDRS,Jitter(%),Jitter(Abs),Jitter:RAP,Jitter:PPQ5,Jitter:DDP,Shimmer,Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,
Shimmer:APQ11,Shimmer:DDA,NHR,HNR,RPDE,DFA,PPE
1,72,0,5.6431,28.199,34.398,0.00662,3.38e-005,0.00401,0.00317,0.01204,0.02565,0.23,0.01438,0.01309,0.01662,0.04314,0.01429,21.64,0.41888,0.54842,0.16006
1,72,0,12.666,28.447,34.894,0.003,1.68e-005,0.00132,0.0015,0.00395,0.02024,0.179,0.00994,0.01072,0.01689,0.02982,0.011112,27.183,0.43493,0.56477,0.1081
1,72,0,19.681,28.695,35.389,0.00481,2.462e-005,0.00205,0.00208,0.00616,0.01675,0.181,0.00734,0.00844,0.01458,0.02202,0.02022,23.047,0.46222,0.54405,0.21014
1,72,0,25.647,28.905,35.81,0.00528,2.657e-005,0.00191,0.00264,0.00573,0.02309,0.327,0.01106,0.01265,0.01963,0.03317,0.027837,24.445,0.4873,0.57794,0.33277
1,72,0,33.642,29.187,36.375,0.00335,2.014e-005,0.00093,0.0013,0.00278,0.01703,0.176,0.00679,0.00929,0.01819,0.02036,0.011625,26.126,0.47188,0.56122,0.19361
1,72,0,40.652,29.435,36.87,0.00353,2.29e-005,0.00119,0.00159,0.00357,0.02227,0.214,0.01006,0.01337,0.02263,0.03019,0.009438,22.946,0.53949,0.57243,0.195
1,72,0,47.649,29.682,37.363,0.00422,2.484e-005,0.00212,0.00221,0.00637,0.04352,0.445,0.02376,0.02621,0.03488,0.07128,0.01326,22.506,0.4925,0.54779,0.17563
1,72,0,54.64,29.928,37.857,0.00476,2.471e-005,0.00226,0.00259,0.00678,0.02191,0.212,0.00979,0.01462,0.01911,0.02937,0.027969,22.929,0.47712,0.54234,0.23844
1,72,0,61.669,30.177,38.353,0.00432,2.854e-005,0.00156,0.00207,0.00468,0.04296,0.371,0.01774,0.02134,0.03451,0.05323,0.013381,22.078,0.51563,0.61864,0.20037
1,72,0,68.688,30.424,38.849,0.00496,2.702e-005,0.00258,0.00253,0.00773,0.0361,0.31,0.0203,0.0197,0.02569,0.06089,0.018021,22.606,0.50032,0.58673,0.20117
1,72,0,75.653,30.67,39.34,0.00465,2.553e-005,0.00238,0.0026,0.00715,0.02132,0.188,0.01069,0.01214,0.01844,0.03206,0.017443,25.672,0.49892,0.61068,0.17387
1,72,0,82.653,30.917,39.834,0.00537,3.216e-005,0.00236,0.00278,0.00709,0.02377,0.282,0.01001,0.01375,0.02395,0.03003,0.017115,24.204,0.46686,0.57984,0.1939
1,72,0,89.635,31.309,40.412,0.00524,3.287e-005,0.00235,0.00251,0.00704,0.02493,0.24,0.01176,0.01395,0.02019,0.03528,0.011876,22.203,0.566,0.60571,0.20984
1,72,0,96.633,31.776,41.034,0.00354,2.388e-005,0.00142,0.0015,0.00427,0.02107,0.171,0.00847,0.0104,0.0192,0.0254,0.015008,24.614,0.61348,0.60661,0.15881
1,72,0,103.64,32.243,41.657,0.0053,3.181e-005,0.00241,0.00231,0.00724,0.02791,0.291,0.0131,0.0126,0.02069,0.0393,0.018093,23.533,0.51577,0.5679,0.21461
1,72,0,110.65,32.71,42.28,0.00456,2.908e-005,0.00152,0.00194,0.00457,0.02878,0.264,0.01379,0.01494,0.02309,0.04138,0.020181,22.203,0.51806,0.56978,0.17508
1,72,0,117.66,33.178,42.904,0.00693,3.93e-005,0.00329,0.00285,0.00987,0.0281,0.274,0.01468,0.0143,0.01952,0.04405,0.04198,20.878,0.52874,0.57711,0.34948
1,72,0,124.64,33.643,43.524,0.00652,3.783e-005,0.00313,0.00311,0.0094,0.03011,0.32,0.01603,0.01733,0.02293,0.0481,0.031634,22.212,0.50991,0.61093,0.23048
1,72,0,131.64,34.109,44.146,0.00571,3.711e-005,0.00296,0.00293,0.00889,0.02522,0.223,0.0126,0.01466,0.02145,0.0378,0.031546,23.129,0.52714,0.5922,0.18211
1,72,0,139.69,34.646,44.861,0.00372,2.221e-005,0.00181,0.00195,0.00542,0.0323,0.288,0.01458,0.01732,0.02908,0.04373,0.010976,22.939,0.49687,0.57726,0.16567
1,72,0,145.64,35.043,45.39,0.00285,1.646e-005,0.00079,0.00109,0.00237,0.01524,0.133,0.00567,0.00682,0.01299,0.01702,0.004652,25.181,0.42536,0.54735,0.16946
1,72,0,152.64,35.509,46.013,0.00629,3.574e-005,0.00278,0.00293,0.00835,0.03791,0.338,0.01915,0.02174,0.03315,0.05745,0.043582,20.757,0.58088,0.56681,0.27924
1,72,0,159.64,35.976,46.635,0.00375,2.221e-005,0.00157,0.00175,0.00471,0.02477,0.244,0.0112,0.01283,0.02063,0.0336,0.014068,24.275,0.43119,0.56869,0.19339
1,72,0,174.66,36.977,47.97,0.00386,2.259e-005,0.00178,0.00195,0.00535,0.02842,0.295,0.01312,0.01514,0.02626,0.03936,0.015298,24.126,0.43806,0.59755,0.20164
1,72,0,5.6431,28.199,34.398,0.00348,1.547e-005,0.00124,0.00133,0.00372,0.01192,0.113,0.00411,0.00463,0.00949,0.01234,0.009238,27.927,0.3734,0.52499,0.17066
1,72,0,12.667,28.447,34.894,0.0095,5.884e-005,0.00446,0.00457,0.01337,0.03337,0.411,0.01828,0.01899,0.01999,0.05484,0.052492,20.533,0.55096,0.55348,0.26094
1,72,0,19.682,28.695,35.389,0.00401,2.413e-005,0.00149,0.00185,0.00446,0.01508,0.16,0.00623,0.00768,0.01039,0.01868,0.029589,26.126,0.51888,0.54699,0.2155
1,72,0,25.647,28.905,35.81,0.0034,2.05e-005,0.00178,0.00162,0.00533,0.01452,0.157,0.00711,0.00765,0.00926,0.02132,0.016636,25.986,0.42271,0.56963,0.11774
1,72,0,33.643,29.187,36.375,0.00317,1.738e-005,0.00128,0.00156,0.00385,0.01098,0.114,0.00474,0.00644,0.0092,0.01421,0.014619,26.514,0.49559,0.53565,0.18891
1,72,0,40.652,29.435,36.87,0.00471,2.827e-005,0.00165,0.00146,0.00496,0.0236,0.226,0.00971,0.01166,0.01774,0.02914,0.032426,25.188,0.4952,0.55286,0.22883
1,72,0,47.649,29.682,37.363,0.00772,3.973e-005,0.00437,0.0042,0.0131,0.05514,0.506,0.03163,0.03072,0.0386,0.0949,0.018124,20.744,0.41898,0.54621,0.21898
1,72,0,54.64,29.928,37.857,0.00686,3.8e-005,0.00369,0.00358,0.01107,0.02752,0.241,0.01358,0.01621,0.02313,0.04075,0.019761,21.77,0.52472,0.55433,0.23521
1,72,0,61.669,30.177,38.353,0.00292,1.816e-005,0.00091,0.00133,0.00273,0.01894,0.168,0.00782,0.01017,0.0194,0.02345,0.00408,26.311,0.46007,0.57198,0.15523
1,72,0,68.688,30.424,38.849,0.0034,1.733e-005,0.00161,0.00174,0.00482,0.03739,0.316,0.0212,0.02195,0.03069,0.0636,0.005567,25.399,0.45462,0.56983,0.16291
1,72,0,75.654,30.67,39.34,0.00379,2.062e-005,0.00149,0.00176,0.00446,0.02032,0.198,0.01045,0.01124,0.01614,0.03134,0.015468,25.512,0.45369,0.56939,0.17577
1,72,0,82.653,30.917,39.834,0.00736,3.907e-005,0.00396,0.0042,0.01187,0.02401,0.28,0.01145,0.01402,0.01875,0.03434,0.029449,21.571,0.56359,0.5566,0.27912
1,72,0,89.635,31.309,40.412,0.00991,5.686e-005,0.00417,0.00546,0.01251,0.04233,0.44,0.01934,0.02627,0.03872,0.05801,0.06965,18.719,0.5559,0.59191,0.39101
1,72,0,96.634,31.776,41.034,0.00257,1.669e-005,0.00117,0.00129,0.00352,0.02342,0.199,0.01146,0.01338,0.01927,0.03438,0.004806,27.847,0.50686,0.62221,0.14519
1,72,0,103.64,32.243,41.657,0.00395,2.306e-005,0.00158,0.00232,0.00473,0.01208,0.135,0.00511,0.00637,0.00879,0.01533,0.022641,28.22,0.44028,0.59017,0.24612
1,72,0,110.65,32.71,42.28,0.00287,1.63e-005,0.00113,0.00125,0.00338,0.01381,0.122,0.0043,0.00568,0.01144,0.0129,0.005444,27.228,0.52224,0.57095,0.17232
1,72,0,117.67,33.178,42.904,0.00886,4.27e-005,0.00427,0.00434,0.0128,0.05175,0.499,0.02832,0.03328,0.03906,0.08495,0.030945,20.889,0.54135,0.5913,0.23974
1,72,0,124.64,33.643,43.524,0.00656,3.773e-005,0.00332,0.00365,0.00995,0.03085,0.329,0.01393,0.01734,0.0257,0.0418,0.036457,23.588,0.47106,0.60709,0.24246
1,72,0,131.64,34.109,44.146,0.00389,2.51e-005,0.00149,0.00166,0.00448,0.02447,0.227,0.01083,0.01298,0.02097,0.03249,0.02053,24.184,0.58492,0.59993,0.20018
1,72,0,139.69,34.646,44.861,0.00413,2.27e-005,0.00163,0.00146,0.00489,0.01551,0.14,0.00736,0.00823,0.01273,0.02207,0.01119,24.626,0.52783,0.57067,0.16415
1,72,0,145.64,35.043,45.39,0.00383,2.351e-005,0.00136,0.00168,0.00408,0.01866,0.2,0.00858,0.01039,0.01611,0.02575,0.013627,26.778,0.51173,0.58429,0.16517
1,72,0,152.64,35.509,46.013,0.00333,1.952e-005,0.00144,0.00177,0.00431,0.02078,0.183,0.00794,0.01078,0.01966,0.02383,0.005536,24.949,0.56169,0.58895,0.19379
1,72,0,159.64,35.976,46.635,0.00354,1.981e-005,0.00181,0.0019,0.00542,0.01234,0.111,0.00571,0.00665,0.01032,0.01713,0.004073,26.116,0.45826,0.58323,0.15706
1,72,0,166.64,36.443,47.257,0.00419,2.766e-005,0.00131,0.00168,0.00392,0.02508,0.236,0.01074,0.01354,0.02499,0.03221,0.01886,24.292,0.56615,0.57068,0.18185
```

. . . . . .

Figura 4.1: Prime righe di P

In Figura 4.1. si vede solo una piccola parte della matrice P e si può notare che nella prima colonna c'è l'1, che indica il riferimento al primo paziente, troveremo così, dopo circa 150 righe, il secondo e così via...ma questa variabile è irrilevante, per cui considereremo la matrice senza la prima colonna. Inoltre, poichè i numeri sono di natura differente (interi, razionali, binari...), lavoreremo sulla matrice P standardizzata.

Le 21 **variabili** analizzate sono le seguenti:

1. **Età**;
2. **Sesso** (0: maschio, 1:donna);
3. **Test time**: tempo trascorso (la parte intera indica il numero di giorni);
4. **Motor UPDRS**: Anomalie motorie come tremore, rigidità, bradycinesia, problemi di postura, mascheratura facciale, ecc.
5. **Total UPDRS**: Disturbi mentali, comportamento e umore, compromissione intellettuale, depressione;

Osservazione 2. Il **segnale vocale** è un suono complesso quasi periodico, presenta quindi variazioni del periodo fondamentale T e dell'ampiezza A, a breve e/o lungo termine. Le prossime variabili **Jimmer** e **Shimmer** effettuano una media delle differenze di durata o di ampiezza di periodi successivi adiacenti.

6. **Jitter(%)**;
7. **Jitter (Abs)**;
8. **Jitter: RAP**;
9. **Jitter: PPQ5**;
10. **Jitter: DDP**;
11. **Shimmer**;
12. **Shimmer(dB)**;
13. **Shimmer:APQ3**;

14. **Shimmer:APQ5**;
15. **Shimmer:APQ11**;
16. **Shimmer:DDA**;
17. **NHR**, e la successiva HNR, sono due misure di rapporto tra rumore e componenti tonali nella voce;
18. **HNR**;
19. **RPDE** è una misura di complessità dinamica non lineare;
20. **DFA** è un esponente di scala di frattale del segnale;
21. **PPE** è una misura non lineare della variazione di frequenza fondamentale.

4.2 Metodo di connessione: linkage

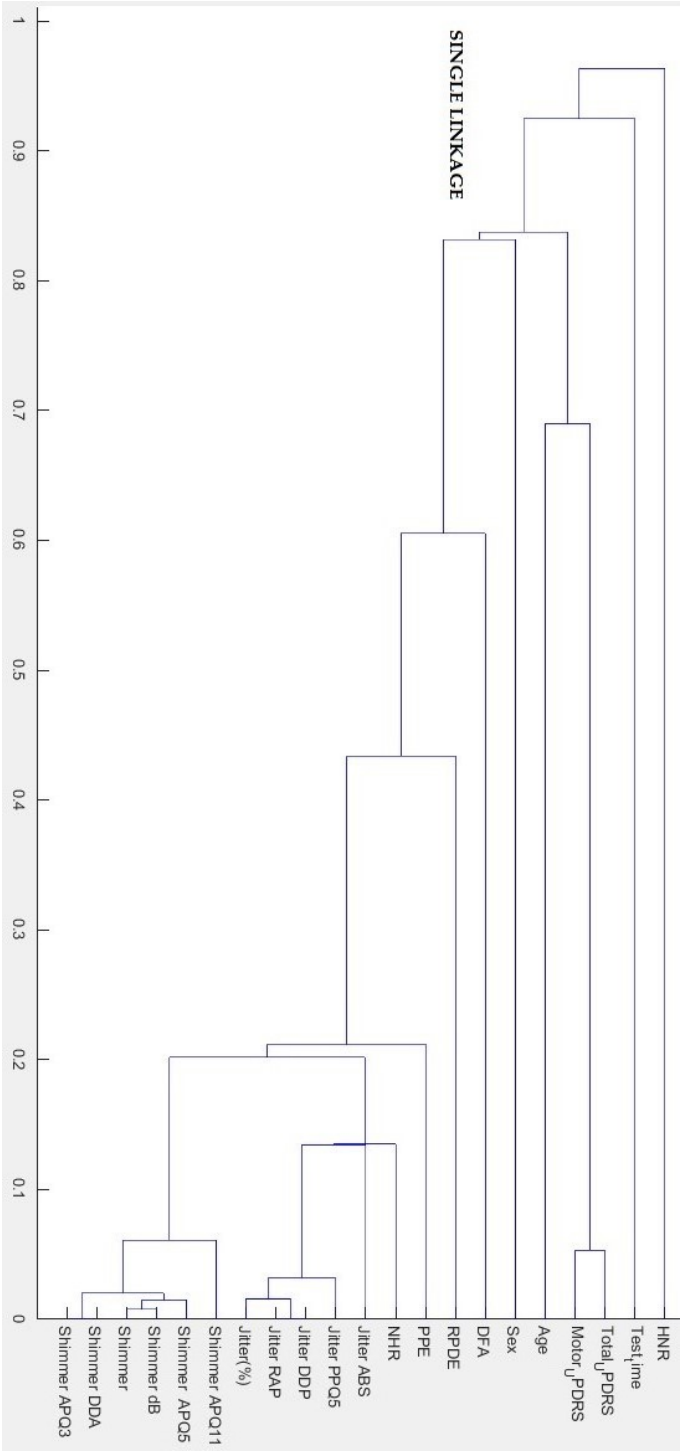
Come avevamo già accennato nell'Introduzione, il linkage è un metodo gerarchico adatto per raggruppare sia variabili che osservazioni. Tra i vari tipi distinguiamo **Single Linkage, Complete Linkage** e **Average Linkage**, ma in particolare ci occuperemo dei primi due, rispettivamente basati sulla minima e massima distanza.

A proposito di distanza invece, fissate le variabili $x, y \in \mathbb{R}^p$, utilizzeremo:

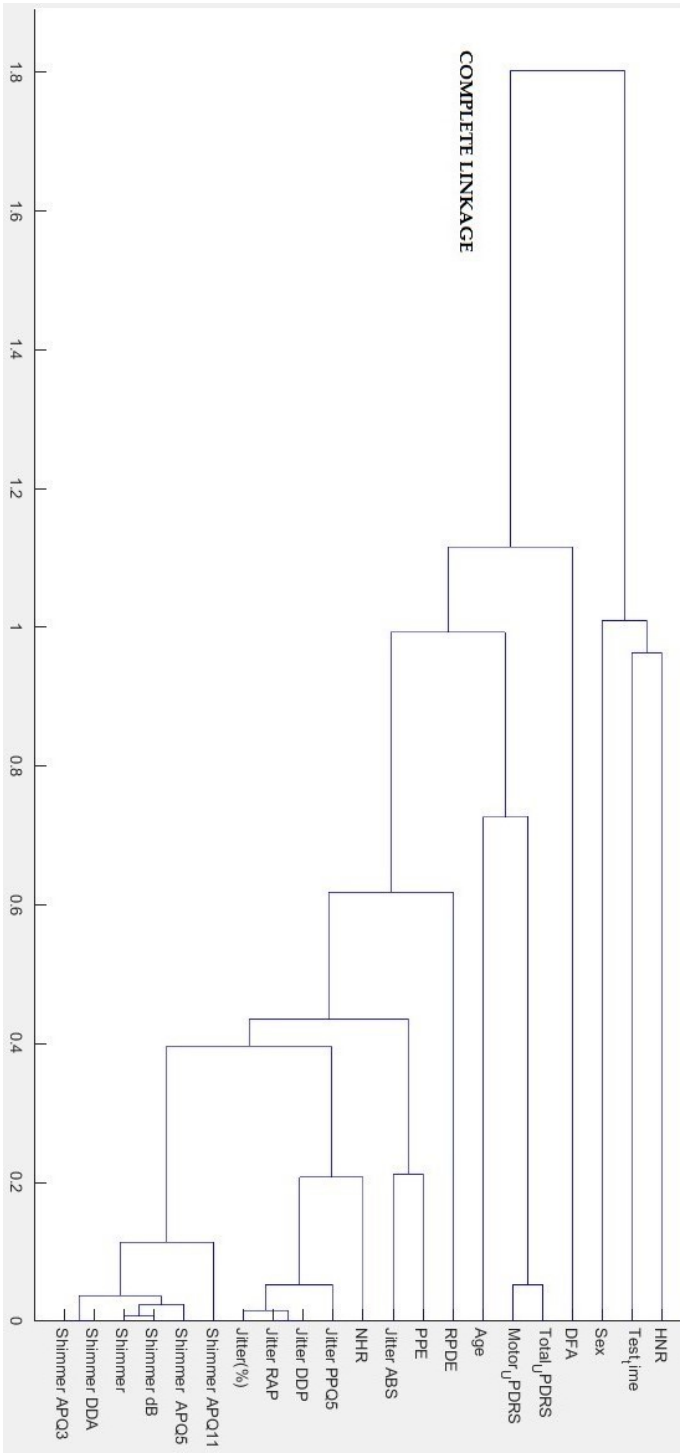
- la distanza **correlation**, $d(x, y) = 1 - r(x, y)$, dove r indica appunto la correlazione tra le variabili considerate;
- la distanza **euclidean**, $d(x, y) = \sqrt{(x-y)^T(x-y)}$

Cambiare il tipo di linkage o distanza nell'algoritmo, può condurre chiaramente a risultati differenti.

- Con la distanza **correlation**, otteniamo il seguente risultato, per quanto riguarda il single linkage:



- Mentre il seguente per quanto riguarda il complete linkage:



Il livello a cui i cluster si aggregano è molto significativo nei dendrogrammi, infatti vediamo che nel **Single Linkage** i primi cluster che si formano sono quelli degli esami Shimmer e Jimmer, e poco più su quelli formati dalle scale Motor e Total. Ma come avevamo già accennato, questo metodo non è molto efficace, poichè non crea dei gruppi netti e significativi ma unisce pian piano tutte le variabili tra di loro.

Il **Complete Linkage** invece ci offre dei cluster in più, come per esempio il legame con l'età e le scale Motor e Total, ma si vede dal livello, che non è un legame molto forte.

Prima di cambiare la distanza per vedere se i cluster cambiano e in caso positivo, come cambiano, facciamo un plot delle correlazione tra alcune variabili (vedi Figura 4.2) e notiamo che (dall'alto verso il basso):

- Shimmer APQ3 (13) e Shimmer DDA (16) sono fortemente correlati, infatti in entrambi i linkage troviamo il cluster;
- Età (1) e HNR (18) non sono per niente correlate, infatti otteniamo un grafico molto sparso;
- Jitter Rap (8) e Jitter (10) avevamo visto che creavano subito un cluster e infatti la correlazione lineare ne è la prova;
- Motor UPDRS (4) e Total UPDRS (5) sono correlati ma non fortemente, infatti si vede anche nei dendrogrammi che il cluster non si forma a livelli bassi.

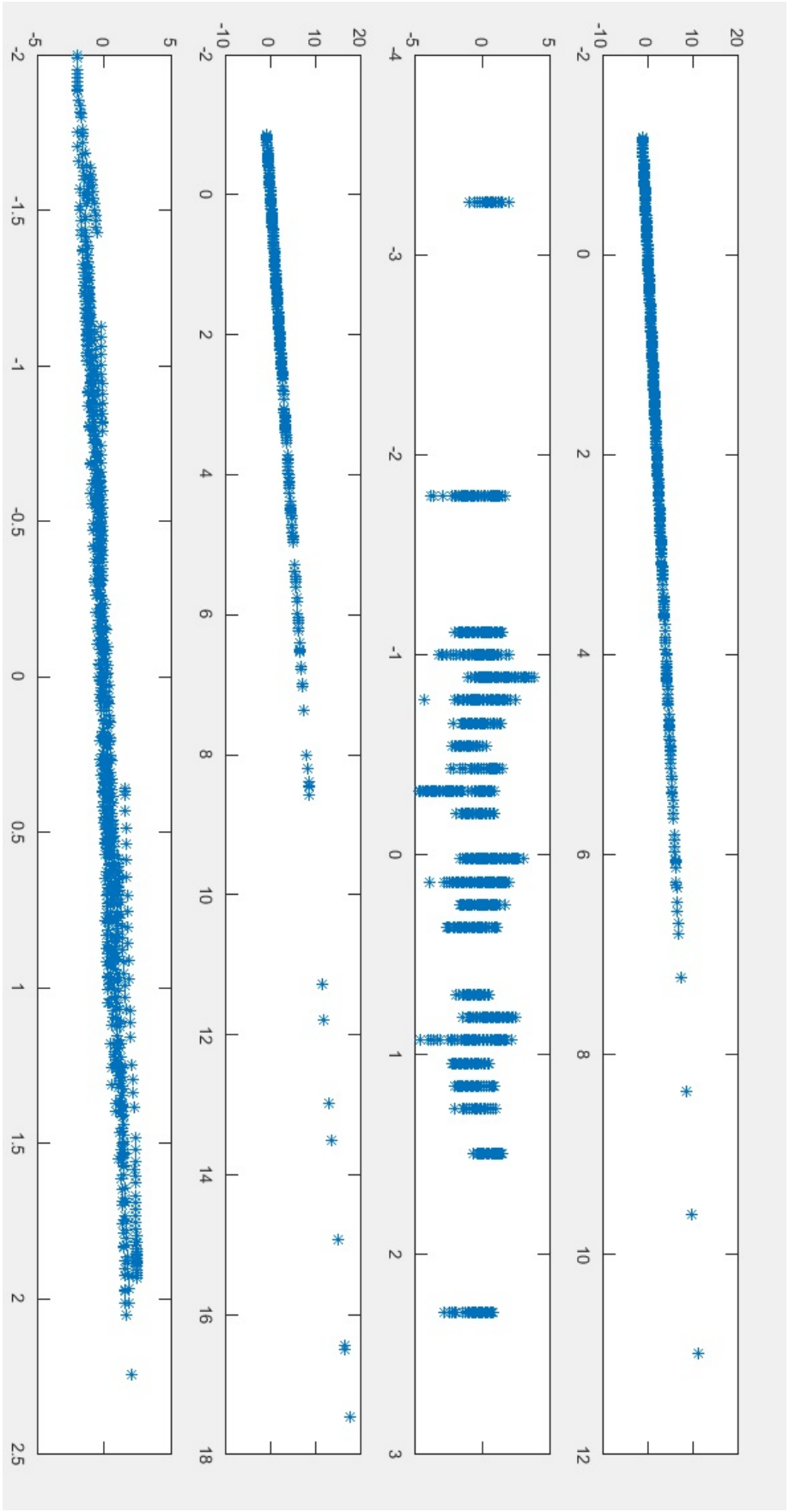
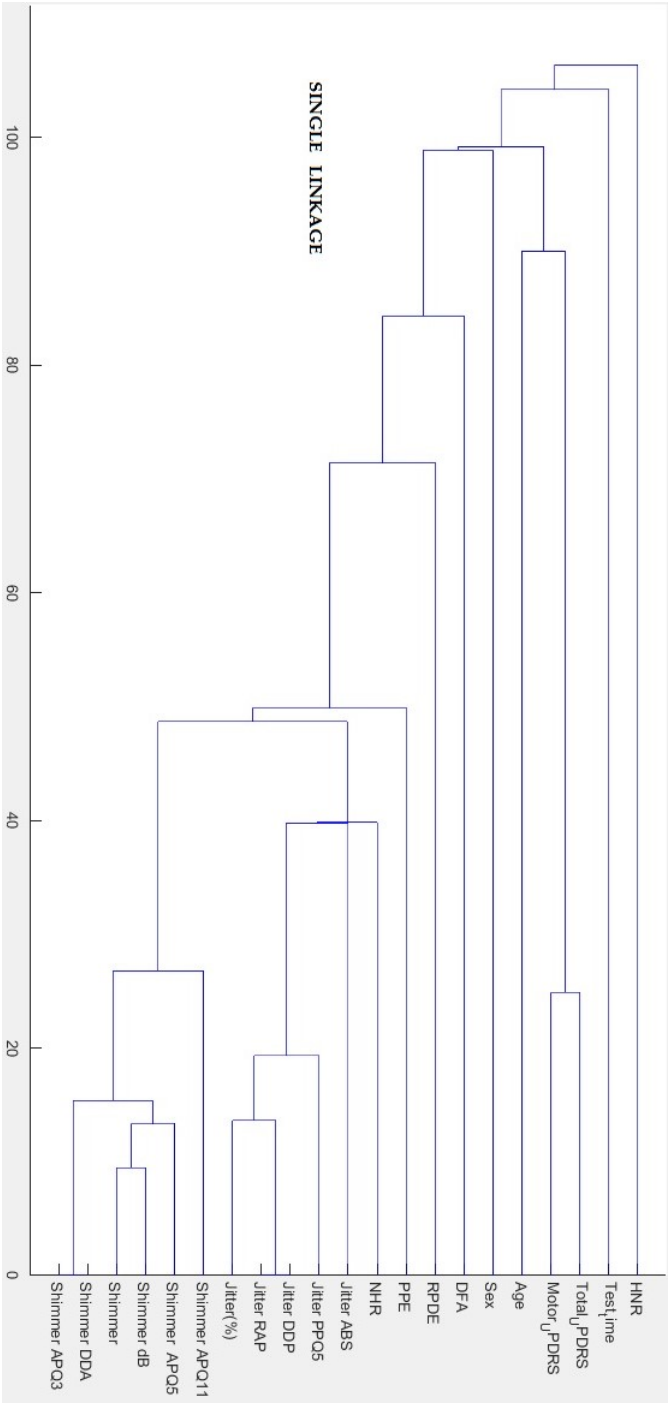
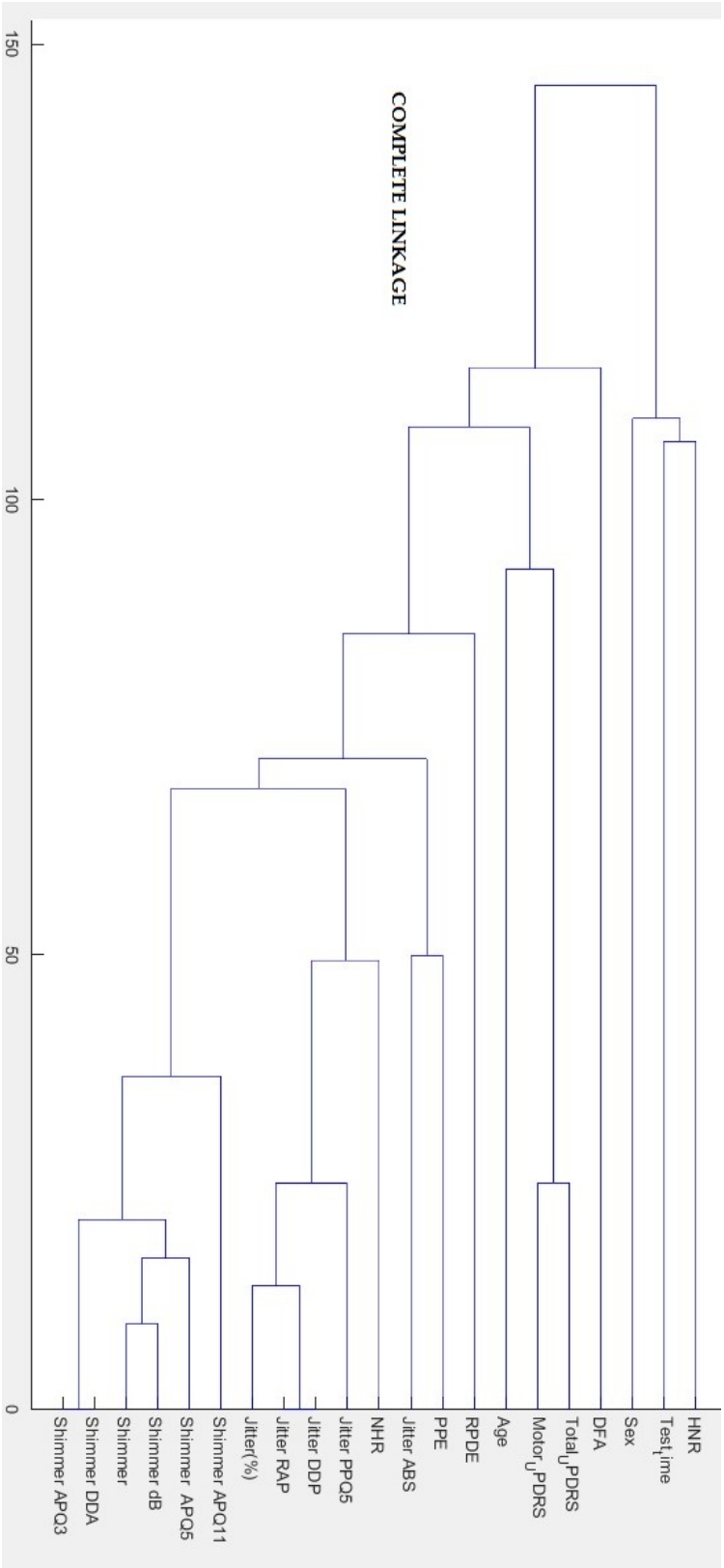


Figura 4.2: Plot delle correlazioni

- Usando invece ora la distanza **Euclidean**, otteniamo i seguenti dendrogrammi, che non sono molto differenti dai precedenti. Quindi non possiamo fare altre considerazioni.




COMPLETE LINKAGE
150
100
50
0
Shimmer APQ3
Shimmer DDA
Shimmer
Shimmer dB
Shimmer APQ5
Shimmer APQ11
Jitter(%)
Jitter RAP
Jitter DDP
Jitter PPQ5
NHR
Jitter ABS
PPE
RPDE
Age
Motor_UPDRS
Total_UPDRS
DFA
Sex
Test_time
HNR

4.3 Metodo delle k-medie

Anche del metodo delle k-medie ne abbiamo già parlato nell'Introduzione: è un metodo gerarchico e abbiamo visto che prende in input il numero k di clusters che vogliamo si formino. Considerando il numero che abbiamo assegnato ad ogni variabile, nella Sezione 4.1, otteniamo:

- Con $k = 3$, i seguenti gruppi:
  - 6,7,8,9,10,11,12,13,14,15,16,17,19,20,21;
  - 2,3,18;
  - 1,4,5;

  però non siamo molto soddisfatti;

- Con $k = 5$ invece il risultato sembra soddisfacente, perchè i gruppi sono i seguenti:
  - 6,7,8,9,10,19,20,21;
  - 11,12,13,14,15,16,17;
  - 2 (il fatto che il sesso stia in gruppo a sè è significativo: la malattia sorge indistintamente dal sesso);
  - 1,4,5;
  - 3,18;

- Solo con $k = 8$, si divide anche l'1 dal gruppo (45), cioè il fattore età dal gruppo dei disturbi (motor e total UPDRS).

In ogni caso, mandando più volte il *run* dell'algoritmo, anche senza cambiare k, si vede che i gruppi non sono troppo stabili.
Questa analisi è stata fatta utilizzando la distanza **correlation**, ma con **euclidean** si ottengono più o meno gli stessi clusters.

4.4 Spectral Clustering

Anche per lo Spectral Clustering bisogna dare in input il numero k di cluster che desideriamo, e scegliere la distanza da usare, in modo da poter poi confrontare i risultati.
L'algoritmo crea la matrice delle distanze D, poi genera $L = D - W$ e calcola i suoi autovalori, ma non in modo ordinato. Per questo motivo li riordiniamo in modo crescente e andiamo a calcolare la matrice di autovettori (ordinati). A questo punto facciamo girare l'algoritmo delle k-medie, associando prima la funzione di minimizzazione $RatioCut$, poi $Ncut$ (vedi Sezione 1.5), ottenendo così risultati differenti.

- Con la distanza **correlation**:
  - Con $k = 3$, otteniamo:
    - Con **RatioCut**:
      - 1,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21;
      - 2;
      - 3;

      ed è palese che questi cluster non hanno un raggruppamento logico.
    - Plottando con **NCut**, otteniamo esattamente la stessa cosa. Dunque proviamo a fare un tentativo con k un po' più grande.
  - Con $k = 5$, otteniamo già un risultato più soddisfacente (a parte il primo gruppo, contenente il solo esame DFA (20), molto simile a quello ottenuto coi precedenti metodi:
    - Con **RatioCut**:
      - 20;
      - 2,3;
      - 1;
      - 6,7,8,9,10,11,12,13,14,15,16,17,18,19,21;

⋆ 4,5;

* In questo caso plottando con **NCut**, otteniamo un risultato migliore, poichè questa funzione massimizza anche la similarità all'interno di ogni cluster:

⋆ 2;

⋆ 3;

⋆ 1;

⋆ 6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21;

⋆ 4,5;

- Con la distanza **euclidean**, otteniamo circa la stessa cosa.

4.5 Rappresentazione grafica con mesh

Una **mesh** poligonale è una collezione di vertici, spigoli e facce che definiscono la forma di un oggetto poliedrico nella grafica 3D.

- Un **vertice** è la rappresentazione di una posizione nello spazio;
- Un **lato** è una connessione tra due vertici;
- Una **faccia** è un insieme di punti nello spazio racchiuso tra vertici e lati.

L'insieme di queste facce può determinare poligoni o strutture molto più complesse.

Inoltre, la **connettività** della mesh descrive la relazione di incidenza tra gli elementi della mesh; mentre la **geometria** specifica la posizione e altre caratteristiche geometriche di ogni vertice.

Dopo aver riordinato in modo crescente gli autovalori della matrice $L = D - W$, e dopo aver costruito la matrice degli autovettori (con autovalori ordinati), facciamo girare sui suoi elementi l'algoritmo delle k-medie con $k = 5$ e distanza *correlation* (come negli esempi precedenti). Al termine riordiniamo i risultati ottenuti nella matrice W e facciamo una **mesh** su di essa, generando così una superficie con griglia colorata (vedi Figura 4.3), oppure applicando **surf** una superficie colorata (vedi Figura 4.4).

Quello che notiamo è l'evidente **compattamento a blocchi**, con diversi colori ma soprattutto diverse forme geometriche, che evidenzia i vari **cluster** che si sono formati.

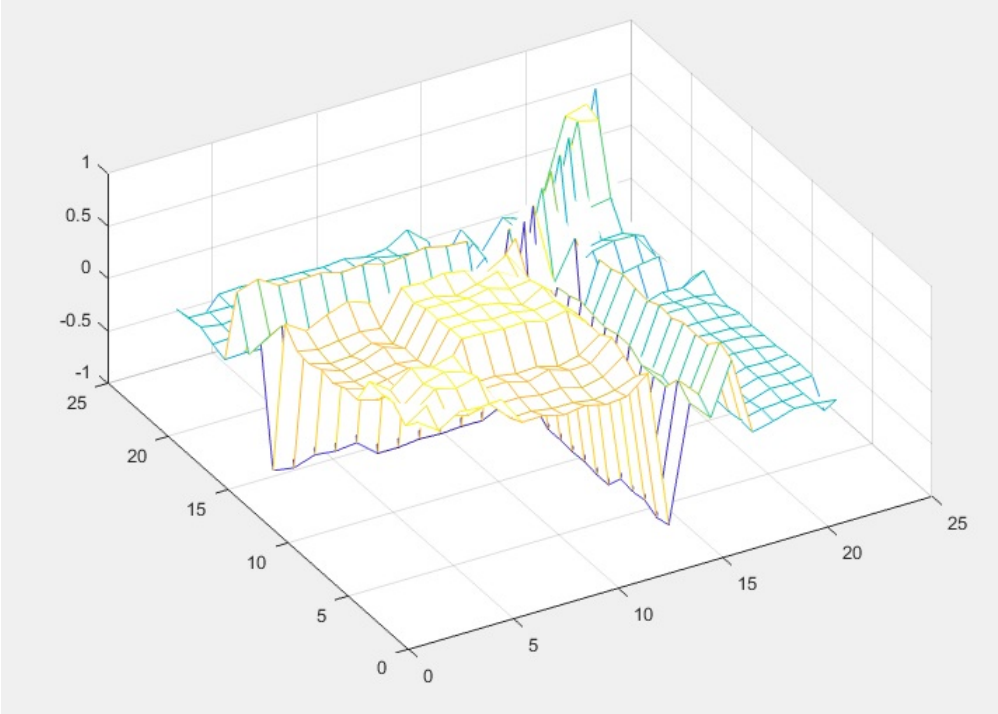


Figura 4.3: Mesh senza colori

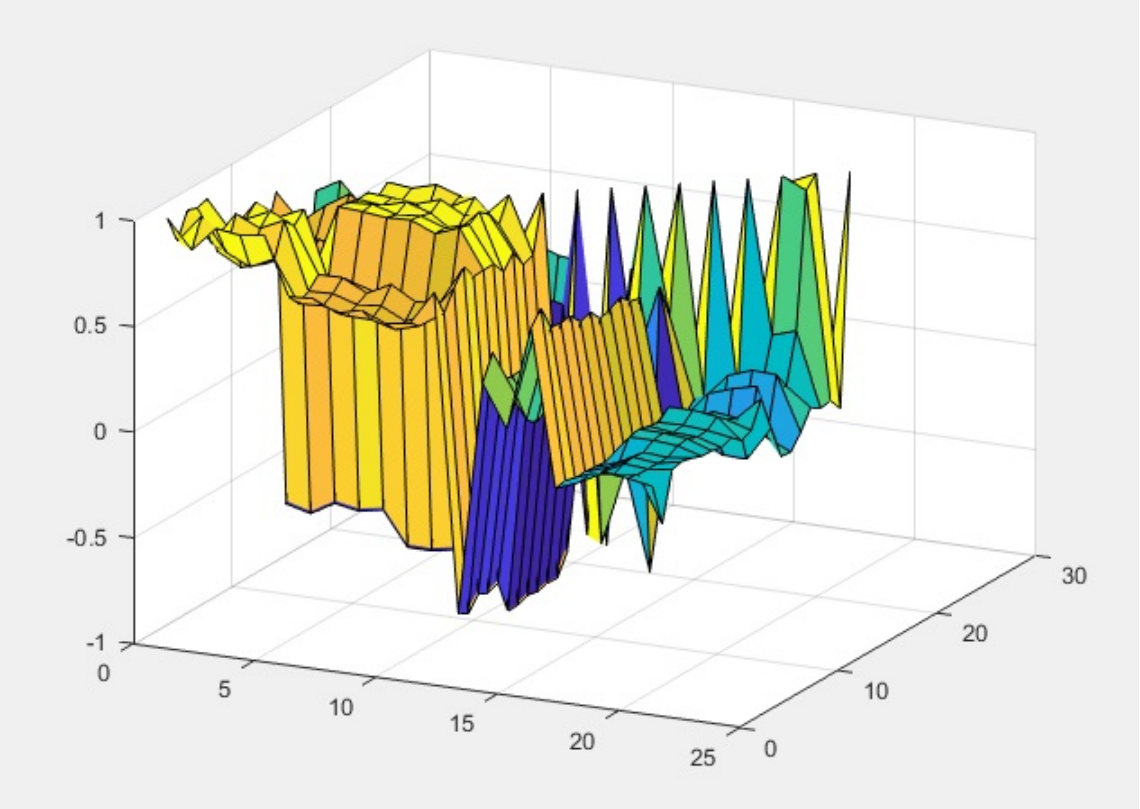


Figura 4.4: Mesh colorato (surf)

Conclusioni

L'**obiettivo** di questa tesi è stato quello di analizzare nel dettaglio un insieme di tecniche di analisi dei dati, volte alla selezione e al raggruppamento di elementi omogenei, in modo che si possano facilmente interfacciare tra di loro e fornire un utilizzo più semplice per chi opera nel settore.
Abbiamo introdotto la trattazione dei principali metodi di clustering, giungendo a queste conclusioni:

- Il clustering gerarchico (**Linkage**) non necessita della definizione a priori del numero di cluster, ma è oneroso dal punto di vista computazionale e inoltre è scarsamente efficiente con grandi moli di dati;

- Il clustering non gerarchico (**K-means**) è di immediata comprensione ed implementazione, è relativamente efficiente, ma occorre specificare a priori il numero di cluster k e inoltre funziona solo su valori numerici in quanto minimizza una funzione di costo calcolando la media dei cluster;

- Anche nello **Spectral Clustering** bisogna specificare a priori il numero di cluster k, ma supera alcune problematiche incontrate dagli algoritmi partizionali: ad esempio in questo caso i punti dati di input vengono mappati in modo non lineare in uno spazio di funzioni multidimensionale. Questo perchè lo Spectral Clustering lavora con gli autovalori della matrice di similarità per eseguire la riduzione di dimensionalità prima di raggruppare. Gli autovettori rilevanti sono quelli che corrispondono ai più piccoli autovalori del Laplaciano (tranne per il più piccolo autovalore che è 0). Dopo aver eseguito la riduzione, possiamo

rilassare il problema ed applicare l'algoritmo di k-medie. Abbiamo visto dunque quanto è importante effettuare un ottimo partizionamento del grafo, e che la qualità del taglio è confermata dal fatto che il secondo autovettore (e quelli a seguire) sia costante a tratti, dove i tratti conformi rendano i cluster, nello spazio a dimensionalità ridotta, ben distinguibili.

Bibliografia

[1] Aldous, D. and Fill, J. (in preparation). Reversible Markov Chains and Random Walks on Graphs.

[2] Bach, F. and Jordan, M. (2004). Learning spectral clustering. In S. Thrun, L. Saul, and B. Scholkopf (Eds.), Advances in Neural Information Processing Systems 16 (NIPS) (pp. 305 312). Cambridge, MA: MIT Press.

[3] Bapat, R., Gutman, I., and Xiao, W. (2003). A simple method for computing resistance distance. Z.Naturforsch., 58, 494 498.

[4] Barnard, S., Pothen, A., and Simon, H. (1995). A spectral algorithm for envelope reduction of sparse matrices. Numerical Linear Algebra with Applications, 2 (4), 317 334.

[5] Belkin, M. (2003). Problems of Learning on Manifolds. PhD Thesis, University of Chicago.

[6] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15 (6), 1373 1396.

[7] Belkin, M. and Niyogi, P. (2005). Towards a theoretical foundation for Laplacian-based manifold methods. In P. Auer and R. Meir (Eds.), Proceedings of the 18th Annual Conference on Learning Theory (COLT) (pp. 486 500). Springer, New York.

[8] Ben-David, S., von Luxburg, U., and P´al, D. (2006). A sober look on clustering stability. In G. Lugosi and H. Simon (Eds.), Proceedings of the 19th Annual Conference on Learning Theory (COLT)(pp. 5 19). Springer, Berlin.

[9] Bengio, Y., Delalleau, O., Roux, N., Paiement, J., Vincent, P., and Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel PCA. Neural Computation, 16, 2197 2219.

[10] Ben-Hur, A., Elisseeff, A., and Guyon, I. (2002). A stability based method for discovering structure in clustered data. In Pacific Symposium on Biocomputing (pp. 6 17).

[11] Bhatia, R. (1997). Matrix Analysis. Springer, New York.

[12] Bie, T. D. and Cristianini, N. (2006). Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems . JMLR, 7, 1409 1436.

[13] Bolla, M. (1991). Relations between spectral and classification properties of multigraphs (Technical Report No. DIMACS-91-27). Center for Discrete Mathematics and Theoretical Computer Science.

[14] Bremaud, P. (1999). Markov chains: Gibbs fields, Monte Carlo simulation, and queues. New York: Springer-Verlag.

[15] Brito, M., Chavez, E., Quiroz, A., and Yukich, J. (1997). Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. Statistics and Probability Letters, 35, 33 42.

[16] Bui, T. N. and Jones, C. (1992). Finding good approximate vertex and edge partitions is NP-hard. Inf. Process. Lett., 42 (3), 153 159.

[17] Chapelle, O., Scholkopf, B., and Zien, A. (Eds.). (2006). Semi-Supervised Learning. MIT Press, Cambridge.

[18] Chung, F. (1997). Spectral graph theory (Vol. 92 of the CBMS Regional Conference Series in Mathematics). Conference Board of the Mathematical Sciences, Washington.

[19] Dhillon, I. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD) (pp. 269 274). New York: ACM Press.

[20] Dhillon, I., Guan, Y., and Kulis, B. (2005). A unified view of kernel k-means, spectral clustering, and graph partitioning (Technical Report No. UTCS TR-04-25). University of Texas at Austin.

[21] Ding, C. (2004). A tutorial on spectral clustering. Talk presented at ICML.

[22] Ding, C., He, X., Zha, H., Gu, M., and Simon, H. (2001). A min-max cut algorithm for graph partitioning and data clustering. In Proceedings of the first IEEE International Conference on Data Mining (ICDM) (pp. 107 114). Washington, DC, USA: IEEE Computer Society.

[23] Donath, W. E. and Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. IBM J. Res. Develop., 17, 420 425.

[24] Fiedler, M. (1973). Algebraic connectivity of graphs. Czechoslovak Math. J., 23, 298 305.

[25] Fouss, F., Pirotte, A., Renders, J.-M., and Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans. Knowl. Data Eng, 19 (3), 355369.

[26] Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. JASA, 97, 611 631.

[27] Gine, E. and Koltchinskii, V. (2005). Empirical graph Laplacian approximation of Laplace-Beltrami operators: large sample results. In Proceedings of the 4th International Conference on High Dimensional Probability (pp. 238 259).

[28] Golub, G. and Van Loan, C. (1996). Matrix computations. Baltimore: Johns Hopkins University Press.

[29] Guattery, S. and Miller, G. (1998). On the quality of spectral separators. SIAM Journal of Matrix Anal. Appl., 19 (3), 701 719.

[30] Gutman, I. and Xiao, W. (2004). Generalized inverse of the Laplacian matrix and some applications. Bulletin de lAcademie Serbe des Sciences at des Arts (Cl. Math. Natur.), 129, 15 23.

[31] Hagen, L. and Kahng, A. (1992). New spectral methods for ratio cut partitioning and clustering. IEEE Trans. Computer-Aided Design, 11 (9), 1074 1085.

[32] Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of statistical learning. New York: Springer.

[33] Hein, M. (2006). Uniform convergence of adaptive graph-based regularization. In Proceedings of the 19th Annual Conference on Learning Theory (COLT) (pp. 50 64). Springer, New York.

[34] Hein, M., Audibert, J.-Y., and von Luxburg, U. (2005). From graphs to manifolds - weak and strong pointwise consistency of graph Laplacians. In P. Auer and R. Meir (Eds.), Proceedings of the 18th Annual Conference on Learning Theory (COLT) (pp. 470 485). Springer, New York.

[35] Hendrickson, B. and Leland, R. (1995). An improved spectral graph partitioning algorithm for mapping parallel computations. SIAM J. on Scientific Computing, 16, 452 469. Joachims, T. (2003). Transductive

Learning via Spectral Graph Partitioning. In T. Fawcett and N. Mishra (Eds.), Proceedings of the 20th international conference on machine learning (ICML)(pp. 290 297). AAAI Press.

[36] Kannan, R., Vempala, S., and Vetta, A. (2004). On clusterings: Good, bad and spectral. Journal of the ACM, 51 (3), 497515.

[37] Kempe, D. and McSherry, F. (2004). A decentralized algorithm for spectral analysis. In Proceedings 30 of the 36th Annual ACM Symposium on Theory of Computing (STOC) (pp. 561 568). New York, NY, USA: ACM Press.

[38] Klein, D. and Randic, M. (1993). Resistance distance. Journal of Mathematical Chemistry, 12, 81 95.

[39] Koren, Y. (2005). Drawing graphs by eigenvectors: theory and practice. Computers and Mathematics with Applications, 49, 1867 1888.

[40] Lafon, S. (2004). Diffusion maps and geometric harmonics. PhD Thesis, Yale University.

[41] Lang, K. (2006). Fixing two weaknesses of the spectral method. In Y. Weiss, B. Scholkopf, and J. Platt (Eds.), Advances in Neural Information Processing Systems 18 (pp. 715 722). Cambridge, MA: MIT Press.

[42] Lange, T., Roth, V., Braun, M., and Buhmann, J. (2004). Stability-based validation of clustering solutions. Neural Computation, 16 (6), 1299 1323.

[43] Lovasz, L. (1993). Random walks on graphs: a survey. In Combinatorics, Paul Erdos is eighty (pp.353 397). Budapest: Janos Bolyai Math. Soc.

[44] Lutkepohl, H. (1997). Handbook of Matrices. Chichester: Wiley.

[45] M., Audibert, J.-Y., and von Luxburg, U. (2007). Graph laplacians and their convergence on random neighborhood graphs. JMLR, 8, 1325 1370.

[46] Meila, M. and Shi, J. (2001). A random walks view of spectral segmentation. In 8th International Workshop on Artificial Intelligence and Statistics (AISTATS).

[47] Mohar, B. (1991). The Laplacian spectrum of graphs. In Graph theory, combinatorics, and applications. Vol. 2 (Kalamazoo, MI, 1988) (pp. 871 898). New York: Wiley.

[48] Mohar, B. (1997). Some applications of Laplace eigenvalues of graphs. In G. Hahn and G. Sabidussi (Eds.), Graph Symmetry: Algebraic Methods and Applications (Vol. NATO ASI Ser. C 497, pp.225 275). Kluwer.

[49] Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I. (2006). Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In Y. Weiss, B. Scholkopf, and J. Platt (Eds.), Advances in Neural Information Processing Systems 18 (pp. 955 962). Cambridge, MA: MIT Press.

[50] Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14 (pp. 849 856). MIT Press.

[51] Norris, J. (1997). Markov Chains. Cambridge: Cambridge University Press.

[52] Penrose, M. (1999). A strong law for the longest edge of the minimal spanning tree. Ann. of Prob., 27 (1), 246 260.

[53] Pothen, A., Simon, H. D., and Liou, K. P. (1990). Partitioning sparse matrices with eigenvectors of graphs. SIAM Journal of Matrix Anal. Appl., 11, 430 452.

[54] Saerens, M., Fouss, F., Yen, L., and Dupont, P. (2004). The principal components analysis of a graph, and its relationships to spectral clu-

stering. In Proceedings of the 15th European Conference on Machine Learning (ECML) (pp. 371 383). Springer, Berlin.

[55] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (8), 888 905.

[56] Simon, H. (1991). Partitioning of unstructured problems for parallel processing. Computing Systems Engineering, 2, 135 148.

[57] Spielman, D. and Teng, S. (1996). Spectral partitioning works: planar graphs and finite element meshes. In 37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996) (pp. 96 105). Los Alamitos, CA: IEEE Comput. Soc. Press. (See also extended technical report.)

[58] Stewart, G. and Sun, J. (1990). Matrix Perturbation Theory. New York: Academic Press.

[59] Still, S. and Bialek, W. (2004). How many clusters, an information-theoretic perspective. Neural Comput., 16 (12), 2483 2506.

[60] Stoer, M. and Wagner, F. (1997). A simple min-cut algorithm. J. ACM, 44 (4), 585 591.

[61] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a dataset via the gap statistic. J. Royal. Statist. Soc. B, 63 (2), 411 423.

[62] Van Driessche, R. and Roose, D. (1995). An improved spectral bisection algorithm and its application to dynamic load balancing. Parallel Comput., 21 (1), 29 48.

[63] von Luxburg, U., Belkin, M., and Bousquet, O. (to appear). Consistency of spectral clustering. Annals of Statistics. (See also Technical Report 134, Max Planck Institute for Biological Cybernetics,2004)

[64] von Luxburg, U., Bousquet, O., and Belkin, M. (2004). On the convergence of spectral clustering on random samples: the normalized case. In J. Shawe-Taylor and Y. Singer (Eds.), Proceedings of the 17th Annual Conference on Learning Theory (COLT) (pp. 457 471). Springer, New York.

[65] von Luxburg, U., Bousquet, O., and Belkin, M. (2005). Limits of spectral clustering. In L. Saul, Y. Weiss, and L. Bottou (Eds.), Advances in Neural Information Processing Systems (NIPS) 17 (pp. 857 864). Cambridge, MA: MIT Press.

[66] Wagner, D. and Wagner, F. (1993). Between min cut and graph bisection. In Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science (MFCS) (pp. 744 750). London: Springer.

Ringraziamenti

Devo ammetterlo: contrariamente a quella che è l'opinione comune, io non vedevo l'ora che arrivasse questo giorno per i ringraziamenti! E finalmente ci siamo.

Vorrei ringraziare in primo luogo la mia famiglia: grazie perchè mi siete stati accanto e non mi avete mai fatto mancare il vostro sostegno e il vostro aiuto in questi tre anni. Senza di voi non sarei mai diventata quella che sono oggi e non avrei potuto coronare i miei sogni.
Un grazie speciale lo voglio fare quindi a te **Papà** perchè anche se all'inizio non eri d'accordo "perchè Bologna è lontana" e perchè "chissà se ce la faremo"...volevo dirti che **ce l'abbiamo fatta!** E poichè tu sei di poche parole, so che un mio enorme **GRAZIE** può bastare a toccarti il cuore. Grazie per quello che fai per me e grazie perchè ci sei. Solo crescendo e standoti lontana, ho capito quanto fossero preziosi quei NO, per la mia educazione e soprattutto per aver ottenuto alcune cose solo crescendo.
Per seconda, ma non per importanza volevo ringraziare TE, **Mamma**. Grazie perchè mi hai messo al mondo e da quel giorno non mi hai MAI lasciato la mano e so che mai lo farai. Grazie perchè sei una presenza fissa, costante, il mio specchio, ma soprattutto il mio **NUMERO PREFERITO**. Il tempo che mi dedichi, è per me vitale. Ma soprattutto grazie, per tutte le cose che mi hai cucinato in questi tre anni nonostante gli 800 km di distanza. La mia pancia e il mio cuore ti ameranno per sempre.
Grazie alle mie Sorelle: a te **Angela**, perchè la tua persona mi fa da esempio,

perchè io vorrei essere come te con un pizzico di me. Scusa se non sono stata per te quello che tu sei stata per me. Grazie per essermi venuti a trovare ogni anno e ne approfitto per dire grazie anche a te **Rocco**. Perdonami per le multe a San Luca e per le vittorie in Via Don Minzoni e per le corse bagnate dove i portici non esistono...ma è quel che combino quando sento qualcuno membro Speciale della Famiglia. Grazie poi a te **Mari**, perchè sai essere scontrosa ma anche premurosa...e nonostante litighiamo spesso per stupidate, so che ci sei sempre e che con te posso sempre confidarmi. E grazie anche a **Domenico** per i suoi pensieri carini, per avermi dato l'opportunità di insegnargli un po' di matematica, ma soprattutto grazie se oggi sei qui vicino a me.
DIRE CHE VI VOGLIO BENE PENSO SIA VERAMENTE RIDUTTIVO.

Prima di procedere, vorrei ringraziare la Prof.ssa **Valeria Simoncini**, una Donna brillante che mi ha fatto innamorare delle sue lezioni e delle sue passioni, dal primo giorno in cui mi notò "smanettare" al computer. E' stato Amore a prima vista... Per questo prof, per avermi convinta quel giorno a ridare l'esame di calcolo, per avermi incoraggiata quando Matlab sclerava e io non ne potevo più, per l'interesse e la cura mostrati nella stesura della tesi, e per tanto altro che **Lei sicuramente sa**, io la ringrazio oggi e sono sicura che continuerò a farlo anche in futuro.

Subito dopo, vorrei ringraziare la mia seconda **Famiglia**, quella che non ho scelto, ma che il destino mi ha gentilmente regalato. In primis, le mie due storiche coinquiline: per chi non le conoscesse sono quelle con le quali (per mancanza di volontà da parte di tutt'e tre ad alzarsi per prender lo zucchero) ho imparato a bere il caffè amaro... ma sono anche quelle persone con le quali sono cresciuta, ho imparato a mangiare tutto, ma soprattutto ho imparato bene cosa vuol dire pazientare e rispettare.

Alice, la mia ancora, il mio opposto, il mio Tutto.
Inutile descriverti, tutti sanno chi sei. Ci siamo beccate per caso il lontano 20 settembre 2014 in via Gianlodovico Bianconi 6. Dopo un giorno e mezzo, i miei son ripartiti e Mamma in lacrime, ci disse **"VOGLIATEVI BENE"**. Ecco, Mamma, l'abbiamo fatto sul serio, ti abbiamo dato veramente ascolto. Sono troppe le cose che abbiamo fatto insieme, e raccontarle...sarebbe illegale. Forse potrebbero chiedermi di restituire la corona d'alloro. Un ringraziamento dal cuore va a te, mia cara Alicetta. Grazie perchè esisti e perchè mi sei sempre vicina pur nei miei numerosi difetti, grazie perchè ovunque possiamo essere, i momenti passati con te sono sempre straordinari ed indimenticabili. Non smetterò mai di ringraziarti per le pizze al Veliero, per i gelati da Gianni, in Funivia o quelli NOSTOP a Funnyhouse, per le giornate di studio e i mille caffè, per le passeggiate in centro, per i giri in bici. Non dimenticherò mai questi anni in cui ti ho veramente conosciuta e sostenuta in tutte le scelte, anche quelle più difficili, proprio come tu hai fatto con me. Credo fortemente che quelli che ci si apriranno davanti saranno anni altrettanto belli e ricchi di esperienze, nei quali potremo fare errori o raggiungere nuovi obiettivi, ma che condivideremo l'una con l'altra. Grazie Ali, perchè quando cerco di volgere lo sguardo in avanti per capire cosa mi aspetterà, non riesco a non vederti **vicino a me**, così che ogni paura si allontana ed è sostituita dalla voglia di intraprendere questo nuovo percorso con te.

Alessandra invece, la mia coscienza, il mio grillo parlante.
Per fortuna che nella scelta della coinquilina, non ho dato ascolto ad Alice sennò mi trovavo quella veneta figa in casa, che magari mi rubava pure i mori palestrati... Scherzi a parte, GRAZIE Ale anche a te, perchè sei entrata nella mia vita con un "posso vedere la stanza?" e non te ne sei andata più, nonostante i nostri caratteri così diversi che ci hanno portato spesso, i primi tempi, a dircene tante. Grazie Ale, perchè hai sempre una parola di conforto oltre che un abbraccio. Tu che non sei la terza come spesso si pensa, ma colei che completa un tris perfetto. Siamo così diverse che solo insieme

possiamo stare bene. Oltre che tanti messaggi minacciosi scritti insieme a maschi deficienti, non dimenticherò mai le chiamate alla Polizia nel cuore della notte per le tue mille paure...e l'immancabile Carabiniere Ciruzz che ci protegge da sotto il portico. Grazie perchè mi hai consolato un miliardo di volte e perdonami se la mia vita è un eterno ritorno, che richiede il tuo eterno conforto. Vorrei che la Nostra Casa non cambiasse mai, ma se la tua vita cambierà rotta per seguire i tuoi sogni, spero che ci sarà sempre nel tuo cuore un posto per me, perchè nel mio... nessuno mai prenderò il tuo.

Poi volevo dire grazie a **Maria**, mia sorella acquisita più che cugina. Nonostante gli impegni, lo studio e il lavoro, non ci permettono di stare insieme tutti i giorni, sei per me un punto di riferimento. E lo sei stata dal primo giorno che ho messo piede a Bologna. Grazie perchè mi hai aiutata e perchè mi hai dato forza quando ero veramente in preda ad un esaurimento nervoso.

Grazie alle mie amichette sarde, con le quali ho condiviso segreti...e vita. **Stuggiu**, ti chiamo così sennò non saresti tu...la più simile a me. Testarda, istintiva e presuntuosa. Ci battibecchiamo e poi andiamo a far la nanna insieme per fare pace. Una piccola cuoca e pasticcera, custode di segreti e di emozioni. Grazie perchè in questi tre anni ci sei stata sempre, grazie per i salatini offerti nei miei momenti peggiori e per il cornetto alle quattro di notte per evitare un assassinio. Hanno fatto di te una persona che amerò per sempre.
Vanessa, la mamma di tutte. Responsabile, seria, leale. Grazie perchè tu mi hai dato risposte a cuore freddo, che mi zittivano e mi convincevano ad ascoltarti. Grazie per quelle colazioni al bar infinite e grazie per tutto il tempo che mi hai concesso, togliendolo ai tuoi impegni. Sei un'amica speciale che non vorrei perdere mai.
Rossella, l'organizzatrice di spritz e merende. Grazie a te invece perchè nonostante non riusciamo sempre a rispettare gli incontri, tu sei sempre li a proporne altri. Ci hai fatto conoscere persone per te speciali, che lo sono

diventate anche per noi e ne approfitto per ringraziare **#IlTimavoSRL** per le nostre notti NOLIMITS e per tutte le volte che abbiamo rischiato, senza paura, di essere sfrattati.

Siccome mi son dilungata troppo, in carrellata e senza un ordine preciso, ringrazio:
Le mie **Zie** e i miei **Zii** (troppi per citarvi tutti, ma ringrazio con tutto il cuore soprattutto chi è qui!), e in particolare Zia Rosa, che ha mantenuto una promessa durata tre anni, Zia Assunta, che è per me un'amica più che una zia, Zia Anna, per il supporto morale e la fiducia datami e Zia Giovanna, mia compagna di viaggi e corse a Bologna.
I miei **Cuginetti**, grandi e piccoli, ma soprattutto **Carmen, Gianuario, Angelica, Pietro, Samuel, Rosangela, Angelomario**, e quelli che hanno creduto in me e sentono di essermi stati affianco in questi anni. **Angelo e Gabriel**, miei piccoli fratellini adottivi, vi ringrazio per gli audio, le letterine, l'Amore e il senso di Mancanza, che mi avete mostrato ogni giorno e ogni qualvolta tornavo e ripartivo. Siete i miei gioielli.
Grazie alle mie **Nonne**, che con chiamate last minute si sono preoccupate circa la mia salute e la mia pancia...sto bene Nonne, benissimo. E vorrei fosse lo stesso sempre anche per voi!

Grazie ai miei **amici liceali**, soprattutto a chi è qui con me, perchè questo vuol dire che non sono mai andati lontani...e di questo sono terribilmente felice. Grazie in particolare a **Latorraca**, mio bimbo e fallito coinquilino, mio migliore amico, mio supporter e mio primo fan e **Umberto**, il mio amico più bello e affascinante, elegante nella sua persona e incredibilmente camaleontico, non svelo qui i tuoi più feroci cambiamenti, che solo in pochi capiranno.
Grazie ai miei amici di sempre, **Francesca e Daiana**, lontane ma sempre vicine, **Paolo e Conte**, **Monaco**, mio super cucciolo, **Gennaro**, che ha sempre creduto in questa "super prof", **Paola e Simone**, **Pietro e Carmelo**, miei eterni guai e miei dolci angoli di cuore... vorrei che oggi ci foste

tutti, ma in ogni caso, se non è così, ci siete e ci siete stati. E per questo vi ringrazio! Ognuno di voi sa personalmente quanto vale per me, e questo è quello che conta.

Credo fermamente che l'amicizia tra uomo e donna esista e Bologna mi ha dato ancora più conferme. A proposito di questo, vorrei ringraziare i miei Super Amici Maschioni **Pasquale mio nuovo best, Ciccio Surace, Silvio Tani cccezionale e Salvatore Martino**, così citati per non farvi perdere l'importanza dei vostri nomi. Grazie per le notti allo chalet, in piazza Verdi, in casa nostra o vostra, grazie per avermi insegnato giochi alcolici, ma soprattutto grazie per essermi stata affianco in un modo così tenero e genuino. Siete delle persone splendide.
Grazie poi a **Diletta, Martina, Ilaria, Marialaura**, altre compagne di vita non indifferenti. Alcune più brave, altre più monelle...e chiaramente l'ordine dei nomi confonde le idee sul chi e sul come. Siete dolcissime e sono felice di esser arrivata qui oggi, con i vostri in bocca al lupo, come ad ogni esame, ma soprattutto con voi...

Grazie a tutti, a chi c'era e c'è, a tutte le persone che ho conosciuto in questi anni, ai miei amici di facoltà, ai miei amici seratisti, alla Polizia sempre presente e a tutti coloro che in qualche modo hanno fatto la loro comparsa, andando o restando.
Grazie, infine a: Rocco, Alessia, Martina, Furio, Megan, Leo, Emanuele, Francesco, Giulia, MariaVittoria...e tanti altri. Voi siete e sarete i **miei ragazzi delle ripetizioni** perchè mi avete dato l'opportunità di crescere professionalmente e moralmente. Vedervi migliorare e festeggiare gli esami, è stato per me **un successo**. Vi porterò tutti nel cuore.

E se sei arrivato a leggere fin qui...GRAZIE.
Grazie soprattutto perchè vuol dire che ci sei.
Buona Mia Festa di Laurea anche a te!