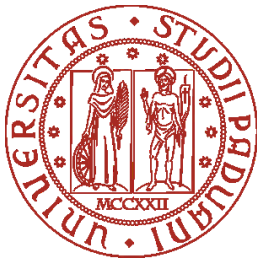
Università degli Studi di Padova

Dipartimento di Scienze Statistiche

Corso di Laurea Triennale in
Statistica per l'Economia e l'Impresa



RELAZIONE FINALE

Metodi Statistici per la Stima del Numero di Cluster

Relatore Dott. Erlis Ruli

Dipartimento di Scienze Statistiche

Laureando Luca Riotto

Matricola 1233240

Anno Accademico 2021/2022

Indice

Prefazione **5**

1 Introduzione al clustering **7**
1.1 Cluster analysis 7
1.2 Metodi gerarchici e di partizione 8
1.2.1 Il clustering gerarchico 8
1.2.2 Metodi di partizione 11
1.3 Metodi basati su modelli 12

2 Stima del numero di gruppi **15**
2.1 Indici esterni 15
2.1.1 Indici classification-oriented 16
2.1.2 Indici similarity-oriented 17
2.2 Indici interni 18
2.2.1 La silhouette 18
2.2.2 Indice di Calinski e Harabasz 19
2.2.3 La statistica Gap 19
2.2.4 L'indice Ray Turi 20
2.2.5 Metodo jump 21
2.2.6 Cluster Stability 21
2.3 Inferenza 22
2.3.1 Test del log-rapporto di verosimiglianza 23
2.3.2 Criteri di informazione 25

3 Simulazioni **27**
3.1 Scenario 1: k=1, p=10 28
3.2 Scenario 2: k=2, p=3 29
3.3 Scenario 3: k=2, p=3 29
3.4 Scenario 4: k=3, p=2 31
3.5 Scenario 5: k=4, p=2 32
3.6 Scenario 6: k=5, p=2 33
3.7 Scenario 7: k=4, p=2 33
3.8 Scenario 8: k=5, p=2 34

3.9 Riepilogo risultati 35

4 Analisi dataset reali **37**
4.1 Dataset iris 37
4.2 Classificazione di cellule tumorali tramite spettri SERRS . . . 39

Conclusioni **43**

A Codice R delle simulazioni **45**

Bibliografia **52**

Prefazione

La cluster analysis è un insieme di tecniche di analisi multivariata dei dati volte alla selezione e al raggruppamento di elementi omogenei in un insieme di dati.
I metodi erano già noti alla fine del XIX secolo, ma l'interesse da parte degli statistici si è avuto attorno agli anni '60. Ad oggi, grazie soprattutto alle potenzialità del software e dell'hardware a disposizione si conta un'enorme varietà di algoritmi, sempre più efficienti e con diversi gradi di difficoltà computazionale. I campi di applicazione sono numerosi, dalle scienze fisiche (fisica, medicina, biologia), a quelle sociali (economia, sociologia, psicologia).
Un clustering può essere definito come una partizione di un insieme di unità elementari, oppure come raggruppamento di unità molto simili tra loro in gruppi che abbiano la caratteristica di essere il più possibile distinti tra loro. Tale partizione viene definita in base ad una misura di similarità o distanza (la più utilizzata è la distanza euclidea) tra le unità. Si tratta di un insieme di metodi di apprendimento non supervisionato, ovvero un insieme di algoritmi che classificano le unità in gruppi senza l'ausilio di etichette che permettono di individuare le relazioni tra le unità, il modello viene costruito con i soli dati di input e l'algoritmo scopre da solo le relazioni nascoste nei dati.
Uno dei problemi più comuni a tutte le tecniche di *clustering* è la difficoltà nel determinare il numero di cluster presenti in un certo insieme di dati in quanto non si hanno a disposizione classi predeterminate o in generale altre informazioni che possano aiutare a classificare le osservazioni.
Lo scopo di questa tesi è proprio quello di analizzare alcuni metodi per l'individuazone del numero ottimale di cluster e confrontare le performance di tali metodi in diversi scenari di simulazione. Tali metodi saranno poi implementati e testati in un dataset reale.
Nel primo capitolo vengono presentati alcuni tra gli algoritmi più utilizzati di analisi dei gruppi, quali: i metodi gerarchici, i metodi di partizione ed i metodi basati su modelli. Nel secondo capitolo vengono illustrati alcuni metodi di stima del numero ottimale di cluster in cui sono distinti gli indici interni e i metodi basati sulla verosimiglianza.
Nel terzo e quarto capitolo vengono analizzate le performance dei diversi metodi presentati nel secondo capitolo mediante rispettivamente uno studio di simulazione e l'analisi di due dataset reali.

Capitolo 1

Introduzione al clustering

1.1 Cluster analysis

Le tecniche esistenti di analisi di raggruppamento possono essere suddivise in due diverse tipologie.
Una prima tipologia, al quale appartengono i metodi gerarchici come: il metodo del legame singolo, del legame completo, del legame medio, il metodo di Ward e i metodi di partizione quali il metodo k-means, tutti metodi che si basano su misure si similarità (distanza).
Le metodologie appartenenti a questa categoria presentano, a fronte di una semplicità concettuale (e spesso computazionale), alcune problematiche legate alla mancanza di un modello statistico sottostante. Sebbene queste tecniche siano tra le più comunemente utilizzate, non permettono di ricorrere a procedure inferenziali (ad esempio il test del rapporto di verosimiglianza e i criteri di informazione automatica) e non affrontano il problema riguardante il numero dei gruppi presenti nei dati, che risulta essere uno dei problemi più delicati nell'ambito della cluster analysis.
Il secondo approccio al clustering è basato su modelli statistici (model-based clusering): i gruppi vengono associati a delle specifiche caratteristiche della distribuzione di probabilità che si assume possa descrivere adeguatamente i dati a disposizione. Le tecniche appartenenti a questa categoria hanno il vantaggio di poter fare inferenza sul numero dei gruppi o sulla bontà della partizione ottenuta. Tale approccio si suddivide in due tecniche di clustering distinte, una basata su modelli parametrici ed una basata sul clustering non parametrico. Il clustering parametrico si basa su un modello statistico parametrico per descrivere la densità dei dati, tale modello viene selezionato tra i modelli a mistura finita con componenti appartenenti ad una o piu famiglie di densità. La procedura di analisi prevede la stima del modello con il metodo della massima verosimiglianza, solitamente utilizzando l'algoritmo Expectation-Maximization (EM), che permette di calcolare la probabilità a posteriori di ogni componente della mistura data una determinata osserva-

zione; tale osservazione verrà infine assegnata alla componente con associata la probabilità posteriori più elevata. Si noti che tale approccio prevede che ogni componente della mistura rappresenti uno specifico gruppo, ciò può essere una limitazione nel caso in cui l'assunzione parametrica alla base sia violata o nel caso in cui i gruppi non siano nettamente separati. Inoltre tale approccio risulta essere computazionalmente oneroso quando si ha a che fare con dataset che contengono molte variabili in quanto il numero di parametri da stimare risulterà essere molto elevato. È comunque evidente come ricorrendo a tale approccio vi siano dei vantaggi statistici che permettono ad esempio di considerare misure di incertezza dell'analisi di raggruppamento (in termini di probabilità a posteriori) e di scegliere automaticamente il numero di gruppi presenti nei dati con criteri automatici quali il criterio di informazione di Bayes (BIC). Per ulteriori approfondimenti riguardo questo approccio si veda ad esempio Fraley and Raftery (2002).
Il clustering permette di creare una partizione delle unità statistiche anche partendo dalla rilevazione di sole variabili qualitative, in tal caso però non si hanno a disposizione alcune tecniche, ad esempio il metodo di Ward e il metodo k-means. In questo lavoro saranno affrontate le metodologie basate sulle sole variabili quantitative.

1.2 Metodi gerarchici e di partizione

1.2.1 Il clustering gerarchico

Sia $\mathbb{X}_{n*p}$ matrice dei dati con n righe e p colonne dove n sono le unità statistiche e p le variabili ed i vettori $x_1, \ldots, x_n$ i vettori riga di $\mathbb{X}$.
Siano $g_1, \ldots, g_k$ k gruppi provenieni da una partizione delle n unità statistiche e $\bar{x}_{g_1}, \ldots, \bar{x}_{g_k}$ i vettori dei centroidi dei k gruppi. Infine sia $d(x_i, x_j)$ una metrica di distanza tra x_i e x_j, tipicamente viene utilizzata la distanza euclidea indicata mediante $d_E(x_i, x_j)$. Il clustering gerarchico è un metodo molto utilizzato che ha il vantaggio di non richiedere la definizione a priori il numero di cluster da ricercare. Il clustering gerarchico raggruppa le unità in un albero di cluster tipicamente utilizzando una metrica di distanza, tuttavia, l'utilizzo di funzioni di distanza non è obbligatorio, alcuni algoritmi di clustering gerarchico utilizzano altri metodi quali ad esempio i metodi basati sulla densità o sui grafi.
Ne esistono due versioni:
Agglomerativo. Si parte assegnando un gruppo diverso per ogni singolo dato in ingresso. Si procede iterativamente ad agglomerare più cluster insieme (sfruttando metriche opportune) fino a quando si arriva ad avere un unico cluster che contiene tutti i dati.
Formalmente; sia n il numero di punti (tipicamente unità statistiche) in uno spazio d-dimensionale e $n_t = n - t$ il numero di cluster in seguito alla t-esima iterazione. Ad ogni aggregazione si ottiene una matrice di distanza di

dimensioni $n_t \times n_t$ che sarà utilizzata per determinare la successiva fusione tra due cluster, in pratica verranno agglomerati i due cluster che presentano minore distnza.

Esistono diversi modi per misurare la distanza fra due gruppi (g_k, g_k). Sono chiamati linkage:

- legame singolo, si pone come distanza tra due gruppi il minimo delle distanze tra le osservazioni dei due gruppi, ovvero;

$$d(g_i, g_j) = \min \{d(x_h, y_k) : x_h \in g_i, y_k \in g_j\}$$

- legame completo, si pone come distanza tra due gruppi il massimo delle distanze tra le osservazioni dei due gruppi, ovvero;

$$d(g_i, g_j) = \max \{d(x_h, y_k) : x_h \in g_i, y_k \in g_j\}$$

- legame medio (average linkage),

$$d(g_i, g_j) = media \{d(x_h, y_k) : x_k \in g_i, y_k \in g_j\}$$

- centroide più vicino, in tal caso i centroidi più vicini vengono agglomerati ad ogni iterazione, la matrice di distanza sarà quindi formata dalle distanze tra i centroidi. Tuttavia questo metodo non è particolarmente auspicabile in quanto i centroidi perdono di informazione sulla diffusione relativa ai differenti cluster, come afferma Aggarwal (2015), tale metodo non discriminerà tra la fusione di coppie di cluster di dimensioni diverse, purché i loro centroidi siano alla stessa distanza ed in genere è statiticamente più probabile che i centroidi dei cluster più grandi siano più vicini tra loro.

Divisivo. Si parte con un unico cluster che contiene tutti i dati e poi, iterativamente, si procede a suddividere i cluster esistenti in più sottocluster.
L'approccio del clustering gerarchco divisivo è quello di utilizzare un algoritmo di clustering A come *subroutine*. L'algoritmo inizializza "$l'albero$" in cui la radice contiene tutte le osservazioni, ad ogni iterazione i nodi dell'albero vengono scissi in più nodi, i cluster. Cambiando il criterio di selezione dei nodi si possono creare alberi bilanciati per altezza o per numero di cluster. Se l'algoritmo A scelto è il $k - means$ con $k = 2$ si parla di **bisecting k-means**, mediante tale algoritmo ogni nodo viene diviso in due nodi figli. Diverse varianti di questo approccio utilizzano strategie differenti per selezionare il nodo da dividere per primo, ad esempio può essere diviso per primo il nodo contenente il maggior numero di osservazioni oppure il nodo con minore distanza dalla radice.

In entrambi i casi, quindi, si esplorano tutte le possibili combinazioni di cluster, da un estremo (un unico cluster) all'altro (un cluster per dato) e viceversa. Nella pratica la versione agglomerativa è quella più utilizzata, per motivi più che altro di efficienza computazionale Il risultato può essere meglio compreso se, mentre otteniamo i cluster, andiamo a costruire il dendrogramma (*Figura 1.1*). Si tratta di un grafico ad albero dove sull'asse delle ordinate è riportata la "distanza" tra i cluster e sull'asse orizzontale vengono riportati i vari dati in ingresso. In questo diagramma, inoltre, le righe verticali corrispondono ad un cluster, quelle orizzontali ad operazioni di unione (se si usa la versione agglomerativa dell'algoritmo, che si legge dal basso verso l'alto) o di divisione (se si usa la versione divisiva dell'algoritmo, che si legge dall'alto verso il basso).

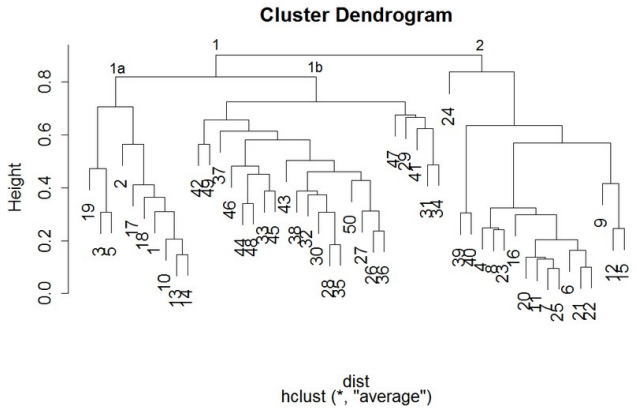


Figura 1.1: Esempio di dendogramma

Un altro metodo di clustering gerarchico molto famoso (che rientra tra i metodi agglomerativi) è quello di Ward. La tecnica di Ward si propone di minimizzare la varianza delle variabili entro ciascun gruppo. Ad ogni stadio vengono fusi i due gruppi che producono un minimo aumento della varianza totale entro i gruppi. Questa tecnica permette di generare gruppi di forma tendenzialmente sferica.
Il metodo di Ward minimizza, nella scelta dei gruppi da aggregare, una funzione obiettivo. Tale funzione è la somma delle devianze interne ai cluster,

ovvero:
$ESS_{tot,k} = ESS_1 + \ldots + ESS_k$, con $k \in \{1, \ldots n\}$ numero di cluster,

$$ESS_j = \sum_{x_i \in g_i} (x_i - \bar{x}_{g_i})^T (x_i - \bar{x}_{g_i})$$

con $j = 1, \ldots, k$. Ogni fusione genera un incremento di $ESS_{tot,k}$, la procedura si ferma quando tutti i gruppi sono stati fusi.

1.2.2 Metodi di partizione

Il k-means è uno degli algoritmi di clustering più diffuso, nonostante ciò è un algoritmo semplicissimo da implementare ed utilizzare. Si tratta di trovare la partizione dei dati per cui si ottiene

$$\min_{\mu_1,\ldots,\mu_k} \sum_{j=1}^{k} \sum_{i \in g_i} d_E(x_i, \mu_j)^2$$

dove, x_i è l'i-esima riga di $\mathbb{X}$, $\mu_j \in R^p$ è la media del gruppo g_j e il numero di gruppi k, viene specificato a priori. L'algoritmo più comune è:

1. Inizializzazione: suddividere le $o_1, \ldots, o_n$ (casualmente) in k gruppi e calcolare le medie dei gruppi, denominati centroidi. ;
2. Aggiornare i cluster: allocare ogni o_i al cluster più vicino;
3. Aggiornare i centroidi: calcolare le medie dei gruppi;
4. Iterare: ripetere 2-3 fino a convergenza (nessun aggiornamento).

Vi sono delle varianti di tale algoritmo, ad esempio Wilkin and X. (2007) ne mettono due a confronto. L'algoritmo k-means non raggiunge buoni risultati quando i cluster naturali hanno:

- diverse dimensioni
- diversa densità, ovvero diversa concentrazione delle osservazioni attorno al proprio centroide.
- forma non sferica (se $p = 2$) o non ipersferica (se $p > 2$)
- outlier.

L'algoritmo k-means può convergere a massimi locali, risulta quindi opportuno applicare più volte tale algoritmo agli stessi dati e scegliere la partizione che presenta la minima devianza interna ai cluster.

1.3 Metodi basati su modelli

Uno degli approcci di clustering che assume un modello statistico per la popolazione da cui sono campionati i dati, è noto come modello di misture finite di densità. Questo modello assume che la popolazione sia in realtà formata da un numero di sottopopolazioni, ovvero i cluster, ciascuna con una sua distribuzione di probabilità multivariata. Adottando questo approccio, il problema di clustering diventa quello di stimare i parametri della mistura di densità assunta, e quindi usare i parametri stimati per calcolare le probabilità (a posteriori) di appartenenza ai cluster per ciascuna osservazione. Inoltre, la determinazione del numero di cluster si riduce ad un problema di selezione del modello per cui esistono procedure oggettive. L'analisi dei cluster basata su modelli a mistura finiti sono anche noti come metodi di clustering model-based (basati su modello).
Siano $\theta = (\theta_1, \ldots, \theta_k)$, $\pi_j \in (0,1)$ $j = 1, \ldots, k$ con k numero di cluster dove $\sum_{j=1}^{k} \pi_j = 1$ e sia $f_j(x; \theta_j)$ una densità di probabilità p-variata. Una mistura di densità è definita come:

$$h(x; \theta, \pi) = \sum_{j=1}^{k} \pi_j f_j(x; \theta_j),$$

Uno dei modelli più utilizzati, in quanto uno dei piu semplici sia concettualmente che computazionalmente, è il modello derivante dalla mistura di distribuzioni normali (Gaussian Mixture Model), formalmente una mistura di densità di vettori casuali normali (p-dimensionali) è definita come segue:

$$f_X(x; \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k) = \sum_{j=1}^{k} \pi_j \frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\left\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right\}$$

dove k è il numero di cluster e $x \in R^p$.
La stima dei parametri utilizzando la massima verosimiglianza non ha una forma chiusa, si ricorre quindi alla massimizzazione mediante l'algoritmo EM (expectation maximization). Tale algoritmo, per una generica mistura di distribuzioni f_j, alla l-esima iterazione è cosi definito:

1. E-Step: calcolare

$$\omega_{i,j}^{(l+1)} = \frac{\pi_j^{(l)} f_j\left(x_i; \theta_j^{(l)}\right)}{\sum_{j=1}^{k} \pi_j^{(l)} f_j\left(x_i; \theta_j^{(l)}\right)} i = 1, \ldots, n; j = 1, \ldots, k$$

2. M-Step 1: fissare $\pi_j^{(l+1)} = \sum_{i=1}^{n} \frac{\omega_{i,j}^{(l+1)}}{n}$

3. M-Step 2: massimizzare

$$\sum_{j=1}^{k}\sum_{i=1}^{n}\omega_{i,j}^{(l+1)}\ln f_j\left(x_i;\theta_j\right)$$

rispetto a $\theta = (\theta_1 \ldots, \theta_k)$ (i parametri della mistura di distribuzioni) per ottenere un valore per $\theta^{(l+1)}$. Tale massimizzazione può essere scomposta in k massimizzazioni separate, grazie alla forma separabile della log-verosimiglianza, così facendo l'algoritmo EM risulta computazionalmente efficiente.

L'algoritmo di expectation maximization è sensibile ai valori iniziali scelti per i parametri, è necessario quindi che vengano scelti con particolare cura. I passaggi da seguire per la scelta dei valori iniziali dei parametri sono i seguenti:

1. determinare un partizionamento iniziale delle osservazioni $\mathbb{X} = [x_1|\ldots|x_n]$ in k sottoinsiemi $g_1, \ldots, g_k$ con $\bigcup_{j=1}^{k} g_j = \mathbb{X}$ di cardinalità rispettivamente $N_1, \ldots, N_k$. Tale partizionamento delle osservazioni può essere ottenuto utilizzando algoritmi come il k-means.

2. fissare dei valori iniziali per π_j: $\pi_j^* = \frac{N_j}{n}\ j = 1, \ldots, k$

3. utilizzare la massima verosimiglianza per stimare i parametri θ_j delle distribuzioni marginali $f_j\left(x_{i,j};\theta_j\right)$ dove $x_{i,j} \in S_j\ j = 1, \ldots, k$.

In molte situazioni un modello mistura che utilizza ditribuzioni multivariate note non risulta essere appropiato, ad esempio nei casi in cui le variabili osservate hanno supporti differenti, capita spesso infatti che alcune variabili osservate abbiano un supporto molto limitato (come variabili percentuali) ed altre siano definite su tutto l'asse dei reali.
È possibile in questi casi costuire distribuzioni multivariate "su misura" attraverso l'utilizzo delle copule, per approfondimenti su tale metodologia si veda Kosmidis and Karlis (2016)

Capitolo 2

Stima del numero di gruppi

Stimare il numero corretto di gruppi in un dataset è un aspetto cruciale per questo tipo di analisi e non sempre risulta semplice individuare una soluzione. Ad oggi, infatti, non esiste una metodologia condivisa per poter prendere questa decisione, ed è per questo motivo che l'individuazione della partizione ottimale è stata per anni, e lo è tutt'ora, uno dei problemi di ricerca ancora aperti della Cluster Analysis. In particolare gli errori decisionali che si possono commettere sono sostanzialmente due. Il primo si verifica quando si conclude che nei dati ci sono k gruppi, ma in realtà il numero di gruppi è inferiore a k, pertanto si ottiene una soluzione contenente troppi cluster. Il secondo tipo di errore si verifica quando si decide per un numero di gruppi inferiore a quelli effettivamente presenti nei dati, ottenendo così una partizione con pochi cluster. Sebbene la gravità dei due tipi di errore cambi a seconda del contesto applicativo, considerare erroneamente un numero troppo esiguo di gruppi comporta una perdita di informazione, derivante dall'unione di gruppi distinti.
In letteratura sono tradizionalmente individuati due tipologie di indici per valutare la bontà del *Clustering*:

- indici esterni, che misurano quanto i *cluster* individuati corrispondono a etichette di classe fornite esternamente (conoscenza pregressa);
- indici interni, che misurano quanto una soluzione di *Clustering* si adatta bene ai dati, quando i dati sono la sola informazione disponibile.

2.1 Indici esterni

Gli indici esterni richiedono la conoscenza del vero numero di gruppi e della loro composizione (non hanno quindi la finalità di stimare il numero di gruppi ma di determinare la bontà di una procedura di clustering). Se si hanno a disposizione le etichette di classe, si esegue il *Clustering* per comparare i risultati provenienti dall'applicazione di diversi algoritmi, con

l'obiettivo di individuare l'algoritmo ottimale per uno specifico dataset.
Tali indici determinano una misura di accordo tra due partizioni: la prima $U = \{u_1, \ldots, u_r\}$ è una struttura di dati prespecificata, mentre la seconda $V = \{v_1, \ldots, v_c\}$ è il risultato di una procedura di clustering. Esistono due tipologie di indici esterni:

- indici classification-oriented
- indici similarity-oriented

2.1.1 Indici classification-oriented

Gli indici *classification-oriented* sono stati inizialmente proposti per valutare la performance di classificatori nell'ambito della classificazione supervisionata per misurare il grado di concordanza tra le etichette di classe predette e la classificazione reale dei dati. Tuttavia se si utilizza una soluzione di *Clustering* l'uso che si fa di questi indicatori sostanzialmente non cambia.
Uno degli indici *classification-oriented* più utilizzato è l'F-measure.
L'**F-measure** combina due misure: la *precision* e la *recall*.
La *precision* è la proporzione di oggetti nel *cluster i* che appartengono ad una specifica classe. La *precision* del *cluster i* rispetto ad una generica classe j è:

$$precision(i, j) = p_{i,j} = \frac{n_{i,j}}{n_i}$$

La *recall* valuta in che misura un *cluster* contiene oggetti di una specifica classe ed è calcolata come rapporto tra il numero di oggetti nel *cluster i* che appartengono alla classe j e il numero di oggetti nella classe j:

$$recall(i, j) = \frac{n_{i,j}}{n_j}$$

L'*F-measure* è data dalla media armnonica della *precision* e della *recall*, può essere calcolata per ciascuno dei *cluster* in rapporto ad ognuna delle classi ed è così definita:

$$F(i, j) = \frac{2 \cdot precision(i, j) \cdot recall(i, j)}{precision(i, j) + recall(i, j)}$$

I valori assunti dall'indice variano tra 0 e 1, più sono prossimi ad 1, migliore risulta la corrispondenza tra i risultati del *Clustering* e la classificazione *a priori* e di conseguenza maggiore sarà la qualità della soluzione.
Tale indice può essere utilizzato per confrontare due o più procedure di clustering, attraverso i seguenti passaggi:

1. ad ogni cluster si associa un gruppo della classificazione a priori (si sceglie il gruppo più rappresentativo)
2. per ogni coppia cluster-gruppo a priori si calcola l'F-measure

3. si calcola l'F-measure media del clustering

è possibie ora confrontare le F-measure medie derivanti da più tecniche di clustering e secondo tale approccio la tecnica di clustering che risulterà migliore è quella con F-measure media più elevata.

2.1.2 Indici similarity-oriented

Gli indici similarity-oriented vengono calcolati attraverso una tabella di contingenza (tabella 2.1), dove $n_{i\cdot} = \sum_{j=1}^{C} n_{i,j}$ e $n_{\cdot j} = \sum_{i=1}^{R} n_{i,j}$ le somme per riga e per colonna di tale tabella, ossia il numero di osservazioni nei gruppi u_i e v_j, e $Z = \sum_{i=1}^{R}\sum_{j=1}^{C} n_{i,j}^2$.

	v_1	v_2	...	v_C	
u_1	$n_{1,1}$	$n_{1,2}$	...	$n_{1,C}$	$n_{1\cdot}$
u_2	$n_{2,1}$	$n_{2,2}$	...	$n_{2,C}$	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
u_R	$n_{R,1}$	$n_{R,1}$	...	$n_{R,1}$	$n_{R\cdot}$
	$n_{\cdot 1}$	$n_{\cdot 2}$	...	$n_{\cdot C}$	n

Tabella 2.1: tabella di contingenza per due partizioni di n osservazioni

Gli indici maggiormente utilizzati per calcolare la similarità tra due partizini sono:

- l'indice di Rand:

$$R = 1 + \frac{(Z - (1/2\sum_{i=1}^{R} n_{i\cdot}^2 + \sum_{j=1}^{C} n_{\cdot j}^2))}{\binom{n}{2}}$$

- l'indice di Jaccard:

$$J = (Z - n)/(\sum_{i=1}^{R} n_{i\cdot}^2 + \sum_{j=1}^{C} n_{\cdot j}^2 - Z - n)$$

- l'indice di Rand corretto. Tale indice proposto da Hubert and Arabie (1985) è una rivisitazione dell'indice di Rand e assume valore pari a 0 quando le partizioni sono determinate casualmente e 1 quando vi è perfetta corrispondenzatra esse. L'indice è così definito:

$$R_c = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{i,j}}{2} - [1/\binom{n}{2}]\sum_{i=1}^{R}\binom{n_{i\cdot}}{2}\sum_{j=1}^{C}\binom{n_{\cdot j}}{2}}{(1/2)[\sum_{i=1}^{R}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{C}\binom{n_{\cdot j}}{2}] - [1/\binom{n}{2}]\sum_{i=1}^{R}\binom{n_{i\cdot}}{2}\sum_{j=1}^{C}\binom{n_{\cdot j}}{2}}$$

2.2 Indici interni

Nella maggior parte delle applicazioni non sono disponibili informazioni riguardanti il vero numero e la composizione dei gruppi e gli indici interni rappresentano una soluzione per la validazione di una procedura di clustering.
Tali indici vengono spesso utilizzati per determinare il numero ottimale di gruppi, nelle sezioni che seguono ne saranno trattati alcuni tra i più utilizzati.
Per un dato partizionamento dei dati $g_1, \ldots, g_k$ siano:

$$W_k = \sum_{r=1}^{k} \sum_{x_i \in g_r} (x_i - \bar{x}_{g_r})(x_i - \bar{x}_{g_r})^T$$

$$B_k = \sum_{r=1}^{k} |g_r|(\bar{x}_{g_r} - \bar{x})(\bar{x}_{g_r} - \bar{x})^T$$

le matrici di devianza *within* e di devianza *between* rispettivamente, dove $\bar{x}_{g_r}$ e $\bar{x}$ denotano rispettivamente la media del gruppo r e la media del data set.
Tali due quantità saranno utilizzate in seguito per la costruzione di alcuni indici.

2.2.1 La silhouette

L'indice silhouette proposto da Rousseeuw (1987), si basa sulla coesione dell'unità su cui viene calcolata con il guppo di appartenenza e sulla sua separazione con gli altri gruppi. Tale indice può essere calcolato per ogni unità statistica.
La statistica silhouette per l'i-esima unità statistica è così definita:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

dove $a(i)$ è la distanza media dall'osservazione i da tutte le unità appartenenti al medesimo gruppo, mentre $b(i)$ è la più piccola tra le distanze medie dell'i-esima osservazione con le unità degli altri gruppi. Al variare di $a(i)$ e $b(i)$ si può scrivere:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & se\ a(i) < b(i) \\ 0, & se\ a(i) = b(i) \\ b(i)/a(i) - 1, & se\ a(i) > b(i) \end{cases}$$

L'indice silhouette varia tra -1 e 1, un $s(i)$ vicino a 1 significa che l'unità i è ben classificata, un $s(i)$ vicino a 0 implica che l'i-esima osservazione è in

una situazione borderline tra due gruppi, infine un $s(i)$ vicino a -1 implica che l'i-esima osservazione è classificata nel gruppo sbagliato.
La silhouette media S_k è definita come segue:

$$S_K = \frac{1}{n}\sum_{i=1}^{n} s(i)$$

dove k è il numero di gruppi.
La silhouette media è uno dei metodi più utilizzati per determinare il corretto numero di gruppi, il k ottimo sarà quello che massimizza S_k.
Si noti che la statistica silhouette, e di conseguenza la silhouette media, può essere calcolata per $2 \leq k \leq n$.

2.2.2 Indice di Calinski e Harabasz

Sia $tr(\mathbb{X}) = \sum_{i=1}^{n} \mathbb{X}_{ii}$ la traccia di una generica martice $\mathbb{X}$. L'indice di Calinski e Harabasz (Caliński and Harabasz, 1974) valuta la qualità della partizione ottenuta in termini di rapporto tra la varianza entro i gruppi e la varianza tra i gruppi ed è definito come segue:

$$CH = \frac{tr(B_k)/(k-1)}{tr(W_k)/(n-k)},$$

dove n e k sono rispettivamente il numero totale di osservazioni e il numero di cluster.
Si può facilmente notare che il numero di gruppi k che risulta ottimale secondo tale indice è quello che lo massimizza in quanto vi è a numeratore un *indicatore di separazione* e a denominatore un *un indicatore di coesione*. La quantità $\frac{n-k}{k-1}$ previene che l'indice aumenti in modo monotono al crescere del numero dei gruppi.

2.2.3 La statistica Gap

La statistica Gap si pone l'obiettivo di valutare, per ogni valore di k, la differenza tra la quantità $\log(trW_k)$ con il suo valore atteso sotto una distribuzione di riferimento specificata ed è così definita;

$$Gap(k) = E\left[\log tr(W_k)\right] - \log tr(W_k)$$

dove E denota il valore atteso sotto una distribuzione di probabilità specificata.
Per stimare il valore atteso di $\log tr(W_k)$ si generano B campioni sotto la distribuzione nulla specificata, ad ogni campione si applica l'algoritmo di clustering e si calcolano $trW_k^1, trW_k^2, \ldots, trW_k^B$. È possibile ora stimare la statistica Gap come segue;

$$\overline{Gap}(k) = \frac{1}{B}\sum_{b=1}^{B} \log tr(W_k^b) - \log tr(W_k).$$

Sia $sd(k)$ la deviazione standard di $\log\left(trW_k^1\right), \log\left(trW_k^2\right), \ldots, \log\left(trW_k^B\right)$ e $ss(k) = sd(k)\sqrt{1+1/B}$.
Il numero ottimale di gruppi viene definito come il minor valore di k tale per cui $\overline{Gap}(k) \geq \overline{Gap}(k+1) - ss(k+1)$. Tibshirani et al. (2001), propongono di calcolare $E\left[log(trW_k)\right]$ assumendo che le osservazioni provengano da una distrubuzione uniforme e due metodi per determinare il supporto della distribuzione.
Metodo GapUnif: il supporto viene identificato per ciascuna variabile dal range dei valori della stessa.
Metodo GapPC: in cui il supporto è allineato secondo le componenti principali del dataset centrato. In particolare, si supponga che la matrice X abbia vettore di medie nullo. Si determina la decomposizione a valori singolari $X = UDV^T$, dove U e V sono matrici ortogonali e D matrice dei valori singolari. Si consideri $X^{'} = XV$ e si genera $Z^{'}$ sulle colonne di X' come in 1). Infine si applica la trasformazione inversa per ottenere $Z = Z^{'}V^T$.

In entrambi i metodi le variabili sono campionate indipendentemente. GapUnif risulta essere vantaggioso per la semplicità applicativa mentre GapPC tiene in considerazione la forma della distribuzione dei dati.

2.2.4 L'indice Ray Turi

L'indice Ray Turi misura la bontà di una partizione come rapporto tra la compattezza e la separazione dei gruppi, ovvero

$$RT = \frac{\frac{1}{n}\sum_{j=1}^{k}\sum_{x\in g_j}\|x-\bar{x}_{g_j}\|^2}{\min_{i<j}\|\bar{x}_{g_i}-\bar{x}_{g_j}\|^2}$$

dove il numeratore fornisce una misura della compattezza dei cluster data dalla media dei quadrati delle distanze di ogni punto rispetto al centroide del cluster di appartenenza e il denominatore una misura della separazione tra cluster data dalla minima distanza quadratica tra i centroidi dei diversi gruppi.

Il valore di k che si ottiene minimizzando l'indice corrisponde al numero di gruppi ottimale secondo tale procedura, l'idea di fondo è quella di minimizzare la distanza entro i cluster (misurata dal numeratore dell' indice) e massimizzare, al tempo stesso, la distanza tra i cluster (misurata dal denominatore dell'indice). Tale indice è stato presentato come metodo di determinazione del k ottimo nella segmentazione di immagini, la sua semplicità interpretativa ci permette di estenderlo all'applicazione di una generica procedura di clustering, per approfondimenti si veda Ray and Turi (1999).

2.2.5 Metodo jump

Sugar and James (2003) definiscono una quantità che misura la distanza media per dimensione tra ogni unità e l'associato centroide di cluster, tale metodo si basa quindi sul concetto di "distorsione", ovvero una misura di dispersione interna ai cluster. Assumendo che $\mathbb{X}$ sia generato da una mistura di distribuzioni di G componenti, ognuna delle quali con matrice di covarianza Γ e sia $\bar{x}_{g_1}, \ldots, \bar{x}_{g_k}$ un set di possibili centroidi associati ai cluster $g_1, \ldots, g_k$, la distorsione, per ogni $k \geq 1$, è

$$d_k = \frac{1}{p}\left[\frac{1}{n}\sum_{j=1}^{k}\sum_{x_i \in C_j}\left(x_i - c_j\right)^T \Gamma^{-1}\left(x_i - c_j\right)\right];$$

Questa è pari alla distanza media di Mahalanobis per dimensione tra $\mathbb{X}$ e i centroidi $\bar{x}_{g_1}, \ldots, \bar{x}_{g_k}$.
Si noti che, se Γ è la matrice identità, la distorsione non è altro che l'errore quadratico medio. Nell'implementazione di tale metodo Sugar and James (2003) utilizzano per convenienza $\Gamma = I_p$.
Sugar and James (2003) dimostrano, sia teoricamente che empiricamente, che per una vasta classe di distribuzuini la curva di distorsione, se trasformata attraverso un'appropiata potenza negativa m (un valore tipico utilizzato è $m = -p/2$) subisce un brusco salto al numero "vero" di cluster.
Il metodo jump può quindi essere riassunto nei seguenti passi:

1. applicare un algoritmo di clustering (Sugar and James (2003) utilizzano il k-means) per differenti valori di k e calcolare la corrispondente distorsione $\hat{d}_k$;

2. selezionare un'appropiata potenza per trasformare la distorsione, tipicamente viene utilizzata una potenza $m = -p/2$;

3. calcolare i "salti" nella distorsione trasformata $J_k = \hat{d}_k^{-m} - \hat{d}_{k-1}^{m}$ $\hat{d}_0^m = 0$;

4. determinare il numero di cluster k^* nel data set mediante $k^* = argmax_k J_k$.

2.2.6 Cluster Stability

La Cluster Stability rappresenta un insieme di metodi di validazione interna di una procedura di clustering. Tali metodi infatti si basano unicamente sui dati a disposizione senza la possibilità di ricorrere a etichette esterne per valutare la bontà della partizione ottenuta e quindi possono essere utilizzate per stimare il vero ed ignoto numero di gruppi.
L'obiettivo di tali metodi è quello di testare la robustezza della procedura di clustering utilizzata mediante tecniche di ricampionamento dei dati. Tra i vari metodi proposti in letteratura ci occuperemo in questa sede della

prediction strength (Tibshirani and Walther, 2005).
L'idea che sta alla base della prediction strength si esplicita attraverso i seguenti passi;

1. suddividere casualmente il dataset $\mathbb{X}$ in 2 sottoinsiemi, assegnarne uno al set di training $\mathbb{X}_{tr}$ ed il restante al test-set $\mathbb{X}_{te}$;

2. applicare una soluzione di clustering sia al training set che al test set, operazioni denotate rispettivamente con $C(X_{tr}, k)$ e $C(X_{te}, k)$;

3. si utilizzano i centroidi determinati mediante il clustering sul training set per suddividere in gruppi le unità appartenenti al test set secondo il criterio tale per cui ogni unità viene assegnata al cluster il cui centroide ha minore distanza da essa;

4. si misura la bontà di previsione delle osservazioni del test set da parte dei centroidi del clustering sul training set. Ovvero si valuta per ogni coppia di osservazioni che vengono assegnate allo stesso gruppo nel clustering sul test set se vengono assegnate allo stesso gruppo anche nella partizione ottenuta tramite i centroidi del clustering sul training set mediante:

$$ps(b,k) = \min_{1\leq j\leq k} \frac{1}{n_{kj}(n_{kj}-1)} \sum_{i\neq i'\in A_{kj}} I(D\left[C(X_{tr},k), X_{te}\right]_{ii'} = 1)$$

   dove $A_{k1}, \ldots, A_{kk}$ denotano gli insiemi delle osservazioni nei cluster test $1, \ldots, k$, mentre $n_{k1} \ldots, n_{kk}$ sono il relativo numero di osservazioni. La funzione indicatrice $I(D\left[C(X_{tr},k), X_{te}\right]_{ii'} = 1)$ vale 1 se gli elementi di i e $i^{'}$ assegnati ad uno stesso cluster tramite $C(X_{te}, k)$ appartengono allo stesso cluster anche attraverso la procedura di partizionamento basata sui centroidi ottenuti da $C(X_{tr}, k)$.

I passaggi (2-4) vengono iterati B volte in cui il test set ed il training set vengono ridefiniti ad ogni iterazione.
Si definisce quindi la prediction strength relativa a k gruppi come:

$$ps(k) = \frac{1}{r}\sum_{b=1}^{B} ps(b,k).$$

Si stima il numero di gruppi ottimale $\hat{k}$ come il più grande k tale che $ps(k) \geq 0.8$ o 0.9.

2.3 Inferenza

Come affermato nel capitolo 1 i modelli a mistura finita offrono l'importante vantaggio di poter applicare procedure inferenziali al fine di determinare il

corretto numero di gruppi. Inoltre i modelli a mistura finita offrono in questo cotesto la possibilità non solo di fare inferenza sul numero di gruppi ma anche di determinare simultaneamente la forma più appropiata delle componenti della mistura, un'alternativa che può essere testata è quella che una mistura di distribuzioni Student-t abbia un'adattamento migliore rispetto ad una mistura di Gaussiane.
Verranno considerate due procedure inferenziali per la selezione del numero di cluster e/o del modello (in questa sede ci occuperemo della sola determinazione del numero di gruppi assumendo una corretta spcificazione del modello a mistura finita di Gaussiane) quali;

1. Il test del log-rapporto di verosimiglinza (LRT)
2. I criteri di informazione automatica.

Altri approcci inferenziali possibili, che non verranno affrontati in questa sede sono i test mediante i fattori di Bayes.

2.3.1 Test del log-rapporto di verosimiglianza

Sia $X_1, \ldots, X_n$ un campione casuale di dimensione n con distribuzione proveniente da una mistura finita di Gaussiane $h(x; \vartheta)$.
L'obiettivo è quello di testare che la vera distribuzione dei dati derivi da una mistura di k_0 componenti piuttosto che k_1 componenti, formalmente il sistema di ipotesi è

$$\begin{cases} H_0: & k = k_0 \\ H_1: & k = k_1, k_1 > k_0. \end{cases}$$

Sia,

$$F_\theta \equiv \{F(x;\theta), \theta \in \Theta \subset \Re^p\}, G_\gamma \equiv \{G(x;\gamma), \gamma \in \Gamma \subset \Re^q\}$$

le famiglie delle distribuzioni a mistura finita di normali con rispettivamente k_1 e k_0 componenti, dove $\theta = (\pi_1, \mu_1, \sigma_1^2, \ldots \pi_{k_1}\mu_{k_1}, \sigma_{k_1}^2)$ di dimensione $p = 3k_1 - 1$ e $\mu_1 < \mu_2 \ldots < \mu_{k_1}$ e $\gamma = (\pi_1, \mu_1, \sigma_1^2, \ldots \pi_{k_0}\mu_{k_0}, \sigma_{k_0}^2)$ di dimensione $q = 3k_0 - 1$ e $\mu_1 < \mu_2 \ldots < \mu_{k_0}$ e siano

- $LR = LR(\theta, \gamma; x) = \sum_{i=1}^n \log \frac{f(X_i;\theta)}{g(X_i;\gamma)}$
- $A_f(\theta) = E_h \left\{ \frac{\partial^2 \log f(X;\theta)}{\partial\theta\partial\theta^T} \right\}$
- $A_g(\gamma) = E_h \left\{ \frac{\partial^2 \log gX;\gamma)}{\partial\gamma\partial\gamma^T} \right\}$
- $B_f(\theta) = E_h \left\{ \frac{\partial \log f(X;\theta)}{\partial\theta} \frac{\partial \log f(X;\theta)}{\partial\theta^T} \right\}$

- $B_g(\gamma) = E_h \left\{ \frac{\partial \log g(X;\gamma)}{\partial \gamma} \frac{\partial \log f(X;\gamma)}{\partial \gamma^T} \right\}$
- $B_{fg}(\theta,\gamma) = B'_{gf}(\theta,\gamma) = E_h \left\{ \frac{\partial \log f(X;\theta)}{\partial \theta} \frac{\partial \log f(X;\gamma)}{\partial \gamma^T} \right\}$

dove θ e γ vengono sostituiti con le rispettive stime di massima verosimiglianza $\hat{\theta}$ e $\hat{\gamma}$
Un'ottimo candidato per testare tale sistema di ipotesi è dato dalla statistica del rapporto di verosimiglianza.
Nel caso di modelli a mistura finita l'usuale distribuzione asintotica per $-2 \ln LR$ sotto H_0 non è soddisfatta in quanto non sono soddisfatte le usuali condizioni di regolarità del modello. Il problema è che la distribuzione nulla si trova sulla frontiera dello spazio parametrico nel senso che quando le due componenti coincidono le proporzioni della mistura diventano non identificabili, la conseguenza di ciò è che il test del log-rapporto di verosimiglianza tende a sovrastimare il numero di gruppi.
Numerosi autori hanno proposto delle soluzioni a tale problema, in seguito ne verranno riassunte alcune.
Wolfe (1971) suggerisce che sotto l'ipotesi nulla la distribuzione della statistica del log-rapporto di verosimiglianza riscalata per una costatnte $c = [n - 1 - p - 0.5(k + 1)]/n$ per una mistura di normali $p-$dimensionali per testare l'ipotesi nulla H_0 contro l'ipotesi alternativa H_1 è una chi-quadrato con $2\nu - 2$ gradi di libertà dove ν è pari alla differenza del numero di parametri nei due casi testati.
Lo et al. (2001) dimostrano che la statistica log-rapporto di verosimiglianza, sotto opportune condizioni di regolarità e sotto alcune assunzioni che non verranno discusse in questa sede, si distribuisce sotto l'ipotesi nulla come una somma di variabili casuali χ^2_1 indipendenti.

TEOREMA 1. Sotto opportune condizioni di regolarità e sotto l'ipotesi nulla la distribuzione asintotica di $2LR$ *è data da una somma pesata di* $p+q$ *variabili casuali* χ^2_1 *indipendenti, dove i pesi* $(\lambda_1 \dots \lambda_{p+q})$ *sono dati dagli autovalori della matrice*

$$W = \begin{bmatrix} -B_f(\hat{\theta})A_f^{-1}(\hat{\theta}) & -B_{fg}(\hat{\theta},\hat{\gamma})A_g^{-1}(\hat{\gamma}) \\ B_{gf}(\hat{\theta},\hat{\gamma})A_f^{-1}(\hat{\theta}) & -B_g(\hat{\gamma})A_g^{-1}(\hat{\gamma}) \end{bmatrix}.$$

Per la dimostrazione del teorema ed ulteriori approfondimenti si veda Lo et al. (2001).
Tale distribuzione asintotica può quindi essere utilizzata per fare inferenza sul numero di cluster.
Un'altro approccio basato sul test del log-rapporto di verosimiglianza è dato da una procedura bootstrap parametrica (McLachlan, 1987) in cui i campioni bootstrap vengono utilizzati per stimare la distribuzione empirica della statistica LRT sotto l'ipotesi nulla.

Vengono simulati B campioni di dimensione n da una mistura a k_0 componenti utilizzando i parametri stimati mediante la massima verosimiglianza, per ogni simulazione viene calcolata la statistica LRT dopo aver adattato ad ogni campione simulato un modello a k_0 e uno a k_1 componenti, in tal modo si ottiene una stima della distribuzione della statistica sotto l'ipotesi nulla.

2.3.2 Criteri di informazione

I criteri di informazione automatica forniscono un ulteriore approccio per la stima del numero di cluster. Tali criteri sono generalmente composti da due quantità, una prima che fornisce una misura della bontà di adattamento del modello ai dati ed una seconda quantità che costitusce una penalizzazione per la comlessità del modello. Si preferisce il modello che assume il valore inferiore del criterio scelto.
I due criteri di informazione più utilizzati sono l' AIC (Akaike information criterion) e il BIC (Bayesian information criterion).
Sia $\hat{l}_p$ la log-verosimiglianza valutata nel punto di ottimo e p il numero di parametri stimati dal modello.
L'AIC (Akaike, 1974) è così definito:

$$AIC = -2(\hat{l}_p - p),$$

Tale criterio risulta però essere inconsistente e tende a sovrastimare il reale numero di gruppi.
Il criterio BIC proposto da Schwarz (1978) deriva da un'approccio inferenziale Bayesiano ma può essere applicato anche nell'ambito dell'inferenza frequentista ed è cosi definito;

$$BIC = -2\hat{l}_p + p\ln(n),$$

dove n è la numerosità campionaria.

Capitolo 3

Simulazioni

In questo capitolo si valutano le performance dei diversi criteri presentati nel capitolo 2, in particolare si vogliono confrontare le performance di due gruppi di criteri, gli indici interni e l'inferenza statistica. Nei metodi di inferenza statistica si è scelta una soglia del rifiuto del test $\alpha = 0.05$.
Sono stati scelti scenari di simulazione presenti in letteratura, in particolare tra gli 8 scenari individuati, i primi cinque sono stati presentati da Tibshirani et al. (2001), mentre il sesto da Sugar and James (2003) ed infine il settimo e ottavo riprendono il quinto e sesto ma con numerosità campionarie più elevate.
Per ogni scenario sono stati simulati 100 campioni. Negli ultimi due scenari la scelta di genereare campioni di numerosità più elevata è giustificata dal fatto che nella realtà si hanno a disposizione sempre più dati. Con i moderni database risulta ormai difficile imbattersi in campioni ridotti, si intende quindi valutare come si comportano i vari metodi quando le numerosità campionarie aumentano.
Tutti i metodi e gli scenari individuati sono stati implementati mediante il linguaggio di programmazione R, versione 4.1.3.
Per ogni scenario di simulazione, viene presentata una tabella contenente i risultati ottenuti nei vari metodi.
Nel determinare le performance degli indici interni si è utilizzato l'algoritmo di partizionamento k-means, in quanto uno dei più utilizzati a livello pratico. Per evitare la convergenza a ottimi locali l'algoritmo è stato ripetuto per 15 volte e si è scelto il partizionamento dei dati in modo tale da minimizzare la devianza interna ai gruppi. Mentre, per determinare le perfomance dell'inferenza statistica nell'individuare il numero ottimale di gruppi, si è scelto l'utilizzo del modello a mistura finita di gaussiane, il modello più utilizzato nel model-based clustering.
Il clustering basato sul modello a mistura di Gaussiane è implementato nella funzione `Mclust` del pacchetto `mclust` (Fraley et al., 2012). Tale funzione fornisce diverse specificazioni per le matrici di covarianza della mistura, tra

le diverse scelte si è preferita una specificazione flessibile quale la *VEV* che permette matrici di covarianza differenti per ogni gruppo e distribuzione marginale ellittica con orientamento variabile per gruppo.
Nel bootstrap LRT si è fissato $B = 999$ mentre per la prediction strength e la statistica gap $B = 100$ per tutte le simulazioni fatte.

3.1 Scenario 1: k=1, p=10

In questo scenario di simulzione sono stati generati 100 campioni multivariati con $p = 10$ e numerosità $n = 200$, le osservazioni di ogni campione seguono una distribuzione uniforme sull'ipercubo di lato unitario.

k	1	2	3	4	5	6	$\geq$7
Silhouette	NA	1	0	0	0	0	99
Calinski e Harabasz	NA	100	0	0	0	0	0
Ray Turi	NA	0	0	0	0	0	100
Jump	0	0	0	0	0	0	100
Gap Unif	84	16	0	0	0	0	0
Gap Pc	100	0	0	0	0	0	0
Prediction Strength	100	0	0	0	0	0	0
Lo-Mendell-Rubin LRT	7	15	10	19	17	8	24
Bootstrap LRT	95	5	0	0	0	0	0
BIC	100	0	0	0	0	0	0

Tabella 3.1: Risultati nel scenario 1.

I primi tre indici in tabella 3.1 sono calcolabili per $k \geq 2$. Mentre la silhouette media e l'indice di Rai-Turi suggeriscono un numero di cluster molto elevato in quasi tutti i campioni simulati, l'indice di Calinski Harabasz risulta essere coerente in quanto suggerisce il numero di gruppi più piccolo possibile in tutti e 100 i campioni.
Tra gli indici interni che possono essere calcolati anche per un solo gruppo, il metodo jump suggerisce una numerosità dei gruppi molto elevata e risulta pertanto essere inadatto in questo scenario, il metodo GapPC e la prediction strength performano molto bene metre il metodo GapUnif centra l'obiettivo 84 volte su 100.
Tra i metodi inferenziali il criterio di informazione BIC suggerisce sempre la numerosità corretta, il test LRT risulta fortemente inadeguato in quanto suggeisce numerosità molto variabili da campione a campione mentre il test bootstrap LRT rifiuta l'ipotesi nulla della presenza di un solo gruppo in favore dell'ipotesi alternativa di presenza di due gruppi nei dati 5 volte su 100, in linea con il livello di significatività fissato per il rifiuto.

3.2 Scenario 2: k=2, p=3

In questo scenario entrambi i cluster contengono 100 osservazioni e sono generati come segue:

1. si costruiscono x_1, x_2, x_3 righe equispaziate di 100 valori nell'intervallo $[-0.5, 0.5]$,
2. a x_1, x_2, x_3 si aggiunge un rumore Gaussiano con deviazione standard pari a 0.1;
3. si genera il secondo cluster seguendo i passi 1-2 ma nell'intervallo $[9.5, 10.5]$

k	1	2	3	4	5	6	≥7
Silhouette	NA	100	0	0	0	0	0
Calinski e Harabasz	NA	100	0	0	0	0	0
Ray Turi	NA	100	0	0	0	0	0
Jump	0	100	0	0	0	0	0
Gap Unif	0	100	0	0	0	0	0
Gap Pc	0	100	0	0	0	0	0
Prediction Strength	0	100	0	0	0	0	0
Lo-Mendell-Rubin LRT	0	94	2	3	1	0	0
Bootstrap LRT	0	96	3	1	0	0	0
BIC	0	100	0	0	0	0	0

Tabella 3.3: Risultati nel scenario 2.

In questo scenario sono presenti due cluster ben separati, tutti i metodi infatti performano molto bene, si osserva in particolare come i due test statistici rifiutano la nulla (quando è vera) un numero di volte in linea con la significatività scelta per il test.

3.3 Scenario 3: k=2, p=3

Questo scenario di simulazione non si discosta particolarmente dal precedente, l'unica differenza si trova nella generazione del secondo cluster in quanto le variabli x_1, x_2, x_3 assumono valori equispaziati nell'intervallo $[0.5, 1.5]$.
In questa sezione viene inoltre simulato una variante di tale scenario per rendere "parzialmente" sovvrapposti i due cluster e non totalmente come presentato da Tibshirani et al. (2001), in particolare, in questa "variante", il secondo cluster viene generato nell'intervallo $[1.5, 2.5]$.
I due scenari simulati in questa sezione vengono definiti, per comodità, come scenario 3.1 e scenario 3.2.

k	1	2	3	4	5	6	≥7
Silhouette	NA	95	0	0	0	1	4
Calinski e Harabasz	NA	100	0	0	0	0	0
Ray Turi	NA	0	1	2	14	12	61
Jump	38	1	0	1	3	1	56
Gap Unif	31	69	0	0	0	0	0
Gap Pc	88	12	0	0	0	0	0
Prediction Strength	70	30	0	0	0	0	0
Lo-Mendell-Rubin LRT	84	12	3	1	0	0	0
Bootstrap LRT	93	7	0	0	0	0	0
BIC	100	0	0	0	0	0	0

Tabella 3.5: Risultati nel scenario 3.1.

In questo scenario i due cluster sono particolarmente sovrapposti (Figura 3.2) A e non tutti i metodi presentati riescono a cogliere la leggera distinzione che vi è tra i due gruppi.

Gli unici due metodi che riescono a cogliere la presenza di due gruppi sono la silhouette media e l'indice di Calinski e Harabasz. I metodi inferenziali non riescono a cogliere la distinzione tra i due gruppi e ciò non è un risultato molto sorprendente vista la forte sovrapposizione tra i due cluster.

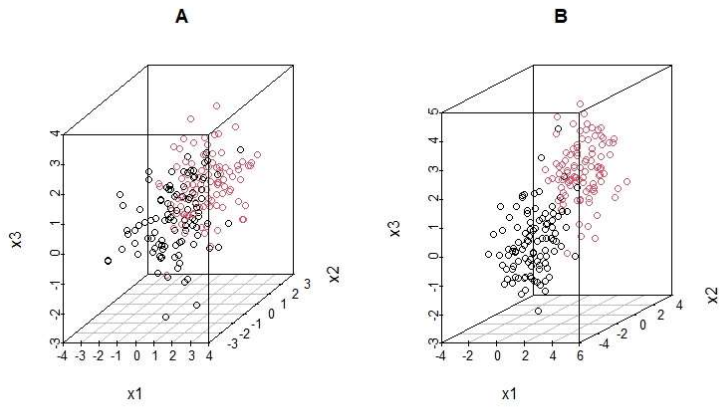


Figura 3.1: Simulazione di un campione dal scenario 3.1 e 3.2.

In Figura 3.2 B è rappresentata la generazione di un campione proveniente dallo scenario 3.2 in cui i due gruppi sono rappresentati mediante

due colori differenti, si può notare come i due cluster in questo caso risultino essere più distinti ma sempre leggermente sovrapposti.
In seguito si riportano i risultati dei metodi in questo scenario.

k	1	2	3	4	5	6	≥7
Silhouette	NA	100	0	0	0	0	0
Calinski e Harabasz	NA	100	0	0	0	0	0
Ray Turi	NA	100	0	0	0	0	0
Jump	0	98	2	0	0	0	0
Gap Unif	0	100	0	0	0	0	0
Gap Pc	5	95	0	0	0	0	0
Prediction Strength	2	98	0	0	0	0	0
Lo-Mendell-Rubin LRT	2	85	8	4	1	0	0
Bootstrap LRT	3	94	3	0	0	0	0
BIC	73	27	0	0	0	0	0

Tabella 3.7: Risultati nel scenario 3.2.

Gli indici interni performano molto bene, per quanto riguarda i metodi basati sull'inferenza statistica invece il criterio di informazione automatica BIC sottostima fortemente il numero reale dei gruppi, il test LRT tende a sovrastimare il numero di gruppi per 13 campioni su 100 estratti infine il test LRT bootsrap fornisce dei buoni risultati.

3.4 Scenario 4: k=3, p=2

In questo scenario vengono generati 100 campioni da una distribuzione mistura di Gaussiane standard bidimensionali centrate in $(0,0),(0,5),(-5,3)$. I campioni simulati proposti da Tibshirani et al. (2001) pevedono che i 3 gruppi fossero di numerosità pari a 25, 25 e 50, per rendere il campione effettivamente (pseudo-)casuale ho deciso di assegnare una probabilità $p = (0.25, 0.25, 0.5)$ di generare da ogni componente della mistura, così facendo ogni gruppo non avrà dimensione effettiva pari a 25, 25 e 50 ma varierà da campione a campione casualmente. Per fare ciò viene generata per ogni unità del campione una variabile casuale Y con supporto $S = (1, 2, 3)$ dove $P(Y = 1) = P(Y = 2) = 0.25$ e $P(Y = 3) = 0.5$, ad ogni valore del supporto di Y si associa uno specifico cluster, quindi per ogni valore generato dal supporto di Y viene generato casualmente un valore dallo specifico cluster

assegnatogli. Si è deciso inoltre di generare campioni di numerosità $n = 200$.

k	1	2	3	4	5	6	≥ 7
Silhouette	NA	0	100	0	0	0	0
Calinski e Harabasz	NA	0	99	1	0	0	0
Ray Turi	NA	0	100	0	0	0	0
Jump	0	0	97	0	0	0	3
Gap Unif	2	0	98	0	0	0	0
Gap Pc	0	0	100	0	0	0	0
Prediction Strength	2	29	69	0	0	0	0
Lo-Mendell-Rubin LRT	0	0	100	0	0	0	0
Bootstrap LRT	0	0	94	6	0	0	0
BIC	0	0	100	0	0	0	0

Tabella 3.9: Risultati nel scenario 4.

In questo scenario tutti i metodi performano molto bene, ad eccezione della prediction strength che individua il corretto numero di gruppi per 69 campioni su 100.

3.5 Scenario 5: k=4, p=2

In questo scenario vengono simulati 100 campioni da una misutura di 4 densità normali bivariate con la stessa metodologia presentata nella sezione pecedente, il vettore delle probabilità di appartenenza ai gruppi è $p = (0.25, 0.25, 0.25, 0.25)$, il vettore delle medie $\mu_1 = (0, 0), \mu_2 = (0, 2.5), \mu_3 = (2.5, 0), \mu_4 = (2.5, 2.5)$, la numerosità campionaria $n = 200$ e covarianze tutte uguali alla matrice identità.

k	1	2	3	4	5	6	≥ 7
Silhouette	NA	1	29	53	4	5	8
Calinski e Harabasz	NA	0	1	73	2	1	23
Ray Turi	NA	0	15	57	0	6	22
Jump	17	0	4	51	0	1	27
Gap Unif	100	0	0	0	0	0	0
Gap Pc	100	0	0	0	0	0	0
Prediction Strength	100	0	0	0	0	0	0
Lo-Mendell-Rubin LRT	96	3	0	1	0	0	0
Bootstrap LRT	96	3	0	1	0	0	0
BIC	99	0	0	1	0	0	0

Tabella 3.11: Risultati nel scenario 5.

L'indice di Calinski e Harabasz individua il corretto numero di gruppi nel %73 dei campioni simulati e risulta essere il criterio con migliore performance in questo scenario, i metodi inferenziali individuano nei dati un solo gruppo.

3.6 Scenario 6: k=5, p=2

In questo scenario vengono simulati 100 campioni da una misutura di 5 densità normali bivariate con la stessa metodologia presentata nel scenario 4, il vettore delle probabilità di appartenenza ai gruppi è $p = (0.20, 0.20, 0.20, 0.20, 0.20)$, il vettore delle medie $\mu_1 = (0, 0), \mu_2 = (2.5, 2.5), \mu_3 = (5, 5), \mu_4 = (-2.5, -2.5), \mu_4 = (-5, -5)$, la numerosità campionaria $n = 200$ e covarianze tutte uguali alla matrice identità.

k	1	2	3	4	5	6	≥ 7
Silhouette	NA	56	40	4	0	0	0
Calinski e Harabasz	NA	0	0	1	99	0	0
Ray Turi	NA	65	33	0	2	0	0
Jump	0	0	4	4	92	0	0
Gap Unif	0	0	86	0	14	0	0
Gap Pc	87	0	0	0	13	0	0
Prediction Strength	15	1	84	0	0	0	0
Lo-Mendell-Rubin LRT	0	0	5	10	74	9	2
Bootstrap LRT	0	0	7	15	76	2	0
BIC	0	0	33	33	34	0	0

Tabella 3.13: Risultati nel scenario 6.

In questo scenario hanno una buona performance il metodo jump e l'indice di Calinski e Harabasz mentre gli altri indici interni tendono a sottostimare il reale numero di gruppi. Tra i metodi inferenziali i due test basati sul rapporto di verosimiglianza individuano il corretto numero di gruppi circa per il 75% dei campioni mentre il criterio BIC solo nel 33% dei campioni.

3.7 Scenario 7: k=4, p=2

Questo scenario riprende il quinto in cui vengono generati campioni da una mistura di 4 distribuzioni Gaussiane con l'unica differenza che in questo caso vengono generati campioni di numerosità $n = 1000$.

k	1	2	3	4	5	6	≥7
Silhouette	NA	0	3	97	0	0	0
Calinski e Harabasz	NA	0	0	100	0	0	0
Ray Turi	NA	0	0	100	0	0	0
Jump	17	0	0	84	0	0	0
Gap Unif	100	0	0	0	0	0	0
Gap Pc	100	0	0	0	0	0	0
Prediction Strength	71	0	0	29	0	0	0
Lo-Mendell-Rubin LRT	94	5	0	1	0	0	0
Bootstrap LRT	92	7	0	1	0	0	0
BIC	99	0	0	1	0	0	0

Tabella 3.15: Risultati nel scenario 7.

Mentre nei metodi inferenziali non vi sono differenze nelle performance rispetto a quelle ottenute nel scenario 5, negli indici interni si notano dei notevoli miglioramenti, in particolare i metodi jump, silhouette media, l'indice di Ray Turi e quello di Calinski e Harabasz subiscono un notevole miglioramento nell'individuazione del numero di gruppi quando le numerosità dei campioni aumentano da $n = 200$ a $n = 1000$.

3.8 Scenario 8: k=5, p=2

Questo scenario riprende il sesto in cui vengono generati campioni da una mistura di 5 distribuzioni Gaussiane con l'unica differenza che in questo caso vengono generati campioni di numerosità $n = 1000$.

k	1	2	3	4	5	6	≥7
Silhouette	NA	83	17	0	0	0	0
Calinski e Harabasz	NA	0	0	0	100	0	0
Ray Turi	NA	94	6	0	0	0	0
Jump	0	0	0	0	100	0	0
Gap Unif	89	11	0	0	0	0	0
Gap Pc	89	11	0	0	0	0	0
Prediction Strength	96	3	1	0	0	0	0
Lo-Mendell-Rubin LRT	0	0	0	0	97	3	0
Bootstrap LRT	0	0	0	0	96	4	0
BIC	0	0	0	0	99	1	0

Tabella 3.17: Risultati nel scenario 8.

In questo scenario, a differenza del precedente, l'aumento delle numerosità campionarie ha portato ad un notevole miglioramento dei risultati delle procedure inferenziali che individuano per quasi tutti i campioni simulati la corretta struttura dei dati mentre ha lasciato inalterati i riusultati prodotti dagli indici interni.

3.9 Riepilogo risultati

La tabella 3.19 riporta un riassunto delle performance dei diversi metodi. Per ogni scenario viene riportata la percentuale di volte in cui ogni metodo individua il corretto numero di gruppi. Si noti che, nel primo scenario, i primi tre metodi non sono in grado di determinare il corretto numero di gruppi per costruzione, ma non viene preso in considerazione questo fattore nel calcolo della percentuale media di individuazione del corretto numero di cluster in quanto nella relatà non si è a conoscenza del fatto che vi sia un solo gruppo.

Scenario	1	2	3.1	3.2	4	5	6	7	8	Media
Silhouette	0	100	95	100	100	53	0	97	0	60.56
Calinski e Harabasz	0	100	100	100	99	73	99	100	100	85.67
Ray Turi	0	100	0	100	100	57	2	100	0	51
Jump	0	100	1	98	97	51	92	84	100	69.22
Gap Unif	84	100	69	100	98	0	14	0	0	51.67
Gap Pc	100	100	12	95	100	0	13	0	0	46.67
Prediction Strength	100	100	30	98	69	0	0	29	0	47.33
Lo-Mendell-Rubin LRT	7	94	12	85	100	1	74	1	97	52.33
Bootstrap LRT	95	96	7	94	94	1	76	1	96	62.22
BIC	100	100	0	27	100	1	34	1	99	51.33

Tabella 3.19: Riepilogo risultati delle simulazioni.

Capitolo 4

Analisi dataset reali

In questo capitolo si valutano le performance dei metodi presentati su due dataset reali, con particolare attenzione all'indice di Calinski e Harabasz in quanto dalle simulazoni è emerso essere il metodo con performance migliori.

4.1 Dataset iris

Il dataset iris (Fisher, 1936), contiene 150 ossevazioni su cui sono state rilevate 4 variabili (lunghezza, larghezza dei sepali e lunghezza, larghezza dei petali in centimetri), le unità appartengono 3 gruppi; setosa, versicolor e verginica. In tale dataset, uno dei gruppi (setosa) è ben distinto dagli altri due che sono invece molto vicini, anche se non completamente sovvrapposti. In questa analisi si è deciso di non standardizzare il dataset in quanto le quattro variabili presentano la stessa unità di misura.
Nella tabella 4.1 viene riportata la stima di k attraverso i metodi presentati.

	Silhouette	Calinski e Harabasz	Ray Turi	Jump	Gap Unif
k	2	3	7	7	6
	Gap Pc	Prediction Strength	Lo-Mendell-Rubin LRT	Bootstrap LRT	BIC
k	5	2	3	3	2

Tabella 4.1: Stima di k nel dataset Iris.

L'indice di Calinski e Harabasz stima in modo corretto il numero di gruppi, tutti gli altri indici interni ad eccezzione della silhouette media stimano un numero di gruppi incoerente con i dati osservati. La silhouette media, il cui andamento al variare di k è mostrato in Figura 4.1, non stima in modo corretto il numero di cluster tuttavia i gruppi versicolor e virginica risultano essere molto vicini tra loro. Tra i metodi inferenziali si osserva una corretta

stima per i metodi basati sul test del rapporto di verosimiglianza mentre il BIC accorpa insime due gruppi.
In questo caso, come si vede dal confronto delle tabelle 3.21 e 3.22 (che mostrano le matrici di confusione per il partizionamento k-means e mistura di Gaussiane rispettivamente), il partizionamento basato sul modello mistura di Gaussiane risulta essere più performante rispetto all'algoritmo $k-means$.

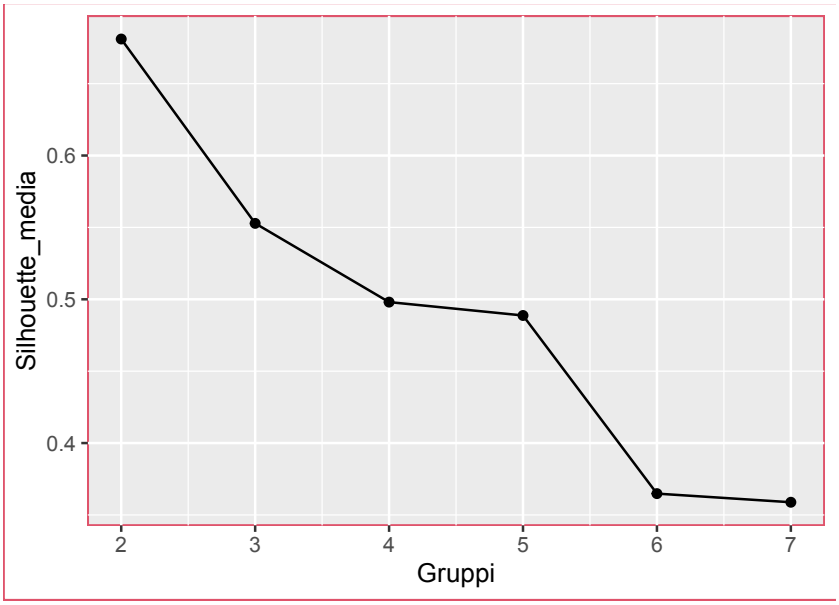


Figura 4.1: Andamento della silhouette media al variare di k.

	Veri gruppi		
Predizioni	1	2	3
1	48	14	0
2	2	36	0
3	0	0	50

Tabella 4.2: Matrice di confusione nel partizionamento k-means.

	Veri gruppi		
Predizioni	1	2	3
1	50	0	0
2	0	45	0
3	0	5	50

Tabella 4.3: Matrice di confusione nel partizionamento a mistura di Gaussiane.

4.2 Classificazione di cellule tumorali tramite spettri SERRS

Il dataset si riferisce ad uno studio sulla classificazione cellule tumorali mediante spettri SERRS e contiene 1149 variabili. Di queste la prima si riferisce all'area delle cellule e le rimanenti si riferiscono agli spettri SERRS. Il campione osservato ha umerosità $n = 3189$ ed è suddiviso in 3 tipologie di cellule (LNCaP, PBMC, U251); per ulteriori approfondimenti si veda Litti et al. (2020). Nel campione sono presenti oltre alle 3 tipologie di cellule alcuni corpi estranei che presentano valori estremi, per questo motivo per ogni variabile, i valori maggiori del percentile di ordine 0.975 e minori del percentile di ordine 0.025 sono stati trattati come *valori anomali* e sono stati sostituiti con i relativi percentili.
In un dataset di tali dimensioni risulta impraticabile partizionare i dati in gruppi mediante un modello a mistura di gaussiane in quanto il numero di parametri da stimare risulta essere molto elevato. Infatti se si adatta ai dati una distribuzione normale con un solo gruppo a tale dataset i parametri da stimare risultano essere $p + p * (p + 1)/2 = 661822$. Per tale motivo si è scelto di utilizzare due insiemi di variabili; l'area e le prime 4 componenti principali (sul dataset standardizzato) degli spettri SERRS, si è scelto di non includere l'area nell'analisi delle componenti principali in quanto rappresenta la variabile più importante al fine della classificazione delle cellule. La proporzione di variabilità spiegata dalle prime 4 componenti principali è 0.42. In Figura 4.2 è rappresentata la proporzione di variabilità spiegata dalle prime 20 componenti principali, dove si può notare che le componenti principali che seguono la quarta apportano un incremento ridotto di proporzione di variabilità spiegata.
Nella tabella che segue è riportata la stima di k con i vari metodi

	Silhouette	Calinski e Harabasz	Ray Turi	Jump	Gap Unif
k	2	5	7	7	1

	Gap Pc	Prediction Strength	Lo-Mendell-Rubin LRT	Bootstrap LRT	BIC
k	1	2	6	> 7	6

Tabella 4.4: Stima di k con l'ausilio della PCA.

Tra gli indici interni i più coerenti sono la silhouette media e la prediction Strength in quanto due dei tre gruppi risultano essere molto vicini tra loro. Come si può vedere in Figura 4.3, la silhuette media non assume valori molto differenti in una soluzione di clustering a due o a tre gruppi, non siamo però in grado di valutare se questa differenza sia significativa.
I tre metodi inferenziali risultano essere inappropiati in quanto sovrastimano il numero di gruppi, si osservi che tale risultato potrebbe essere influenzato dalla scelta della mistura di distribuzioni di probabilità utilizata.

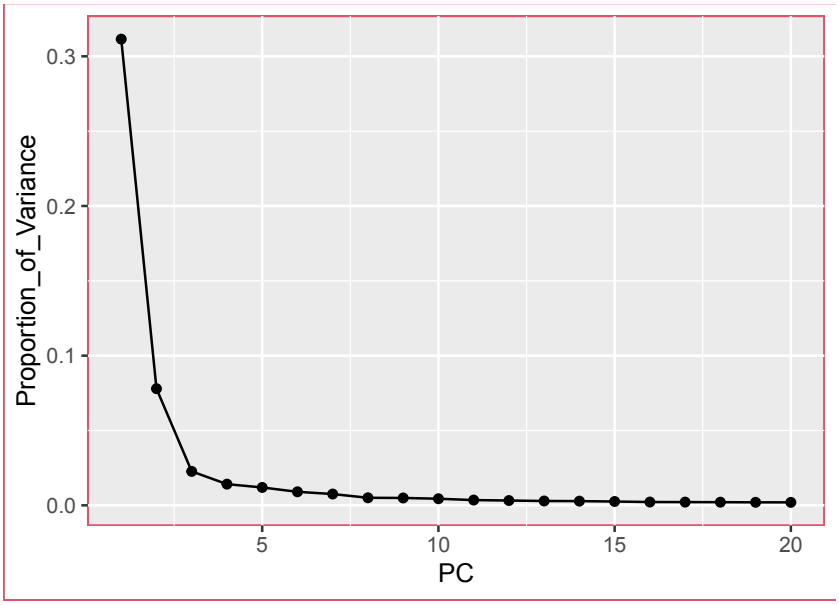


Figura 4.2: Proporzione di variabilità spiegata dalle prime 20 componenti principali

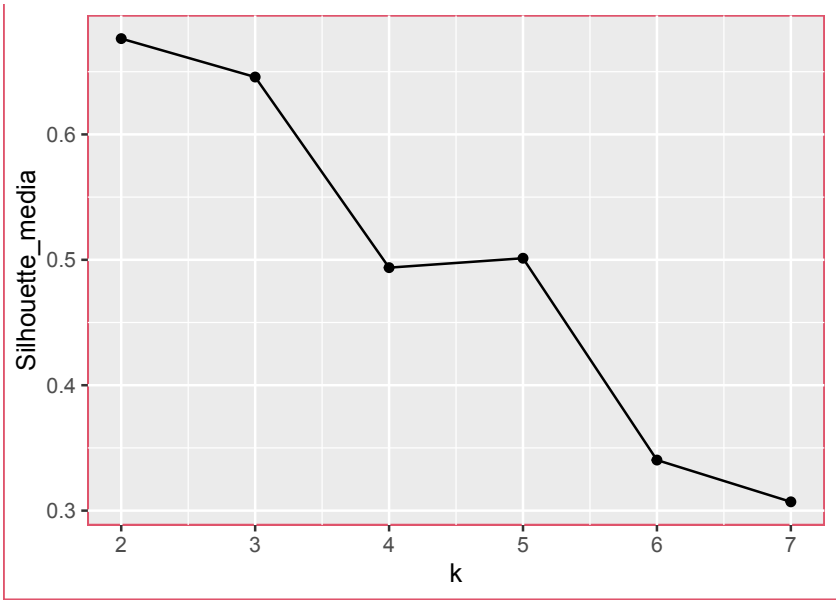


Figura 4.3: Silhouette media al variare di k.

L'algoritmo k-means è molto efficiente e può essere facilmente utilizzato quando si ha a disposizione un dataset di elevata dimensionalità. Si è scelto di stimare k utilizzando tutte le variabili (trattando i dati anomali come in precedenza) mediante l'indice Ray Turi, la silhouette media e l'indice di Calinski Harabasz in quanto i più efficienti.

	Silhouette	Calinski e Harabasz	Ray Turi
k	2	3	7

Tabella 4.5: Stima di k nel dataset completo.

Utilizzando tutte le variabili originali del dataset la silhouette media e l'indice di Ray Turi rimangono invariati mentre l'indice di Calinski e Harabasz stima ora il k corretto (Tabella 4.5).
Questo miglioramento evidenza i limiti dei metodi inferenziali che non possono essere utilizzati con dataset ad elevata dimensionalità.
Infine si è condotta un'analisi del dataset completo non trattando i valori anamali e si è osservato che l'indice di Calinski e Harabasz risulta essere sensibile ai valori anomali in quanto stima $k = 5$, la silhouette media e l'indice di Ray Turi rimangono invariati e producono rispettivamente le stime $k = 2$ e $k = 7$.

Conclusioni

L'obiettivo di questo lavoro è stato quello di individuare e confrontare alcuni metodi di stima del numero di gruppi nel clustering, in particolare si è voluto confrontare le performance degli indici interni con quelle delle procedure inferenziali.
Nonostante i numerosi metodi presenti in letteratura ad ora non vi è una metodologia affermata per individuare con precisione il corretto numero di cluster.
I concetti alla base dell'analisi di raggruppamento sono la coesione e la separazione tra le unità statistiche e tra i gruppi. Il problema che sorge spontaneo è se si può veramente defnire se e quando due gruppi siano realmente distinti tra loro. Mentre gli indici interni cercano di rispondere a tale quesito in termini di distanza tra i gruppi (e tra le unità all'interno dei gruppi) i metodi inferenziali invece partizionano il dataset iniziale in due o più gruppi, quando tale partizionamento apporta informazione aggiuntiva, ovvero quando i gruppi risultano essere statisticamente distinti tra loro.
Come si è visto dallo studio di simulazione, nessuno dei metodi presentati riesce ad individuare con un elevato livello di precisione il corretto numero di gruppi. In particolare non si è vista una notevole differenza nelle performance tra gli indici interni ed i metodi inferenziali, fatta eccezione per l'indice di Calinski e Harabasz.
Quest'ultimo è quello che ha prodotto risultati migliori negli scenari simulati individuando il corretto numero di cluster per l'85.67% dei campioni simulati. A seguire vi sono l'indice silhouette media e il test del log-rapporto di verosimiglianza bootstrap che individuano il corretto numero di gruppi aprossimativamente nel 60% dei campioni simulati. Tutti gli altri metodi hanno prodotto performance attorno al 50%.
L'indice di Calinski e Harabasz presenta però il difetto di non permettere di individuare la presenza di un unico cluster nei dati, è necessario quindi individuare una procedura per escludere la presenza di un unico cluster per poi procedere all'utilizzo di tale indice. Si pò notare dalla prima simulazione che quando nei dati vi è la presenza di un solo gruppo tale indice individua sempre due gruppi. Il problema si riduce quindi nel riuscire a discriminare, in tutti i casi in cui l'indice individua due gruppi, quando il numero reale ed ignoto di gruppi è realmente pari a due e quando è invece pari a uno. Si nota

inoltre che qundo il reale numero di cluster è pari a uno l'indice silhouette media individua (quasi) sempre numerosità molto elevate ($k >= 7$) mentre ha ottime performance quando il reale numero di cluster è due.
Si può quindi seguire la seguente procedura "automatica":

- si determina il k secondo l'indice di Calinski e Harabasz, se tale k stimato è maggiore di due lo si considera la stima ottimale del numero di gruppi;
- se $k = 2$ si determina il k stimato mediante lindice silhouette media, se riulta pari a due si assegna $\hat{k} = 2$ altrimenti si assegna $\hat{k} = 1$.

Tale metodologia risulta essere ottimale nelle simulazioni fatte e quindi sotto l'assunzione che le distrubuzioni marginali dei gruppi siano ellittiche, nella realtà è sempre bene affiancare tale approccio con un un'analisi grafica, può essere utile visualizzare i grafici delle variabili a tre a tre (come in figura 3.3).
Dall'analisi del dataset sulle cellule tumorali si evidenzia che l'indice di Calinski e Harabasz sembra essere sensibile alla presenza di valori anomali, questo risultato può essere un punto di partenza per ulteriori ricerche sulla robustezza dei metodi visti in presenza di dataset contenenti valori anomali. I metodi inferenziali, seppur dalle simulazioni sono risultati essere meno performanti dell indice di Calinski e Harabasz, non sono da escludere nella stima del numero di cluster. In particolare, il metodo bootstrap LRT funziona molto bene quando i gruppi risultano essere ben distinti, in quanto non tende a sovrastimare il numero di cluster (partizionando i gruppi un numero di volte eccessivo) ma solamente a sottostimarlo quando i cluster risultano essere molto vicini (se non sovrapposti). Come si è visto nella classificazione delle cellule tumorali tali metodi hanno la limitazione di non poter essere utilizzati quando si hanno a disposizione un elevato numero di variabili, una soluzione a questo problema può essere data dai metodi di riduzione della dimensionalità (come ad esempio la PCA) che però producono una perdita dell'informazione a disposizione.

Appendice A

Codice R delle simulazioni

Il codice seguente riporta le funzioni utilizzate negli otto scenari di simulazione. In particolare la funzione "simfunkk" e la funzione "simfuncl" prendono in input il scenario dal quale si vuole simulare assieme ai parametri propri degli scenari e ritornano i risultati prodotti rispettivamente dagli indici interni e dai metodi inferenziali.

```
library(clusterSim)
library(tidyLPA)
library(mclust)
library(cluster)
library(clusterCrit)
library(cstab)
library(fpc)

#funzione per generare da una mistura v.c normali multivariate
rmmnd <- function(n, mu, sigma, p, d){
  #mu e sigma sono delle liste,
  #sigma deve essere la matrice di covarianza elevata
  #alla potenza 0.5
  label <- 1:length(p)
  xx=cut(runif(n),breaks=c(0,cumsum(p)),
         labels=label)
  m<- rep(0,d)
  s<- diag(d)
  y <- matrix(NA, n, d)
  for(i in 1:n){
    y[i,] <- mu[[xx[i]]]+sigma[[xx[i]]] %*%
      t(rmvnorm(1, mean = m, sigma = s))
  }
  return(y)
}

#funzione per simulare il secondo scenario, due cluster
#allungati in in tre dimensioni generati a partire da valori
```

```
#equispaziati
rtwoelon <- function(n){
  t <- seq(-0.5, 0.5, length=n)
  x1 <- x2 <- x3 <- t
  x1 <- x1 + rnorm(n, 0.1)
  x2 <- x2 + rnorm(n, 0.1)
  x3 <- x3 + rnorm(n, 0.1)
  c1 <- cbind(x1,x2,x3)
  t2 <- seq(9.5, 10.5, length=n)
  y1 <- y2 <- y3 <- t2
  y1 <- y1 + rnorm(n, 0.1)
  y2 <- y2 + rnorm(n, 0.1)
  y3 <- y3 + rnorm(n, 0.1)
  c2 <- cbind(y1,y2,y3)
  return(rbind(c1,c2))
}

#funzione per simulare  dal terzo scenario
rtwoclose_elon <- function(n){
  t <- seq(-0.5, 0.5, length=n)
  x1 <- x2 <- x3 <- t
  x1 <- x1 + rnorm(n, 0.1)
  x2 <- x2 + rnorm(n, 0.1)
  x3 <- x3 + rnorm(n, 0.1)
  c1 <- cbind(x1,x2,x3)
  t2 <- seq(.5, 1.5, length=n)
  y1 <- y2 <- y3 <- t2
  y1 <- y1 + rnorm(n, 0.1)
  y2 <- y2 + rnorm(n, 0.1)
  y3 <- y3 + rnorm(n, 0.1)
  c2 <- cbind(y1,y2,y3)
  return(rbind(c1,c2))
}

#la funzione che segue genera valori dai diversi scenari previsti
#e restituisce i risultati delle stime di k degli indici interni
simfunkk <- function(n=NULL, B=100, mu=NULL, sigma=NULL,
                   p=NULL, d=NULL, n.group=8, scenario=3){
 if(scenario!= 1 & scenario!= 2 & scenario!= 3) {
  for(j in 1:length(p)){
   a <- eigen(sigma[[j]])
   sigma[[j]] <- a$vectors %*% diag(a$values^(1/2)) %*% t(a$vectors)
    }
  }
 if(scenario==1) rnum <- function() rmunif(n)
 if(scenario==2) rnum <- function() rtwoelon(n)
 if(scenario==3) rnum <- function() rtwoclose_elon(n)
 if(scenario==4) rnum <- function() rmmnd(n, mu, sigma, p, d)
 if(scenario==5) rnum <- function() rmmnd(n, mu, sigma, p, d)
```

```
if(scenario==6) rnum <- function() rmmnd(n, mu, sigma, p, d)
if(scenario==7) rnum <- function() rmmnd(n, mu, sigma, p, d)
if(scenario==8) rnum <- function() rmmnd(n, mu, sigma, p, d)

#definisco le matrici che conterranno i valori degli indici
av.sil <- matrix(NA, B, n.group-1)
v.sil <- matrix(NA, B, n.group-1)
ray <- matrix(NA, B, n.group-1)
har <- matrix(NA, B, n.group-1)
jump.mat <- matrix(NA, B, n.group)
gap.unif <- matrix(NA, B, n.group)
gap.pc <- matrix(NA, B, n.group)
predi.stren <- NULL #vettore che conterra' gia il numero
#ottimo di gruppi secondo tale criterio
for(i in 1:B){
 #generazione del campione secondo lo scenario predefinito
 x <- rnum()
 di <- dist(x)
 km<- kmeans(x, 1)
 for(j in 2:n.group){
 kmm <-list()
 within.dev <- NULL
 c.prec <- km$cluster
 for(pr in 1:15){
 #itero per 15 volte il k_means in quanto e' un algoritmo che puo'
 #convergere a ottimi locali e scelgo la partizione che minimizza
 #la devianza within
 kmm[[pr]] <- kmeans(x,j)
 within.dev[pr] <- kmm[[pr]]$tot.withinss
 }
 pr <- which.min(within.dev)
 km <- kmm[[pr]]
 sil.km <- silhouette(km$cluster, di)
 av.sil[i,j-1] <- mean(sil.km[,3])  #slhouette media
 v.sil[i,j-1] <- var(sil.km[,3])
 #indice di Ray Turi
 ray[i,j-1] <- as.numeric(intCriteria(as.matrix(x), km$cluster,
                                              crit="Ray_Turi"))
 #indice di Calinski e Harabasz
 har[i,j-1] <- as.numeric(intCriteria(as.matrix(x),km$cluster,
                                            crit="Calinski_Harabasz"))
 gap.unif[i, j-1] <- index.Gap(x, cbind(c.prec, km$cluster), B=100,
                                            method="k-means")$diffu
 #statistica Gap
 gap.pc[i, j-1] <- index.Gap(x, cbind(c.prec, km$cluster),
 B=100, method="k-means", reference.distribution =  "pc")$diffu
 if(j==n.group) c.prec <- km$cluster
 }
kmm <-list()
```

```
 within.dev <- NULL
 for(pr in 1:15){
  kmm[[pr]] <- kmeans(x,n.group+1)
  within.dev[pr] <- kmm[[pr]]$tot.withinss
  }
 pr <- which.min(within.dev)
 km <- kmm[[pr]]
 gap.unif[i, n.group] <- index.Gap(x, cbind(c.prec, km$cluster), B=100,
                                method="k-means")$diffu #statistica Gap
 gap.pc[i, n.group] <- index.Gap(x, cbind(c.prec, km$cluster),
          B=100, method="k-means", reference.distribution = "pc")$diffu
 jump.mat[i, ] <- cDistance(x, kseq = 2:n.group)$Jump
 #metodo Jump
 predi.stren[i] <- prediction.strength(x, Gmin=2, Gmax=n.group)$optimalk
 #prediction strength che fornisce gia' il k ottimo per ogni campione
 }
 criteria <- list(av.sil, v.sil, ray, har, jump.mat,
                             gap.unif, gap.pc, predi.stren)
 names(criteria)= c("silhouette.media", "var.silhouette", "Ray_Tury",
    "Calinski_Harabasz", "jump", "Gap.unif","Gap.pc", "Prediction_strength")
 return(criteria)
}
#la funzione che segue genera valori dai diversi scenari previsti
#e restituisce i risultati delle stime di k dei metodi inferenziali
simfuncl <- function(n=NULL, B=100, mu=NULL, sigma=NULL,
                         p=NULL, d=NULL, n.group=8, scenario=3){
 if(scenario!= 1 & scenario!= 2 & scenario!= 3) {
  for(j in 1:length(p)){
   a <- eigen(sigma[[j]])
   sigma[[j]] <- a$vectors %*% diag(a$values^(1/2)) %*%
t(a$vectors)
    }
  }
  parfun <- function(p, k){
    (k-1) + k*p + k + k*p*(p-1)/2 + p-1
  }
  if(scenario==1) rnum <- function() rmunif(n)
  if(scenario==2) rnum <- function() rtwoelon(n)
  if(scenario==3) rnum <- function() rtwoclose_elon(n)
  if(scenario==4) rnum <- function() rmmnd(n, mu, sigma, p, d)
  if(scenario==5) rnum <- function() rmmnd(n, mu, sigma, p, d)
  if(scenario==6) rnum <- function() rmmnd(n, mu, sigma, p, d)
  if(scenario==7) rnum <- function() rmmnd(n, mu, sigma, p, d)
  if(scenario==8) rnum <- function() rmmnd(n, mu, sigma, p, d)

  #inizializzo i vettori che conterranno il numero k ottimale
  #secondo il relativo criterio
  Bic.crit <- NULL
  lrt.crit <- NULL
```

```
  bootLRT.crit <- NULL
  for(i in 1:B){
    x <- rnum()
    pp <- ncol(x)
    bootLRT.crit[i] <- max(mclustBootstrapLRT(x, modelName = "VEV")$G)
    Bic.k <- NULL
    p.val <- NULL
    n <- nrow(x)
    cl1 <- Mclust(x, 1, modelName = "VEV")
    for(j in 1:n.group){
      cl2 <- Mclust(x, j+1, modelName = "VEV")
      p.val[j] <- calc_lrt(n, cl1$loglik, parfun(pp, j), j,
                           cl2$loglik, parfun(pp, j+1), j+1)[4]
      Bic.k[j]<-cl1$bic
      cl1 <- cl2
    }
    Bic.crit[i] <- which.max(Bic.k)
    lrt.crit[i] <-  min(which(p.val>0.05))
    print(i)
  }
  crit <- list(bootLRT.crit, Bic.crit, lrt.crit)
  names(crit) <- c("BootstrapLRT", "BIC_CRIT",
 "LRT_LOMENDELLRUBIN")
  return(crit)
}
```

Bibliografia

C. Fraley and E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

C. C. Aggarwal. *Data Mining: The Textbook*, volume 1. Springer, 2015.

G. Wilkin and Huang X. K-means clustering algorithms: implementation and comparison. In *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*. IEEE, 2007.

I. Kosmidis and D. Karlis. Model-based clustering using copulas with applications. *Statistics and computing*, 26(5):1079–1099, 2016.

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics: Theory and Methods*, 3(1):1–27, 1974.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

S. Ray and R. H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, volume 137, page 143. Citeseer, 1999.

C. A. Sugar and G. M. James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.

R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

J. H Wolfe. A monte carlo study of the sampling distribution of the likelihood ratio for mixtures of multinormal distributions. Technical report, Naval personnel and training research lab San Diego CA, 1971.

Y. Lo, N. R. Mendell, and D. B. Rubin. Testing the number of components in a normal mixture. *Biometrika*, 88(3):767–778, 2001.

G. J. McLachlan. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 36(3):318–324, 1987.

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.

C. Fraley, A. E Raftery, L. Scrucca, T. B. Murphy, Fop M., and M. L. Scrucca. Package 'mclust', 2012.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

L. Litti, A. Colusso, M. Pinto, E. Ruli, A. Scarsi, L. Ventura, G. Toffoli, M. Colombatti, G. Fracasso, and M. Meneghetti. Serrs multiplexing with multivalent nanostructures for the identification and enumeration of epithelial and mesenchymal cells. *Scientific Reports*, 10(1):1–10, 2020.