Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

TRATTAMENTO DEL CONCETTO DI CLUSTERING A PARTIRE DALLA DEFINIZIONE DI MISURE DI SIMILARITA' E CRITERI VALUTATIVI

Relatore:
Prof. Gastone Castellani

Presentata da:
Niccolò Barbieri

Anno Accademico 2020/2021

ABSTRACT

Con la presente tesi ci si poneva l'obbiettivo di studiare lo stato dell'arte riguardo al tema degli algoritmi di clustering; principalmente su dati non categorizzati.
A tal fine sono stati analizzate prima le tematiche riguardanti misure di similarità e criteri valutativi, ponendo esempi sia per dati di tipo numerico che per dati categorizzati; questi risultano essere temi paralleli al più centrale tema proposto, ma restano argomenti fondamentali al fine della comprensione del funzionamento degli algoritmi di clustering.
Successivamente è stata portata avanti una disamina sulla tematica principale, ovvero sul tema degli algoritmi veri e propri, trattando in primo luogo le famiglie principali, ossia il clustering gerarchico e la controparte partizionale, per queste distinzioni è stato sfruttato l'articolo[1]; per poi elencare anche metodi alternativi; spesso più specifici ma non meno importanti; infine è stata inserita una breve rappresentazione di alcuni potenziali utilizzi di questi algoritmi, sia in ambito di ricerca che in ambito aziendale.

Sommario

ABSTRACT 5
1. INTRODUZIONE 11
2. MISURE DI SIMILARITÀ 12
2.1. Misure su dati numerici: 12
2.2. Esempi di misure su dati numerici: 13
2.2.1. Misura di Minkowski 13
2.2.2. Misura di Pearson 13
2.2.3. Misura di Coseno 13
2.2.4. Misura di Chebyshev 13
2.2.5. Misura di Jaccard 14
2.2.6. Misura di Manhattan 14
2.2.7. Coefficiente di Dice 14
2.3. Conclusione Sulle Misure Su Dati Numerici: 15
2.4. Misure su dati Categorizzati: 15
2.5. Esempi di Misure su Dati Categorizzati 16
2.5.1. Misura di Gower 16
2.5.2. Misura di Goodall 17
2.5.3. Misura della Frequenza di Occorrenza Inversa 17
2.5.4. Misura di Eskin 18
2.5.5. Misura di Smirnov 18
2.5.6. Misura di Frequenza di Occorrenza 18
2.5.7. Misura di Anderberg 19
2.5.8. Misura di Lin 19
2.5.9. Misura di Gambaryan 20
2.6. Conclusioni Sulle Misure Su Dati Categorizzati 20
3. CRITERI VALUTATIVI 21
3.1. Criteri Valutativi Interni e Criteri Valutativi Esterni 21
3.1.1. NCC un criterio misto 22
3.2. Esempi di Criteri Valutativi Interni 23
3.2.1. Entropia 23
3.2.2. Silhouette 23
3.2.3. Criterio di dispersione 25
3.2.4. Misura dell'Utilità Delle Categorie 25
3.2.5. Somma degli errori quadratici 25
3.2.6. Compattezza 26

3.2.7. Criterio di Condorcet 26
3.3. Esempi di Criteri Valutativi Esterni 27
3.3.1. Misura Basata sulla Mutua Informazione 27
3.3.2. Indice di Rand 27
3.3.3. Indice di Jaccard 28
3.3.4. Indice di Fowlkes-Mallows 28
3.3.5. Misura F 29
3.4. Conclusioni Sui Criteri Valutativi 29
4. ALGORITMI DI CLUSTERING 30
4.1. Clustering Gerarchico 31
4.1.1. Single Linkage 32
4.1.2. Average Linkage 32
4.1.3. Complete Linkage 32
4.1.4. Clustering Gerarchico Migliorato 33
4.2. Clustering Partizionale 34
4.2.1. K-Means 35
4.2.2. Fuzzy C-Means 36
4.3. Ulteriori Metodi di Clustering 37
4.3.1. Clustering di Grafi 37
4.3.2. Clustering tramite Griglia 38
4.3.3. Spectral Clustering (normalizzato e non) 38
4.3.4. Clustering Basato su Modelli 39
4.3.5. Clustering Multi-Obbiettivo 40
4.3.6. Clustering Basato su Distribuzioni 41
4.3.7. Clustering Basato su Metodi Evolutivi 41
4.3.8. Clustering Basato su Ricerca 42
4.3.9. Clustering di Sovrapposizione 42
4.3.10. Clustering Basato su Collaborazione Fuzzy 43
4.4. APPLICAZIONI 44
4.4.1. Applicazioni nella Ricerca: Omics 44
4.4.2. Applicazioni nell'Analisi Immagini 45
4.4.3. Applicazioni Esterne alla Ricerca 46
4.5. Conclusioni 46
5. BIBLIOGRAFIA 47

1. INTRODUZIONE

Nelle scienze odierne il problema del raggruppamento e della categorizzazione di oggetti in classi di similarità è un tema che sempre più trova spazio in svariati ambiti tematici, che spaziano dall'ingegneria fino alle scienze umane, passando anche per problemi del quotidiano come possono essere le analisi dei dati prodotti dai social media o dei dati riguardanti la semeiotica e l'anamnesi delle malattie. Per problemi simili prendono sempre più piede soluzioni basate su determinati algoritmi detti di clustering che hanno come scopo, effettivamente, il creare raggruppamenti di oggetti in base a determinate proprietà; questi cluster solitamente nelle scienze odierne prendono nomi differenti e specifici ma in ogni ambito la necessità di classificare i dati è una costante.

Questo tipo di algoritmi, finalizzati al raggruppamento e la classificazione di dati, vengono chiamati anche classificatori e possono essere divisi in due gruppi in base a come sono costruiti i dati da classificare: nel caso si operi su dati etichettati, cioè dati di cui già si conosce la categoria; si parla di algoritmi supervisionati, mentre nel caso si operi su dati non etichettati si parla di algoritmi non supervisionati. I primi, che sono stati ampiamente analizzati nella letteratura, trovano principale esempio nelle reti neurali che solitamente si basano sull'analisi di dati di questo tipo; i secondi invece, più complessi da trattare in questo caso necessitano di algoritmi ad hoc che sfruttino determinate proprietà per accorpare i dati tra di loro e formare così gli agglomerati (cluster) da poter sfruttare per i motivi sopra elencati; questi algoritmi non supervisionati sono i così detti algoritmi di clustering di cui tratteremo, questi risultano più complessi da trattare della controparte supervisionata, in quanto non solo sono svantaggiati nel calcolo a causa dell'assenza di etichette, ma emergono problemi inattesi, ad esempio aumentando le dimensioni del set di dati, che causano comportamenti che vanno oltre al semplice aumento di complessità computazionale.

Potremmo dare una buona definizione di clustering partendo da come Rokach e Maimon[2] descrivono l'operato di questi classificatori: cioè che il clustering consiste nel dividere i pattern di dati in sottogruppi in modo che gli elementi più simili si trovino nello stesso sottogruppo. Solitamente gli algoritmi sono costruiti in modo che questi sottogruppi siano disgiunti, anche se alcune eccezioni esistono (come i classificatori fuzzy) ma disgiunti o meno tutti i classificatori formano un ricoprimento del set di dati iniziali o in sostanza: $\bigcup_i S_i = D$ dove gli S_i sono i sottogruppi formati e D è il set di dati iniziale. Le divisioni sono basate su svariati concetti di similarità, che dipendono dalla scelta dei dati e dal problema che ci si pone, il metodo più comune è valutare la distanza degli oggetti da un determinato punto che può essere il centro del cluster oppure anche la distanza tra tutti gli elementi. Esistono poi svariati metodi di valutazione dell'operato di un determinato algoritmo in base all'aderenza del risultato con quanto atteso in principio.

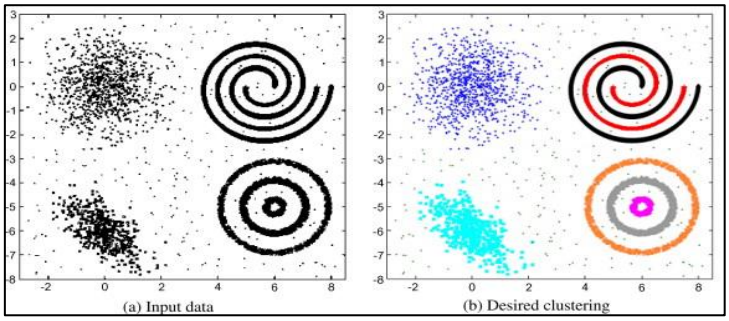


Figura 1: In figura notiamo il dataset di partenza (a sinistra); e il risultato del processo diviso in 7 cluster colorati diversamente[3]

2. MISURE DI SIMILARITÀ

Il concetto di similarità è fondamentale nella trattazione che stiamo portando avanti, infatti è proprio attraverso questo strumento, cioè la similarità tra due membri, che si vanno poi a definire i cluster, più due elementi sono "simili" più sarà probabile che si trovino nello stesso agglomerato. Un'ottima definizione per la similarità può essere data sostenendo che la similarità è la forza della relazione tra due data items, rappresenta quindi quanto sono simili i dati[4]; detto ciò, è facile notare quanto sia fondamentale la scelta di un'appropriata misura della similarità tra due elementi, in modo che all'interno dei cluster che ci aspettiamo questa misura sia massima, molto spesso le prestazioni dell'algoritmo dipendono fortemente dalla scelta di questa funzione in modo che sia ottimizzata per il problema.

Per dati numerici è più semplice la trattazione in quanto solitamente viene definita una nozione di similarità a partire dal concetto di distanza che si definisce nello spazio dei dati, questa definizione di similarità va a restringere i tipi di funzioni, ovviamente, in quanto la definizione di distanza implica condizioni più stringenti come la necessità che la forma sia simmetrica e che sia minima nel caso si consideri la distanza tra un punto e se stesso; se si richiede che sia effettivamente nulla se e solo se applicata a due elementi uguali, e si aggiunge che rispetti la disuguaglianza triangolare allora si viene a definire una distanza metrica.

Nel caso di dati non numerici, in particolare trattando dati categorici, l'analisi è più complicata in quanto questo tipo di dati non è intrinsecamente dotato di un concetto di ordine; quindi, non si può utilizzare il concetto di distanza sopra espresso, solitamente vengono utilizzate delle strutture note come matrici di similarità che permettono di esprimere la similarità tra due elementi.

2.1. Misure su dati numerici:

Come abbiamo detto in precedenza, per valutare la similarità di due elementi nell'ambito di dataset composti da dati di tipo numerico si utilizzano solitamente distanze come strumento di valutazione. Perché una funzione si possa definire distanza si richiede che sia simmetrica rispetto alle due entrate: $d(x_i, x_j) = d(x_j, x_i)$ e che sia minima nel caso le due entrate siano uguali, se poi si richiede in aggiunta che la distanza sia positiva e verifichi la disuguaglianza triangolare avremo una metrica, che poi viene applicata allo spazio formato dai vettori costruiti usando come coordinate le variabili numeriche che caratterizzano i nostri elementi del dataset, rendendolo uno spazio metrico con le sue proprietà.

Ci sono svariati esempi di misure utilizzate in vari ambiti, se ne presentano alcune, le più diffuse; a partire dalla misura euclidea, inserita in una raccolta di misure più ampia i cui membri vengono chiamati misure di similarità di Minkowski, considerando poi altri esempi come il coseno di similarità, le misure di Pearson, di Jaccard, di Chebyshev, di Manhattan, e infine il coefficiente di Dice.

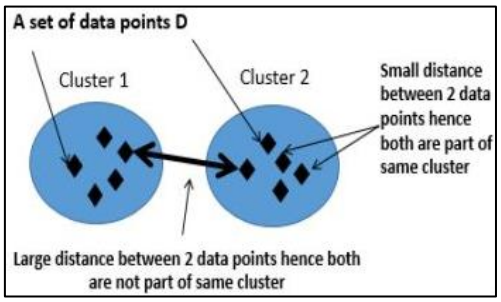


Figura 2: In figura un esempio generale del concetto di similarità[5]

2.2. Esempi di misure su dati numerici:

2.2.1.Misura di Minkowski

Le misure di Minkowski sono una famiglia di misure che associano un valore di similarità a due dati a partire dalla relazione di distanza:

$$d_r(i,j) = \|ij\|_r = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^r\right)^{\frac{1}{r}}$$

Tra queste misure distinguiamo la ben nota misura euclidea definita per r = 2:

$$d_2(i,j) = \sqrt{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^2}$$

In queste relazioni con x_i indichiamo

2.2.2.Misura di Pearson

Questo coefficiente di correlazione è stato definito per la prima volta da Bravais[6] e poi mostrato da Pearson[7] tramite la relazione:

$$d(x_i, x_j) = \frac{(x_i - \overline{x_i})^T}{\|x_i\|} * \frac{(x_j - \overline{x_j})}{\|x_j\|}$$

Dove nella relazione indichiamo con $\overline{x_i}$ e con $\overline{x_j}$ i valori medi dei termini x_i e x_j stimati su tutte le dimensioni; questa risulta quindi una misura normalizzata che varia tra -1 e +1

Questo indice statistico viene solitamente utilizzato per stimare la correlazione lineare tra i dati per questo è anche chiamato indice di correlazione lineare.

2.2.3.Misura di Coseno

La misura di coseno o coseno di similarità è una misura utilizzata solitamente in ambito di estrazione di dati e nella raccolta di informazioni, in particolare nell'analisi testuale.

La misura della similarità viene calcolata come il coseno dell'angolo tra i vettori rappresentati dalle caratteristiche dei dati, in particolare considerando x_i ed x_j due elementi del dataset avremo:

$$d(x_i, x_j) = \cos(\theta) = \frac{x_i^T * x_j}{\|x_i\|\|x_j\|}$$

2.2.4.Misura di Chebyshev

Questa particolare misura viene definita a partire da una distanza; questa particolare distanza, detta anche distanza del massimo o distanza della scacchiera, associa presi due membri del set, il massimo valore di separazione tra le coordinate.

$$d(x_i, x_j) = max_k |x_{k,i} - x_{k,j}|$$

2.2.5.Misura di Jaccard

Questo indice statistico viene solitamente usato per esprimere la differenza tra diversi set di campionamento, in particolare valuta la similarità come il rapporto tra l'unione e l'intersezione dei set:

$$d(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Questa particolare misura ammette un'altra espressione nota anche come coefficiente di Tanimoto, nel caso di set binari, infatti, si può estendere il coseno di similitudine per ottenere questa espressione definita da Strehl et al.[8] come coefficiente esteso di Jaccard:

$$d(x_i, x_j) = \frac{x_i^T * x_j}{\|x_i\|^2 + \|x_j\|^2 - x_i^T * x_j}$$

2.2.6.Misura di Manhattan

La distanza di Manhattan o geometria del Taxi è una particolare forma metrica; questa definizione parte dalla possibilità di trattare i membri del dataset come vettori aventi per coordinate le variabili proprie del set.

Questa funzione stima quindi la distanza tra i vettori come somma delle differenze in modulo fatte coordinata per coordinata:

$$d(x_i, x_j) = \sum_{k=1}^{n} |x_{i,k} - x_{j,k}|$$

Tra i vantaggi di questa misura c'è la semplice estendibilità ad ordini di grandezza superiori, proprietà da tenere conto nella scelta della misura.

Il nome deriva dalla struttura stradale di numerose città, come ad esempio New York, in cui le strade sono strutturate in modo ortogonale; quindi, un mezzo in moto in una città del genere che segua le strade calcola il suo percorso proprio con questa metrica.

2.2.7.Coefficiente di Dice

Questo indice sviluppato da Sørensen[9] e Dice[10] indipendentemente, parte da una definizione simile a quella del coefficiente di Jaccard:

$$d(x_i, x_j) = 2\frac{x_i^T * x_j}{\|x_i\|^2 + \|x_j\|^2}$$

Oppure in notazione insiemistica:

$$d(x_i, x_j) = 2\frac{|A \cap B|}{|A| + |B|}$$

La principale differenza tra questo indice e quello di Jaccard è che questo risulta una semimetrica perché non soddisfa la disuguaglianza triangolare, mentre la misura di Jaccard risulta effettivamente una metrica; queste due misure sono strettamente legate però, è possibile infatti nota una qualsiasi delle due ricavare l'altra, posti D e J rispettivamente l'indice di Dice e di Jaccard avremo:

$$J = \frac{S}{2 - S} \quad e \quad S = \frac{2J}{(J + 1)}$$

2.3. Conclusione Sulle Misure Su Dati Numerici:

Per questa prima disamina abbiamo considerato, come detto in precedenza, solo misure che sono effettivamente delle distanze, questa scelta porta numerosi benefici che provengono dalle proprietà della distanza, ma comporta anche qualche limite, oltre alla complessità nell'applicare queste misure in dataset non numerici abbiamo altri potenziali problemi che originano dal fatto che le distanze si limitano a calcolare la distanza spaziale tra due punti, questo può divenire limitante nel caso si considerino serie di dati per le quali si considera più opportuno definire il concetto di vicinanza e quindi di similarità tramite associazione e separazione dei pattern dei punti.

2.4. Misure su dati Categorizzati:

Come abbiamo detto in precedenza l'operazione di misura della similarità, se fatta su dati non prettamente numerici, risulta decisamente più complessa, per visualizzare ciò possiamo pensare ad un esempio, supponiamo di voler raggruppare le schede personali di Facebook o di qualsiasi social network in cluster di elementi simili, per fare ciò bisognerà valutare quanto sono simili e quanto sono diversi i vari profili, se prendiamo ad esempio due individui che hanno caratteristiche simili, quanto saranno diversi se uno dei due risulta sposato mentre l'altro no?

Questa domanda ovviamente è un esempio semplicistico dei problemi che si trovano nell'analisi di dati di tipo non prettamente numerico, per ovviare a questi problemi nel caso di dati di questo tipo non si sfrutteranno più le distanze di cui abbiamo parlato in precedenza ma bensì alcune funzioni di somiglianza o diversità basate su matrici dette di similarità; queste matrici servono a determinare se i dati accordano tra loro o meno e quindi affibbiare un valore di similarità ai due dati.

Come detto in precedenza le misure in questo tipo di dati sono di due tipi, le misure di similarità il cui valore alto si associa a dati simili o uguali, quindi vicini dal punto di vista di una potenziale metrica, e misure di diversità, più affini a quelle viste fino ad ora, che ad un valore alto associano dati molto diversi e quindi distanti dal punto visto di una potenziale metrica.

Nel caso di dati misti sarà poi necessario un passaggio preventivo per operare e cioè sarà necessario discretizzare quelle variabili dei dati che non sono categorizzate per poter operare in modo unico su tutto il set.

Costruiamo ora una definizione generale di una potenziale funzione di similarità su un set di dati categorizzati.

Partiamo definendo un set di oggetti Q composto da m elementi X_i caratterizzati a loro volta da n variabili $X_i=(X_{i,1},\dots,X_{i,n})$ prendiamo ora gli elementi X_1 e X_2 possiamo definire la misura della similarità tra questi due elementi come $S(X_1,X_2)=\sum_{k=1}^{n}\omega_k S_k(X_{1,k},X_{2,k})$ dove definiamo $S_k(X_{1,k},X_{2,k})$ come la coordinata k di una determinata funzione prescelta; e definiamo ω come il peso di ogni coordinata, che per semplicità possiamo porre ad 1 per ogni valore di k in quanto il nostro discorso è in generale e non si propone di fare studi sulla struttura del database per determinare quali variabili sono più importanti nel nostro problema.

Una volta calcolato il termine $S(X_1,X_2)$ possiamo definire la matrice S_G a partire dai $S(X_1,X_2)$ cioè $S_G=\begin{bmatrix} S(X_1,X_1) & \cdots & S(X_1,X_m) \\ \vdots & \ddots & \vdots \\ S(X_m,X_1) & \cdots & S(X_m,X_m) \end{bmatrix}$ dove la diagonale è interamente uguale a 1.

Ora consideriamo invece *[X_k]* cioè la raccolta dei possibili valori che la kappesima variabile può assumere nel set; definiamo anche $||[X_k]||$ il numero di possibili valori che la variabile può assumere, e quindi la cardinalità di *[X_k]*, che chiaramente può essere al massimo *m*.
Ora possiamo definire la frequenza *$f_k(w)$* cioè la frequenza con cui la variabile k assume il valore *w* e la probabilità di ottenere il valore *w*: $p_k(w) = \frac{f_k(w)}{m}$ mentre la probabilità stimata risulta: ${p_k}^2(w) = \frac{f_k(w)*(f_k(w)-1)}{m(m-1)}$.
Infine, consideriamo le possibili definizioni di $S_k(X_{1,k}, X_{2,k})$ abbiamo 3 tipi principali di S_k che possiamo considerare:

1) Misure che assegnino nel caso ci sia match un valore diverso da 0 mentre assegnano 0 se il match non avviene:

$$S_k(X_i, X_j) = \begin{cases} y\ se\ i = j\ (con\ y \in [0,1]) \\ 0\ se\ i \neq j \end{cases}$$

2) Misure che assegnino un valore 1 nel caso il match avvenga e un valore e un valore diverso da 1 nel caso i dati risultino incompatibili:

$$S_k(X_i, X_j) = \begin{cases} 1\ se\ i = j \\ y\ se\ i \neq j\ (con\ y \in [0,1]) \end{cases}$$

3) Misure che assegnino valori diversi e non preventivamente fissati sia che i dati risultino compatibili sia che non lo risultino:

$$S_k(X_i, X_j) = \begin{cases} y\ se\ i = j\ (con\ y \in [0,1]) \\ z\ se\ i \neq j\ (con\ z \in [0,1]) \end{cases}$$

2.5. Esempi di Misure su Dati Categorizzati

2.5.1.Misura di Gower

La misura di Gower (GOW) è una misura di similarità definita da Gower[11] molto utilizzata in quanto risulta semplice e flessibile; infatti, questa funzione può essere usata sia su dati di tipo numerico che su dati di tipo categorizzato; anche se ora noi consideriamo solo un'applicazione su dati del secondo tipo.
Risulta una misura di tipo 1 in quanto nel caso di dati incompatibili assegna un valore 0 se si ha accordo tra i dati avremo il valore 1; in particolare risulta che:

$$S_k(X_i, X_j) = \begin{cases} 1\ se\ X_{i,k} = X_{j,k} \\ 0\ se\ \ X_{i,k} \neq X_{j,k} \end{cases}$$

Avremo poi definito l'elemento di matrice $S(X_1, X_2)$ come valor medio dei coefficienti ottenuti confrontando le varie variabili del set:

$$S(X_1, X_2) = \sum_{k=1}^{n} \frac{\omega_k S_k(X_{1,k}, X_{2,k})}{n}$$

2.5.2.Misura di Goodall

Questa particolare misura risulta basata sulla probabilità; in particolare è una misura di tipo 1 che utilizza un insieme di valori detto *MSFVS(w) (Most Similar Attributes Value Set)* questo viene definito come l'insieme dei valori u associati alla variabile kappesima tali per cui la probabilità associata è minore o uguale a quella di *w*.
Consideriamo ora X_i, X_j, due membri del set, di cui consideriamo il kappesimo attributo, avremo che la probabilità che *u* sia incluso nell' $MSFVS(X_{i,k})$ è definita da:

$$p_k{}^2(w) = \frac{f_k(w) * (f_k(w) - 1)}{m(m-1)}$$

Andando poi a sommare le varie probabilità otteniamo la diversità:

$$D(X_{i,k}) = \sum_{w \in MSFVS(X_{i,k})} p_k{}^2(w)$$

Da cui possiamo poi stimare la similarità prendendone il complementare in caso di match:

$$S_k(X_i, X_j) = \begin{cases} 1 - D(X_{i,k}) & se\ X_{i,k} = X_{j,k} \\ 0 & se\ X_{i,k} \neq X_{j,k} \end{cases}$$

2.5.3.Misura della Frequenza di Occorrenza Inversa

Questa misura parte da considerazioni su un coefficiente detto *IDF (Inverse Document Frequency)* utilizzato in analisi testuale per stimare l'importanza di un termine all'interno di un documento; questo particolare coefficiente è definito come $IDF = log_{10}(\frac{d_w}{D})$ dove d_w è il numero di documenti in cui compare la parola *w*, e D indica il numero di documenti totali.
Si può trovare, in realtà, una definizione alternativa per l'indice IDF che sfrutta la frequenza f_w con cui la parola *w* compare nel testo: $IDF = log_{10}(f_w)$; facendo ora un parallelo con il nostro problema avremo che f_w corrisponde a $f_k(w)$ cioè la frequenza con cui la variabile *k* assume il valore *w*.
Per come abbiamo definito il nostro coefficiente *IDF* possiamo affibbiare a *w* i valori $X_{i,k}$ e $X_{j,k}$ e avremo se i dati risultano incompatibili dovremo considerare entrambi i coefficienti *IDF* da cui:

$$missmatch = log_{10}(f_k(X_{i,k}))\ log_{10}(f_k(X_{i,k}))$$

Per mantenere poi i vincoli imposti sulla misura andiamo a definire $S_k(X_i, X_j)$ come:

$$S_k(X_i, X_j) = \begin{cases} 1 & se\ X_{i,k} = X_{j,k} \\ \dfrac{1}{missmatch + 1} & se\ X_{i,k} \neq X_{j,k} \end{cases}$$

In questo modo $S_k(X_i, X_j)$ risulta compreso nell'intervallo [0,1] cosa non garantita dalla semplice definizione di $missmatch$.

2.5.4.Misura di Eskin

Questa misura risulta di tipo 2 e quindi enfatizza il peso delle variabili che possono assumere più valori, al contrario della precedente misura, la funzione di Eskin risulta compresa nell'intervallo $\left[\frac{2}{3}, \frac{m^2}{m^2+2}\right]$ dove m indica il numero di possibili valori che la variabile considerata può assumere, nel caso si abbiano solo due possibili valori la misura raggiunge il valore minimo, mentre invece il massimo valore viene raggiunto se la variabile assume nel dataset *m* valori distinti.
In particolare, avremo che la nostra misura è espressa dalla relazione:

$$S_k(X_i, X_j) = \begin{cases} 1 & se\ X_{i,k} = X_{j,k} \\ \dfrac{m^2}{m^2+2} & se\ \ X_{i,k} \neq X_{j,k} \end{cases}$$

2.5.5.Misura di Smirnov

La misura di Smirnov risulta una misura del tipo 3 che risulta particolare in quanto non solo considera la frequenza del valore in esame dell'attributo considerato, ma considera anche la distribuzione degli altri valori che quell'attributo può assumere. Questa misura basa quindi l'assegnazione di valori sulla teoria della probabilità sia che i dati siano compatibili sia che non lo siano assegnando valori alti quando la frequenza con cui si ottengono dati affini per il valore in esame è bassa. La funzione nel caso di matching è limitata all'intervallo [2,2m] con il valore minimo raggiunto quando il valore in esame compare nel set m volte, mentre il massimo viene raggiunto quando ci sono due valori per l'attributo, il primo che compare 1 volta mentre il secondo m-1; nel caso in cui il matching non avviene avremo invece come intervallo limite [0,(m/2)-1] con il valore minimo raggiunto quando si hanno solo due possibili valori , mentre il massimo viene ottenuto quando la frequenza dell'attributo in esame raggiunge quasi il 100%.
Possiamo quindi definire la nostra funzione ora:

$$S_k(X_i, X_j) = \begin{cases} 2 + \dfrac{m - f_k(X_{i,k})}{f_k(X_{i,k})} + \displaystyle\sum_{s \in \{[x_k] \setminus X_{i,k}\}} \frac{f_k(s)}{m - f_k(s)} & se\ X_{i,k} = X_{j,k} \\ \displaystyle\sum_{s \in \{[x_k] \setminus \{X_{i,k} X_{j,k}\},\}} \frac{f_k(s)}{m - f_k(s)} & se\ X_{i,k} \neq X_{j,k} \end{cases}$$

2.5.6.Misura di Frequenza di Occorrenza

Questa funzione di misura risulta di tipo 2 come la precedente, il concetto alla base però, come si può intendere dal nome, è esattamente opposto a *IOF (Inverse Occurrence Frequency)*:

$$S_k(X_i, X_j) = \begin{cases} 1 & se\ X_{i,k} = X_{j,k} \\ \dfrac{1}{missmatch + 1} & se\ \ X_{i,k} \neq X_{j,k} \end{cases}$$

Ma in questo caso avremo $missmatch = log_{10}(\frac{m_k}{f_k(X_{i,k})})\ log_{10}(\frac{m_k}{f_k(X_{j,k})})$

2.5.7.Misura di Anderberg

Nella definizione di questa misura risulta fondamentale la relazione tra gli attributi, in quanto questa misura assegna valori alti della similarità nel caso si osservino dai confronti pochi match, e valori bassi nel caso si trovino pochi casi di incompatibilità attraverso il confronto.
Una particolarità di questa misura è che, al contrario delle altre, non ubbidisce alla relazione scritta in apertura e che quindi non si basa sui pesi, ma bensì viene a definirsi nell'intervallo [0,1] come:

$$S(X_i, X_j) = \frac{\sum_{k\in 1\le k\le n:\, X_{i,k}=X_{j,k}} \left(\frac{1}{p_k(X_{i,k})}\right)^2 \frac{2}{m_k(m_k+1)}}{\sum_{\substack{k\in 1\le k\le n: \\ X_{i,k}=X_{j,k}}} \left(\frac{1}{p_k(X_{i,k})}\right)^2 \frac{2}{m_k(m_k+1)} + \sum_{\substack{k\in 1\le k\le n: \\ X_{i,k}\neq X_{j,k}}} \frac{1}{2(p_k(X_{i,k})p_k(X_{j,k}))} \frac{2}{m_k(m_k+1)}}$$

2.5.8.Misura di Lin

La similarità di Lin è una misura di tipo 3 che si basa su alcuni concetti di teoria dell'informazione, in particolare permette di trovare un valore reale di similarità per un insieme di parole basandosi sulla probabilità di occorrenza di ciascuna parola.
Il nostro sistema può trovarsi in due particolari configurazioni, di cui verrà poi valutata la quantità di informazione contenuta, la prima delle due, la configurazione $Common(X_i, X_j)$, definisce lo stato in cui i valori associati all'attributo numero *k* per i due oggetti in esame sono uguali, quindi corrisponde allo stato di match; ne consegue che la probabilità $P(Common(X_i, X_j))$ comprenda già il confronto tra i due oggetti dato che l'informazione contenuta nei due attributi $X_{i,k}$ e $X_{j,k}$ è la stessa: $I(Common(X_i, X_j)) = -log\ (P\left(Common(X_i, X_j)\right) = 2log\ (p_k(X_{i,k}))$; mentre la seconda configurazione, $Differences(X_i, X_j)$, rappresenta lo stato in cui i valori degli attributi in esame sono differenti nei due oggetti; in questo caso la probabilità $P(Differences(X_i, X_j))$ deve includere le probabilità di ottenere i due valori nello specifico; e quindi dobbiamo trovare il valore dell'informazione nei due diversi oggetti per l'attributo k ottenendo quindi $I(Differences(X_i, X_j))$=$-log\ (P\left(Differences(X_i, X_j)\right) = 2log\ (p_k(X_{i,k}) + p_k(X_{j,k}))$.
Dopo aver descritto queste proprietà possiamo quindi definire la misura di Lin come:

$$S_k(X_i, X_j) = \begin{cases} 2log\ (p_k(X_{i,k})) & se\ X_{i,k} = X_{j,k} \\ 2log\ \left(p_k(X_{i,k}) + p_k(X_{j,k})\right) & se\ X_{i,k} \neq X_{j,k} \end{cases}$$

Possiamo dare un altro significato alla misura di Lin che abbiamo definito confrontandola con le altre misure che abbiamo visto; in particolare possiamo studiare l'andamento della nostra funzione che risulta massima nel caso in cui gli oggetti siano completamente diversi rispetto all'attributo k, mentre risulta minima se sono completamente identici rispetto a questo attributo.
Il segno meno che compare nella relazione è necessario dato che le probabilità sono valori minori di 1 che quindi danno risultati negativi se sottoposti al logaritmo, possiamo definire una misura alternativa in cui il valore più grande corrisponde allo stato di massima similarità come ci aspetteremmo da una misura del genere:

$$S_k(X_i, X_j) = \begin{cases} -2log\ (p_k(X_{i,k})) & se\ X_{i,k} = X_{j,k} \\ -2log\ \left(p_k(X_{i,k}) + p_k(X_{j,k})\right) & se\ X_{i,k} \neq X_{j,k} \end{cases}$$

2.5.9.Misura di Gambaryan

Questa misura permette di avere un approccio diverso allo studio della similarità, questa misura di tipo 1 infatti risulta massima se la frequenza del valore di cui si sta verificando il match corrisponde a $\frac{m}{2}$ mentre è minima quando la frequenza del valore considerato è m, al contrario delle misure precedentemente viste la misura di Gambaryan utilizza una probabilità singola invece che il termine quadratico; perciò, avrò $p_k(w) = \frac{f_k(w)}{m}$.

Possiamo quindi ora definire la nostra misura che risulta sempre compresa nell'intervallo [0,1]:

$$S_k(X_i, X_j) = \begin{cases} -\left[p_k(X_{i,k}) log_2\left(p_k(X_{i,k})\right) + \left(1 - p_k(X_{i,k})\right) log_2\left(1 - p_k(X_{i,k})\right)\right] & se\ X_{i,k} = X_{j,k} \\ 0 & se\ X_{i,k} \neq X_{j,k} \end{cases}$$

2.6. Conclusioni Sulle Misure Su Dati Categorizzati

Corpi di dati molto estesi, che vengono sfruttati tramite processi di analisi che richiedano l'impiego di algoritmi ad hoc per costruire cluster risultano, a causa della grande applicabilità di questi strumenti in qualsiasi ambito, per la maggior parte composti da dati misti che contengono, quindi parti categorizzate, come abbiamo detto in precedenza, in questo caso la parte più complessa risulta essere la scelta del modo adeguato per definire similarità e differenza tra i vari membri del dataset.

Abbiamo presentato 9 funzioni distinte che permettono di svolgere questo compito, queste sono state estratte da un articolo[12] nel quale poi, attraverso vari metodi valutativi, si vuole analizzare quale può essere la miglior proposta; ci si aspetta che, in accordo con la letteratura, non esista una misura di similarità ideale, in particolare perché la misura è intrinsecamente legata alle variabili del problema e alle loro caratteristiche.

Nel documento sopra citato, in controtendenza con la letteratura, si viene ad ottenere un risultato ben definito, risulta infatti che alcune misure (la prima, la quinta e la sesta), risultano decisamente migliori sotto tutti gli indicatori considerati; in particolare la misura di Gower risulta la più stabile nel complesso, essendo basata sul matching semplice.

Risulta fondamentale ricordare e notare due fatti principali, il primo che queste misure di similarità non tengono conto delle relazioni tra gli attributi del dataset, ogni attributo viene infatti trattato in modo isolato e tutti hanno lo stesso peso sul risultato, la similarità viene infatti calcolata come somma delle singole similarità stimate sui singoli attributi, pesate o meno poi nella somma; il secondo punto importante è la quantità di informazioni contenute nella singola misura di similarità, le misure che risultano migliori sono infatti tutte misure del primo tipo, che contengono il minimo numero di informazioni, infatti ad esempio la misura di Gower stima la similarità come il numero di match che avvengono, nel caso di misure del tipo 2 o 3 avremo informazioni aggiuntive (nel caso di incompatibilità o in entrambi a seconda del tipo prescelto) che quindi potrebbero richiedere maggiore cura nella creazione dei cluster; quindi risulta chiaro che nonostante abbiamo una misura che risulta migliore attraverso vari criteri valutativi, è necessario controllare e analizzare bene il metodo di clustering, il tipo di database, e la struttura delle variabili quando si va a fare la scelta della misura da utilizzare, per ottenere il massimo dall'algoritmo.

3. CRITERI VALUTATIVI

Se la scelta della misura di similarità è fondamentale per l'ottimizzazione, esiste un'ulteriore funzione la cui importanza è imprescindibile nel processo che andiamo a definire; cioè il criterio di valutazione. Queste funzioni hanno lo scopo principale di stimare la bontà del processo e quindi la funzionalità dei cluster prodotti; questo ovviamente porta con sé numerosi utilizzi alternativi, prima fra tutti la possibilità di risolvere problemi di ottimizzazione su iper-parametri del problema; come ad esempio il numero di cluster, questi parametri possono influire molto sull'efficacia del processo, ma non esiste un modo a priori di definirli; per fare ciò, solitamente, si sceglie di minimizzare una funzione che ha lo scopo di valutare i cluster prodotti; un criterio valutativo quindi.

La letteratura divide questi criteri in due grossi gruppi, i criteri valutativi interni e i criteri valutativi esterni, a cui si aggiungono due alternative cioè la valutazione manuale fatta dall'utente e la valutazione a posteriori andando ad analizzare l'utilità del risultato.

3.1. Criteri Valutativi Interni e Criteri Valutativi Esterni

Concentrandoci sui gruppi che comprendono criteri basati sulla statistica, e non considerando quindi i metodi più pratici e meno comprovati dalla teoria; andiamo a definire le differenze fondamentali tra criteri interni ed esterni.

Un criterio si dice interno se le valutazioni che porta avanti si basano sulla compattezza del cluster, e quindi l'analisi si basa sui membri stessi di un cluster, valutandone la coerenza, mentre nel confronto tra cluster diversi si va a considerare la separazione che si viene a formare tra loro; in pratica quindi, un criterio interno si definisce tale se basa l'analisi che esegue solo sui dati dei cluster stessi.

Un criterio esterno sfrutta solitamente dati non usati per il clustering che quindi funzionano da riferimento per la valutazione, questi criteri valutano quindi quanto i cluster prodotti siano prossimi a questi *benchmark* ideali.

Ora andiamo ad elencare alcuni esempi di questi criteri, prendendone sia definiti per dati di tipo categorico che per dati numerici e quindi continui, per fare ciò useremo alcune notazioni comuni in particolare avremo:

NOTAZIONE	**SIGNIFICATO**
R_i	Cluster iesimo
T	Numero totale dei cluster prodotti
m_i	Numero di membri di R_i
m	Numero totale di membri del dataset
n	Numero di attributi che descrivono un generico membro
r,s	Generici elementi del dataset
$d_{r,s}$	Distanza tra r e s

3.1.1.NCC un criterio misto

Questo particolare strumento è utilizzato per valutare il comportamento dell'algoritmo nel caso di dati categorici, in particolar questa misura valuta il confronto tra la misura intra-cluster e la distanza esterna inter-cluster.

Per prima cosa andiamo a definire la misura intra-cluster S_{intra}, chiamata anche da Rèndon et al[13] l'accordo intra-cluster; questa è definita come la differenza tra il numero di attributi che caratterizzano i nostri dati e la distanza intra-cluster tra gli elementi.

Possiamo definire la distanza $d_{r,s}$ tra gli elementi r e s uguale a 0 se gli attributi coincidono completamente, se invece si hanno delle incompatibilità tra gli attributi avremo $d_{r,s} = N * 1$ dove N è il numero di attributi non compatibili.

Avremo quindi che la nostra misura intra-cluster S_{intra} può essere definita come:

$$S_{intra}(R_t) = \sum_{r \in R_t} \sum_{\substack{s \in R_t \\ s \neq r}} (n - d_{r,s})$$

Definiamo ora la distanza inter-cluster D_{inter} che rappresenta la distanza tra due oggetti non appartenenti allo stesso cluster:

$$D_{inter}(R_t) = \sum_{r \in R_t} \sum_{s \notin R_t} d_{r,s}$$

Definite queste due grandezze possiamo esprimere il nostro indice NCC come segue:

$$NCC = \sum_{t=1}^{T} \left(S_{intra}(R_t) + D_{inter}(R_t) \right)$$

$$NCC = \sum_{t=1}^{T} \sum_{r \in R_t} \left(\sum_{\substack{s \in R_t \\ s \neq r}} (n - d_{r,s}) + \sum_{s \notin R_t} d_{r,s} \right)$$

Un'espressione alternativa di questo indice si può dare attraverso la matrice binaria *Y* avente $y_{rs}=1$ se *r* e *s* appartengono allo stesso cluster e $y_{rs}=0$ altrimenti; avremo quindi:

$$NCC = \sum_{r=1}^{m} \sum_{s \neq r} (n - 2d_{r,s}) y_{rs} + \sum_{r=1}^{m} \sum_{s \neq r} d_{r,s}$$

Osservando questa espressione possiamo notare una relazione tra le due misure; si osserva infatti che quando la misura esterna risulta piccola allora la distanza interna aumenta, questo significa che quando gli oggetti tendono ad essere molto simili all'interno del cluster questo stesso cluster tenderà a riempirsi di elementi simili; è altresì vero che quando la distanza inter-cluster aumenta i cluster stessi tendono a diventare sempre più eterogenei se comparati tra loro; quindi più distinguibili; infine si nota che se si ha un aumento della similarità e della distanza l'indice tende a crescere; altrimenti tende a decrescere.

3.2. Esempi di Criteri Valutativi Interni

3.2.1.Entropia

L'Entropia trova la sua definizione nella teoria dell'informazione come una grandezza volta a misurare il disordine presente nel dataset, trova quindi un'applicazione nel clustering per valutare l'omogeneità degli attributi all'interno dello stesso cluster.

Possiamo dare due definizioni sull'entropia in questo ambito: la prima definisce l'entropia a partire da una sequenza di eventi ognuno con probabilità p_i che permettono di definire il concetto di informazione congiunta come $I(p_1,p_2,\ldots,p_k)$ possiamo quindi definire l'entropia H come il valor medio di questa informazione congiunta $H=E[I]$.

Da questa consegue la seconda definizione: se una scelta si divide in due possibili scelte ulteriori l'entropia sarà la somma delle due entropie generate dalle due scelte.

Supponendo quindi di avere un'entropia $H[I_1,I_2,\ldots,I_T]$ generata come risultato di un processo di clustering che produce l'insieme $[R_i]$ possiamo definire la probabilità $p(u)_{l,t}$, cioè la probabilità che l'attributo l assuma il valore u all'interno del cluster t, come $p(u)_{l,t} = \frac{n_{l,t}}{m_t}$ dove $n_{l,t}$ è la frequenza con cui l'attributo l assume il valore u nel cluster t.

Una volta definita $p(u)_{l,t}$ passiamo a definire l'entropia: $H_{l,t} = -\sum_n^{V_l} p(n)_{l,t} \ln\left(p(n)_{l,t}\right)$; dove abbiamo usato V_l per indicare il numero di valori che la variabile l può assumere (nel caso di dati categorici).

Da questo consegue che quindi l'entropia del cluster t risulta $H_t = \sum_{l=1}^{n} \frac{H_{l,t}}{n}$ e quindi per l'intero risultato avremo $H = \sum_{t=1}^{T} \frac{H_t}{T}$

Chiaramente minore è l'entropia ottenuta migliore è il risultato ottenuto, perché questo indica che in media l'entropia dei singoli cluster è minore e quindi questi sono più omogenei, essendo rappresentati da un minor disordine interno.

3.2.2.Silhouette

Questo indice permette di stimare la coesione interna ad un cluster, è una funzione che associa ad un risultato alto il significato di un clustering ben eseguito; in particolare questo metodo è basato sulla definizione di una distanza ben definita come ad esempio quella euclidea o quella di Manhattan, o qualsiasi altra distanza, che indicheremo come $d_{i,j}$ dove i e j sono due elementi del dataset.

Per prima cosa definiamo un particolare cluster W che contiene l'elemento i che andiamo a considerare, avremo poi $w(i) = \frac{1}{|W|-1}\sum_{j\in W, i\neq j} d_{i,j}$ che rappresenta la distanza media tra gli elementi del cluster (mediata su $|W|-1$ elementi in quanto non consideriamo $d_{i,i}$).

Poniamo poi con $y(i,c)$ la distanza media tra i e gli elementi di un generico cluster C, diverso da W, di cui andiamo a considerare il minimo: avremo quindi $z(i) = min_{C_j\neq W} \frac{1}{|C_j|}\sum_{j\in C_j, i\neq j} d_{i,j}$ che ci permette di trovare il cosiddetto "vicino prossimo", cioè il cluster più affine all'elemento che stiamo considerando e quindi la seconda miglior scelta per contenere l'elemento i.

Posto questo possiamo definire l'indice di silhouette:

$$s(i) = \begin{cases} 1 - \dfrac{w(i)}{z(i)} & se\ \ w(i) < z(i) \\ 0 & se\ \ w(i) = z(i) \\ \dfrac{w(i)}{z(i)} - 1 & se\ \ w(i) > z(i) \end{cases}$$

Che risulta chiaramente oscillare tra -1 e 1.
Questo indice può quindi descrivere 3 situazioni distinte:

1) L'indice è prossimo a 1, si parla quindi di clustering stretto, che avviene quando gli oggetti all'interno di *W* sono più simili tra loro che agli oggetti del vicino più prossimo, non è quindi necessaria una seconda scelta in cui inserire *i*.
2) La seconda situazione avviene quando il valore dell'indice è prossimo a zero, questo significa che le due alternative sono indifferenti; quindi, l'elemento *i* potrebbe essere inserito in entrambi i cluster
3) La terza situazione avviene quando l'indice di silhouette è prossimo a -1, che indica che l'oggetto considerato è più simile ai membri del vicino più prossimo che al cluster in cui si trova, quindi andrebbe inserito in nell'altro cluster.

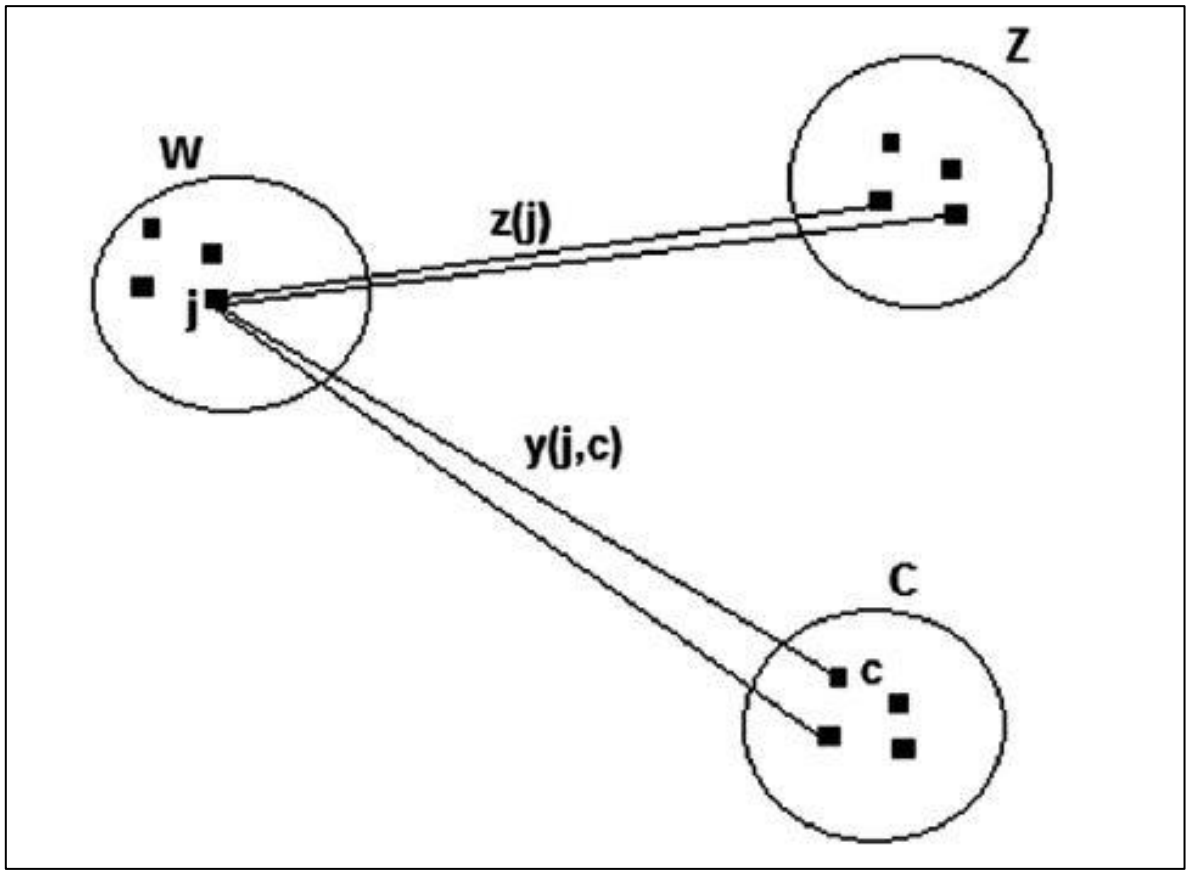


Figura 3: In figura vediamo una raffigurazione degli elementi che consideriamo nella definizione dell'indice di silhouette[12]

Ovviamente possiamo estendere la nozione di indice di silhouette anche nel caso non sia possibile definire una distanza, ad esempio nel caso di dati categorizzati, in questo caso potremmo associare a *w(i)* il valor medio della misura di similarità all'interno del cluster, sostituendo la distanza con questa misura, questo ha come principale effetto il cambio dei segni in quanto due oggetti sono tanto più simili quanto più è alto il valore della misura di similarità, mentre per la distanza era il contrario, perciò avremo:

$$s(i) = \begin{cases} 1 - \dfrac{w(i)}{z(i)} & se\ \ w(i) > z(i) \\ 0 & se\ \ w(i) = z(i) \\ \dfrac{w(i)}{z(i)} - 1 & se\ \ w(i) < z(i) \end{cases}$$

La definizione di *z(i)* risulta lievemente più complicata in quanto va fatta in astratto, definiamo quindi Y come l'insieme dei *y(i,c)* cioè le misure di similarità medie tra i e il cluster C_j; da cui dovremo estrarre il minimo, cioè il nostro *z(i).*
Possiamo poi definire la silhouette di un determinato cluster *t* come la media aritmetica dei termini di silhouette dei singoli elementi contenuti nel cluster.

3.2.3.Criterio di dispersione

Preso il kappesimo cluster avremo che la sua dispersione è definita come:

$$S_k = \sum_{x \in C_k} (x - w_k)(x - w_k)^T$$

Dove abbiamo usato w_k per indicare il vettore medio interno al cluster C_k

3.2.4.Misura dell'Utilità Delle Categorie

L'utilità delle categorie è una grandezza che misura la bontà di una categoria; posto l'insieme $F = \{f_i\} con\ i = 1, \dots, n$ in cui i vari f_i sono attributi di un generico elemento, e definiamo poi una categoria binaria $C = \{c, \bar{c}\}$ possiamo definire l'utilità della categoria come segue:

$$CU(C,F) = \left[p(c) \sum_{i=1}^{n} p(f_i|c)\, log\big(p(f_i|c)\big) + \ p(\bar{c}) \sum_{i=1}^{n} p(f_i|\bar{c})\, log\big(p(f_i|\bar{c})\big) \right] - \sum_{i=1}^{n} p(f_i) log\ (p(f_i))$$

Dove indichiamo con *p(c)* la probabilità che un elemento sia contenuto nella categoria positiva *(c)* e $p(f_i|c)$ sarà la probabilità condizionata che un elemento abbia la caratteristica f_i posto che appartiene a c mentre $p(f_i)$ è la probabilità di osservare la caratteristica f_i a priori.

3.2.5.Somma degli errori quadratici

Questo risulta uno dei criteri più usati nell'analisi numerica; è definito come la somma delle deviazioni quadratiche dalla media; posto quindi C_k un generico cluster, e w_k la media vettoriale delle istanze presenti nel cluster considerato avremo:

$$s = \sum_{k=1}^{T} \sum_{\forall x_i \in C_k} \|x_i - w_k\|^2$$

3.2.6.Compattezza

La compattezza del cluster è un metodo per valutare la bontà del processo tramite una stima metrica; questo indice, infatti, si basa sul calcolo della distanza media tra gli elementi del cluster, per questo la compattezza, nota anche come diametro del cluster, è un indice che si vuole minimizzare; supponendo che più piccolo è il cluster più simili sono gli elementi che si inseriscono.

Fissata quindi una generica distanza *d* possiamo definire la misura del diametro come:

$$dm(i) = \sum_{j=1}^{m_i} \sum_{k=j+1}^{m_i} \frac{d(j,k)^2}{m_i(m_i - 1)}$$

Da cui possiamo definire l'indice di compattezza *Cps*:

$$Cps = \sum_{i=1}^{T} dm(i)(\frac{m_i}{m})$$

Anche in questo caso è possibile dare una definizione parallela nel caso di dati categorici; partendo dal definire la distanza avremo:

$$d(X_{j,l}, X_{k,l}) = \begin{cases} 0 & se\ X_{k,l} = X_{j,l} \\ 1 & se\ X_{k,l} \neq X_{j,l} \end{cases} e\ dm(i) = \sum_{j=1}^{m_i} \sum_{k=j+1}^{m_i} \sum_{l=1}^{p} \frac{d(X_{j,l}, X_{k,l})^2}{m_i(m_i - 1)}$$

Una volta stimato il diametro il termine *Cps* risulta definito ugualmente.

3.2.7.Criterio di Condorcet

Il criterio di Condorcet si propone con un approccio alternativo al problema della valutazione, viene definito infatti come;

$$c = \sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} s(x_j, x_k) + \sum_{C_i \in C} \sum_{\substack{x_j \in C_i \\ x_k \notin C_i}} d(x_j, x_k)$$

Dove con $s(x_j, x_k)$ indichiamo la funzione di similarità mentre con $d(x_j, x_k)$ la distanza.
Possiamo estendere il criterio di Condorcet definendo il criterio–C definito da Fortier e Solomon[14]; per il quale fissata una soglia γ avremo:

$$c = \sum_{C_i \in C} \sum_{\substack{x_j, x_k \in C_i \\ x_j \neq x_k}} (s(x_j, x_k) - \gamma) + \sum_{C_i \in C} \sum_{\substack{x_j \in C_i \\ x_k \notin C_i}} (d(x_j, x_k) - \gamma)$$

3.3. Esempi di Criteri Valutativi Esterni

3.3.1.Misura Basata sulla Mutua Informazione

Con mutua informazione s'intende in teoria dell'Informazione la quantità di conoscenza che possiamo ottenere su una variabile nota un'altra-

Possiamo definire una misura esterna basata sulla mutua informazione a partire da un insieme di m istanze accorpate in g cluster; e in particolare considerando l'attributo z a cui è associato un dominio $dom(z) = \{c_1, \dots, c_k\}$ avremo:

$$C = \frac{2}{m}\sum_{l=1}^{g}\sum_{h=1}^{k} m_{l,k} log_{g,k}\left(\frac{m_{l,h}m}{m_{.,l}m_{l,.}}\right)$$

Dove abbiamo che $m_{l,h}$ indica i membri del cluster C_l che sono anche nella classe c_h mentre $m_{.,l}$ indica il numero totale di membri della classe c_l e $m_{l,.}$ indica il numero totale di elementi del cluster C_l.

3.3.2.Indice di Rand

Questo semplice criterio viene usato per confrontare i cluster con i riferimenti usati per la valutazione.

Possiamo definire quindi questo indice dando valutazioni sulla correttezza dell'esecuzione del nostro algoritmo, in particolare possiamo dividere i dati in 4 gruppi *TP*, *FP*, *TN*, *FN*; rispettivamente veri e falsi positivi, e veri e falsi negativi, a partire da questi concetti possiamo quindi definire l'indice di Rand come:

$$R = \frac{TP + TN}{TP + FP + TN + FN}$$

Dove praticamente stiamo valutando il rapporto tra gli elementi giustamente categorizzati, e il totale degli elementi che formano il dataset; ne consegue che questo indice oscilla sempre tra 0 e 1, dove più alto è il risultato migliore è il processo di clustering.

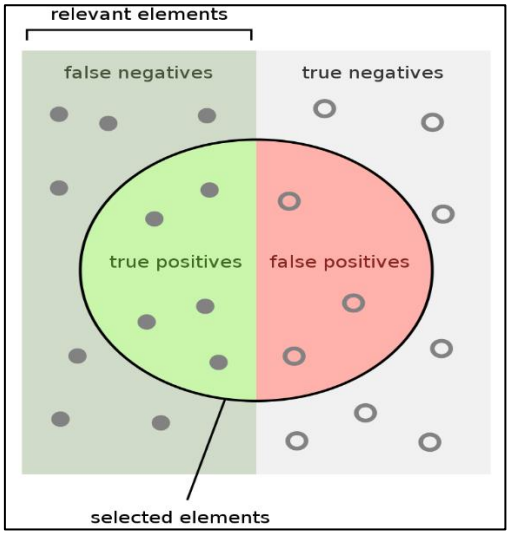


Figura 4: In figura vediamo rappresentati positivi e negativi, effettivi o meno

3.3.3.Indice di Jaccard

Questo indice viene utilizzato per considerare la somiglianza tra due dataset; avremo quindi presi i set *A* e *B*:

$$J = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

Questo risulta uguale a 1 se entrambi i dataset sono vuoti, è compreso tra 0 e 1 altrimenti. Può essere infatti visto come il numero di elementi comuni ai set diviso per il numero totale di membri dei set.

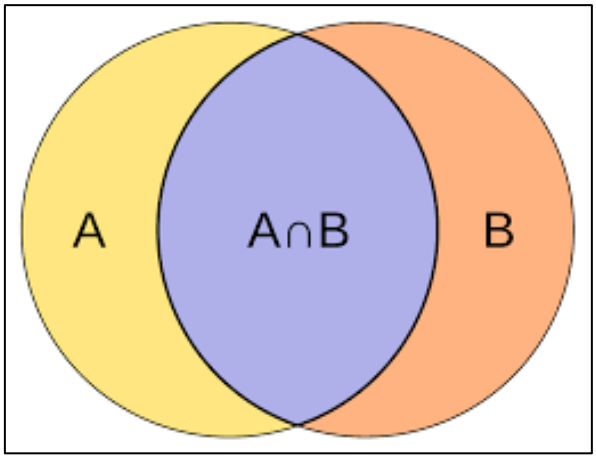


Figura 5: In figura vediamo due insiemi generici, A e B, la loro intersezione risulta evidenziata in viola

3.3.4.Indice di Fowlkes-Mallows

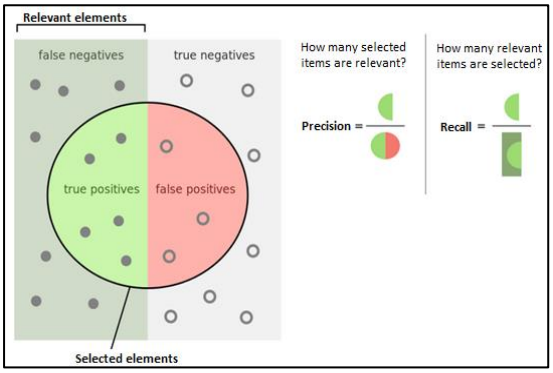
Questo indice risulta utile per verificare la somiglianza dei risultati del clustering alla fine del processo; maggiore è il valore dell'indice più i cluster sono simili.
Possiamo definire questo indice come segue:

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{FP + FN}}$$

Uno sviluppo molto utile di questo indice viene dall'analisi di dati non relazionati, risulta infatti che mentre altri indici tendono a essere imprecisi nell'analisi di questo tipo di dati; questo indice mantiene la sua coerenza approcciandosi allo zero; lo stesso si può dire per set rumorosi sui quali agisce senza subire troppo l'effetto del rumore.

3.3.5.Misura F

Questo indice statistico nasce perché a volte è necessario pesare diversamente i due possibili errori, cioè falsi postivi e falsi negativi, possiamo definire infatti due nuovi elementi che sono rispettivamente *P* cioè il rateo di precisione e *R* il rateo di richiamo definiti come segue:



$$P = \frac{TP}{TP + FP} \quad e \quad R = \frac{TP}{TP + FN}$$

Figura 6: In figura vediamo sempre una rappresentazione di positivi e negativi, mentre a destra abbiamo l'espressione di P ed R

Ora possiamo definire il parametro η detto parametro di richiamo, che risulta sempre positivo o al massimo nullo; possiamo quindi esplicitare la misura:

$$F = \frac{(\eta^2 + 1)P \cdot R}{\eta^2(P + R)}$$

In questo modo risulta che il termine R ha peso nullo se η è uguale a 0, mentre al crescere di questo parametro il peso del termine di richiamo è sempre maggiore.

3.4. Conclusioni Sui Criteri Valutativi

Entrambi i macrogruppi di criteri che abbiamo analizzato comportano dei problemi che provengono dal concetto stesso di criterio che abbiamo definito, in particolare notiamo che per definizione queste strutture sono limitate nella loro valutazione; i criteri interni, ad esempio, non sono altro che misure di similarità che non vengono utilizzate in quanto non si hanno algoritmi ottimizzati per raggruppare i dati con quelle misure; ma questo ovviamente provoca un lieve bias, ad esempio algoritmi che tendono a raggruppare sfruttando le distanze tra oggetti (come può essere K-Means) saranno sopravvalutati da questo tipo di criteri; in pratica questo genere di criteri tende a valutare quanto sono simili i due problemi di ottimizzazione in atto (similarità e criterio valutativo) invece che stimare l'utilità del cluster in se.
Nel caso di criteri esterni invece il problema si pone in quanto se fossimo dotati di riferimenti effettivamente precisi e completamente rappresentativi verrebbe meno la necessità di compiere il clustering; quindi, questi riferimenti risultano più precisi possibile ma comunque limitati ad un punto di vista ben definito, che non permette quindi completa oggettività.
Resta quindi come unica opzione per massimizzare la precisione l'utilizzo, quando possibile, del controllo umano, che risulta si soggettivo e quindi non può escludere l'uso dei metodi elencati, ma risulta importante come ultima parola sulla valutazione.

4. ALGORITMI DI CLUSTERING

Dopo aver definito gli strumenti che ci permettono di misurare le proprietà dei nostri dati, e di valutare il prodotto che viene dai nostri algoritmi, passiamo ora a definire cosa si intende effettivamente quando si parla di algoritmo di clustering.

Non esiste una definizione precisa di cosa sia un cluster, e di conseguenza anche la definizione di algoritmo di clustering è abbastanza vaga, come dimostra l'esistenza di così tanti algoritmi così tanto diversi; possiamo di fatto definire il concetto di cluster come un gruppo di dati simili sotto un certo principio, e di conseguenza un algoritmo di clustering è un processo algoritmico che permette di accorpare i dati in gruppi simili sotto un determinato principio, rappresentato da quella che definiamo misura di similarità.

Possiamo dividere questi algoritmi in base ad alcune proprietà; ad esempio, se l'algoritmo risulta esclusivo o meno; si parla di algoritmo esclusivo se il processo è pensato per inserire ogni dato in un unico cluster, al contrario se è possibile associare più di un cluster allo stesso dato parliamo di algoritmi di sovrapposizione o non esclusivi.

Molto spesso però queste divisioni sono labili, in quanto uno stesso algoritmo può essere implementato in entrambe le forme e viene usato nella forma più conveniente a seconda dei casi; una divisione invece che non permette alternative è quella sulla tecnica di base che viene sfruttata per operare il clustering in particolare le principali categorie sono il clustering gerarchico e quello partizionale; anche se esistono categorie più specifiche come ad esempio il clustering su grafi o il clustering su griglie.

Partendo ad analizzare queste due categorie principali avremo che la differenza principale sta nel fatto che mentre il clustering gerarchico tende a costruire strutture dette dendrogrammi, cioè dei grafici di relazione in cui si viene a formare una relazione gerarchica tra i cluster, nel caso del clustering partizionale la differenziazione avviene in base alla distanza da un determinato punto detto centroide del cluster.

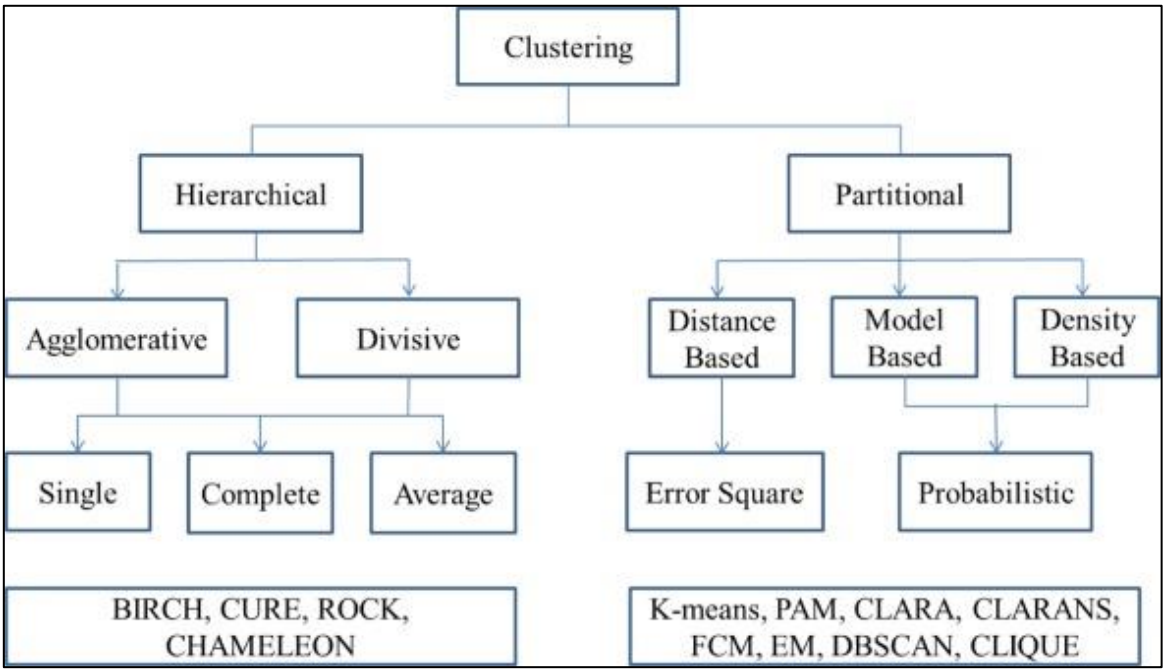


Figura 7:In questa immagine vediamo schematizzati i principali gruppi di algoritmi che tratteremo[15]

4.1. Clustering Gerarchico

Il clustering gerarchico si basa sul separare e agglomerare oggetti fino a raggiungere il risultato desiderato. Possiamo dividere quindi, questi algoritmi in base a se formano i cluster per divisione o per accorpamento di dati; esistono infatti algoritmi divisivi che partono supponendo un unico cluster contenente tutti i dati e vanno ad operare divisioni su questa struttura; in alternativa esistono algoritmi aggregativi che trattano ogni elemento come un cluster a sé e propongono l'accorpamento di cluster diversi; entrambi questi metodi tendono quindi a formare una caratteristica struttura detta dendrogramma, che rappresenta le gerarchia delle divisioni o degli accorpamenti, e quindi verrà percorsa in un verso o nell'altro a seconda del tipo di algoritmo considerato.

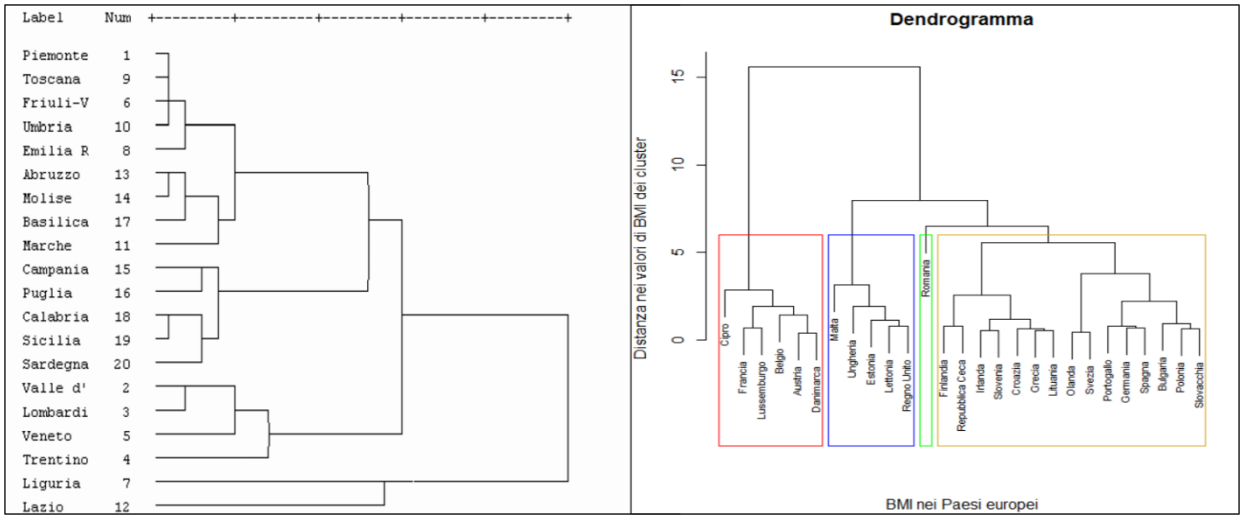


Figura 8:In figura possiamo notare due dendrogrammi; a sinistra il grafico mostra le regioni italiane accorpate in base ad indici socioeconomici[16]; mentre a destra vediamo una rappresentazione delle nazioni europee in base all'indice BMI[17].

Sia che si utilizzi quindi un meccanismo *bottom-up* o un meccanismo *top-down* è necessario poter decidere su quali cluster operare e quali lasciare invariati; in particolare solitamente viene definita una distanza tra i cluster, secondo vari metodi che vanno poi a definire che tipo di *linkage* si sta sfruttando; e si vanno a considerare ogni volta i cluster più vicini da accorpare; oppure i punti più distanti da dividere; a seconda che si stia operando con algoritmi divisi o agglomerativi.
Il maggior limite di questo tipo di algoritmi risulta essere il rumore di fondo, rumore e eccezioni nel dataset rendono infatti più complesso il processo; in più nella procedura gerarchica non risulta definito un metodo di correzione in corso d'opera, e risulta quindi che una volta che un oggetto viene assegnato ad un cluster questi algoritmi tendono a non considerarlo più; dal punto di vista prestazionale risulta, infine, che questi algoritmi sono molto pesanti nel calcolo; rendendo la loro applicabilità in dataset molto ampi molto limitata; esistono però implementazioni recenti che portano modifiche ai classici algoritmi, permettendo così il loro utilizzo anche su insiemi di dati estesi.

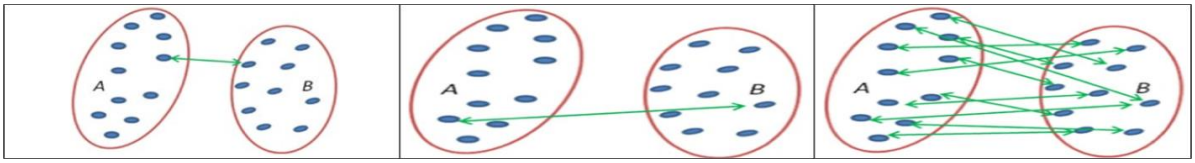


Figura 9: Possiamo vedere i tipi di linkage elencati[1], in particolare a sinistra abbiamo il single linkage, al centro un esempio di complete linkage mentre a destra una rappresentazione dell'average linkage.

4.1.1.Single Linkage

Come detto in precedenza la principale differenza tra algoritmi gerarchici è il tipo di *linkage* che si va a considerare, cioè come si stima la distanza tra due cluster differenti, il *single linkage* noto anche come *nearest neighbour clustering*; è il più semplice metodo di *linkage*, in questo caso infatti si vanno a considerare i due elementi più vicini tra i membri dei due cluster; o nel caso si utilizzi una misura di similarità; la similarità dei due cluster è la massima similarità tra membri dei due cluster.

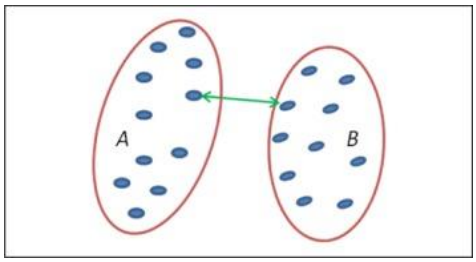


Figura 10:Single Linkage[1]

4.1.2.Average Linkage

In questo caso il *linkage* tra due cluster è noto anche come metodo della minima varianza, e la distanza tra due cluster risulta la media delle distanze dei membri dei due cluster, considerando quindi ogni combinazione.

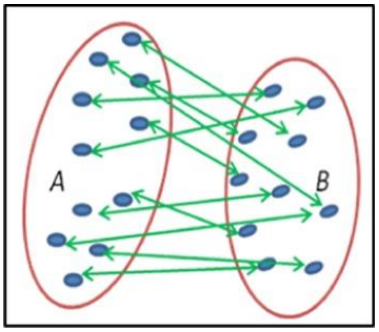


Figura 11:Average Linkage[1]

4.1.3.Complete Linkage

Questo metodo di *linkage*, noto anche come *linkage* del diametro, o del massimo; si basa sulla distanza massima tra i due cluster; quindi, considera i due elementi più lontani, o meno simili nel caso si stia sfruttando una misura di similarità.

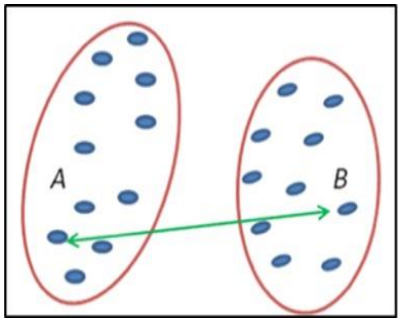


Figura 12: Complete Linkage[1]

4.1.4.Clustering Gerarchico Migliorato

Il principale limite degli algoritmi gerarchici, come detto, risulta essere l'immodificabilità dei link una volta prodotti, sono stati sviluppati però alcuni algoritmi che sfruttano metodi gerarchici ma che apportano miglioramenti e che quindi permettono di risolvere alcuni dei problemi sopra elencati:

i) Clustering Gerarchico a Riduzione Iterativa Bilanciata (BIRCH)[18]

Questo algoritmo si basa sull'uso di una terna nota come *Clustering Feature (CF*) composta dal numero di elementi del cluster, dalla somma lineare degli attributi del cluster, e dalla somma in quadratura degli stessi attributi; queste terne indicate come *(n,LS,SS)*, sono raccolte in un diagramma ad albero permettendo di alleggerire la memoria che non deve computare costantemente l'intero cluster ma solo questi rappresentanti; raggiungendo una complessità di $O(N)$ molto meglio rispetto ai classici algoritmi che comportano un $O(N^2)$; e rendendo questa implementazione ottima per grandi database.

ii) Clustering Robusto tramite Collegamenti (ROCK)[19]

Questo metodo viene sfruttato per il clustering di dati categorizzati, in particolare sfrutta i collegamenti che si vengono a formare tra i cluster al posto della distanza tra gli stessi per definire quali cluster accorpare; viene infatti impostata una soglia per la similitudine tra i punti, e se due punti superano la soglia allora sono vicini, questo significa che tra loro ci sarà un link, per decidere quali cluster accorpare si vanno a cercare quelli con più link in comune; questo algoritmo può quindi essere implementato per dati che non sono dotati di una distanza.

iii)Clustering basato su Rappresentanti (CURE)[20]

Questo algoritmo risulta ottimale nel trattare grossi set di dati, risulta anche robusto nel trattare le eccezioni, e in più permette come risultati anche cluster di forme non sferiche, risolvendo così un altro classico problema dei metodi gerarchici; ossia una particolare predilezione nel formare strutture sferiche; in queste tematiche risulta effettivamente più preciso di *BIRCH* anche se a livello prestazionale è meno efficiente, $O(N^2 log\ (N))$ contro un $O(N)$.

Dal punto di vista dell'implementazione questo algoritmo si basa sulla scelta di una serie di punti ben distribuiti che servono da rappresentati del cluster, alleggerendo così il peso computazionale; questi punti vengono poi contratti verso il centro del cluster, rendendo meno pesante l'effetto dei dati estremi, e permettendo anche di agire meglio su cluster non sferici.

iv)Clustering tramite modelli dinamici (CHAMELEON)[21]

Questo metodo basa l'accorpamento tra cluster non solo sulla vicinanza tra i cluster, ma tiene anche conto dell'interconnessione tra questi; andando quindi a studiare sia le connessioni interne tra i due cluster, sia la vicinanza tra i punti esterni degli stessi; riuscendo ad adattarsi meglio ai dati rispetto ai classici algoritmi statici. Questo algoritmo risulta però limitato dal punto di vista dell'applicabilità; quanto riesce ad operare solo su set di dati a basse dimensioni, risultando inapplicabile a dimensioni superiori.

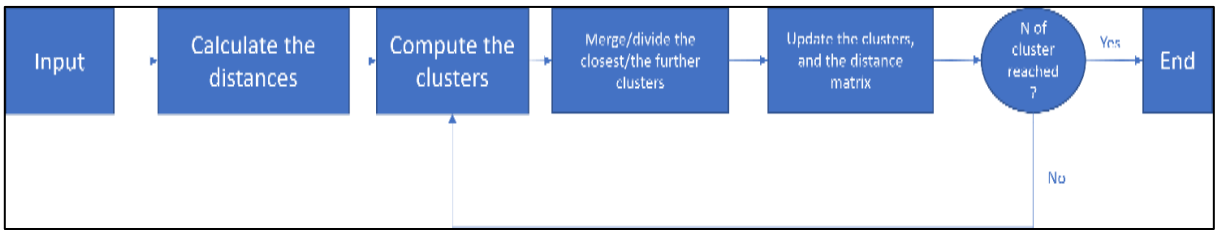


Figura 13: In figura, un diagramma di flusso esplicativo per un generico algoritmo gerarchico

4.2. Clustering Partizionale

Al contrario dei metodi gerarchici gli algoritmi partizionali partono da presupposti molto vicini alle nozioni di geometria; infatti vanno ad operare la divisione e il raggruppamento dei dati attraverso la valutazione della distanza da un determinato punto noto come centroide, il quale può appartenere o meno al dataset; uno svantaggio di questo tipo di algoritmi è la necessità di dotare l'inizializzazione di un numero di cluster *k* determinato, che risulta necessario per posizionare i centroidi e quindi operare la tassellatura dello spazio; anche se al contrario delle alternative, necessitando della dotazione di una distanza, risultano facilmente espandibili su più dimensioni, in quanto il tutto si riconduce ad un problema di minimizzazione dove si vuole minimizzare la distanza dei punti dai centroidi. In questa famiglia di algoritmi troviamo quello che risulta essere l'algoritmo più usato cioè *K-Means*; ma esistono anche alcune varianti che partono dallo stesso metodo di partizione ma ammettono ad esempio una dinamica fuzzy, permettendo quindi di avere elementi contenuti non esclusivamente in un cluster; oppure abbiamo variazioni dell'algoritmo *K-Means* che non randomizzano completamente le posizioni dei centroidi (*K-Means++*), che vincolano i centroidi ad essere punti del set (*K-Medoids*) o infine che vanno ad utilizzare la mediana invece che la media nella tassellatura (*K-Medians*).

Tutti questi algoritmi risultano facilmente applicabili su set numerici, in quanto basta dotare il set di una distanza (solitamente quella euclidea) con cui si andrà a tassellare lo spazio; risultano molto usati anche per alcune conseguenze concettuali alla loro applicazione, prima fra tutte la creazione di un diagramma di Voronoi sullo spazio; vengono poi molto sfruttati nel machine learning in quanto sono concettualmente affini alla classificazione basata sul vicino più prossimo.

Presentano ovviamente dei limiti, in primis abbiamo che il problema di ottimizzazione che porterebbe al trovare una soluzione risulta un problema NP-arduo, e quindi è necessario utilizzare una soluzione approssimata, rappresenta in effetti dall'algoritmo di Lloyd[22], che però porta spesso ad ottenere solo un punto ottimale locale per il problema. Questi algoritmi risultano quindi molto utili in quanto facilmente scalabili a dimensioni superiori, e soprattutto risultano ottimi per dataset in cui si prestano cluster sferici e quindi i dati risultano ben separabili; d'altro canto tra i problemi troviamo in primis la mancata risolvibilità del problema, che risulta NP-Hard, ma anche una grossa sensibilità alle eccezioni e la dipendenza nella buona riuscita dall'inizializzazione con un numero di cluster coerente da parte dell'utente, risultando spesso quindi algoritmi non robusti.

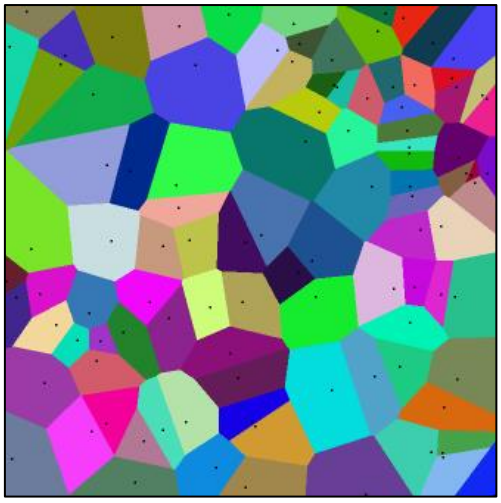


Figura 14: In figura vediamo un diagramma di Voronoi, in particolare abbiamo una tassellatura di un insieme casuale di punti[23].

4.2.1.K-Means

K-Means risulta l'algoritmo più comune quando si portano avanti operazioni di analisi dati basate sul clustering di dataset; per questo risulta molto studiato e ben testato[24].
La classificazione dei dati avviene minimizzando una certa funzione *J* e posizionando quindi il dato considerato nel cluster che rende migliore questa funzione, il numero dei cluster è richiesto in inizializzazione all'utente, rendendo quindi l'algoritmo potenzialmente meno robusto di una controparte automatica.
Possiamo definire la funzione *J*:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Dove abbiamo che || || rappresenta una qualsiasi distanza ricercata tra un generico punto $x_i^{(j)}$ e il centro del cluster c_j, n è il numero di punti presenti e k il numero di cluster; il processo di clustering parte dall'assegnazione dei centroidi, i punti vengono poi accoppiati ad un centroide minimizzando *J*, i centroidi vengono poi valutati, e viene poi valutato se è necessario spostarli, nel caso non sia necessario il processo risulta concluso, altrimenti si ripete il passaggio precedente.
Da questo algoritmo sono nate poi alternative come l'algoritmo *LBG* usato nella quantizzazione vettoriale per comprimere il segnale.
Come già detto questi algoritmi sono molto diffusi, ma risultano limitati dal fatto che non esiste un modo preciso di trovare la partizione iniziale e il numero di cluster presenti, e risulta problematica anche la gestione di dataset rumorosi o con molte eccezioni, che vanno a sbilanciare la struttura sferica che verrebbe a formarsi.

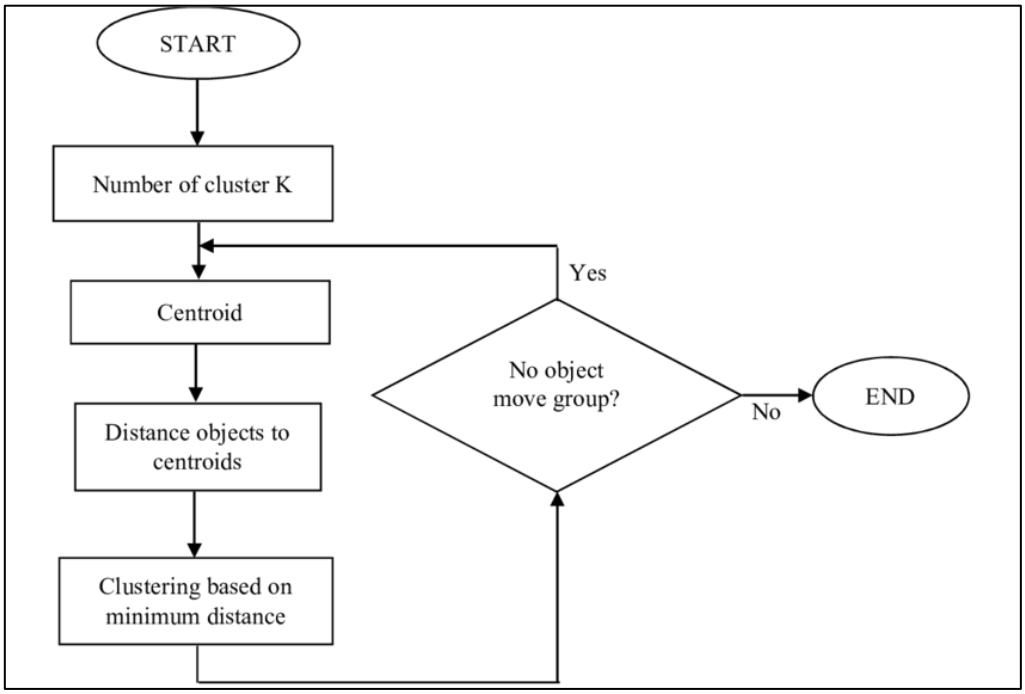


Figura 15: In figura vediamo un diagramma di flusso esplicativo del comportamento di K-Means[25]

4.2.2.Fuzzy C-Means

FCM è un particolare algoritmo non esclusivo che parte dagli stessi presupposti di K-Means ma permette ad un punto di essere associato a più di un cluster, definito in un primo momento da Dunn[26] e sviluppato successivamente da Bezdek[27] questo algoritmo si basa, similmente a K-Means sulla minimizzazione di una funzione *J:*

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^{m}\left\|x_i - v_j\right\|^2 \; con \; 1 < m < \infty$$

Dove *m* indica l'esponente della matrice di partizione fuzzy che va a definire il grado di sovrapposizione fuzzy cioè su quanto siano sfocate le separazioni tra i cluster, cioè indica il numero di elementi che sono associati a più di un cluster, x_i è la coordinata iesima dell'elemento N-dimensionale; mentre v_j è il centro del jesimo cluster; u_{ij} indica il grado di appartenenza di x_i nel cluster *j:*

$$u_{ij} = \left[\sum_{l=1}^{c}\left(\frac{\|x_i - v_j\|}{\|x_i - v_l\|}\right)\right]^{-1} \text{e} \quad v_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Anche il funzionamento di questo algoritmo è simile a quello di K.-Means, infatti, avremo in primo luogo il posizionamento dei i centri iniziali, i dati vengono inseriti nei cluster così formati; viene successivamente rivalutata la matrice u_{ij} che permette di ridefinire i centri; questo viene ripetuto fino a convergenza o finché non si sfora il numero massimo di iterazioni previsto.
Anche questo algoritmo, come tutti i metodi partizionali, soffre la presenza di molto rumore o di eccezion; e in più risulta dipendente dal numero di cluster che si inserisce in fase di inizializzazione; per risolvere questi problemi sono stati proposti alcuni metodi, in primis il metodo della montagna[28] che si basa sulla costruzione di una funzione montagna dai dati, che poi viene distrutta per ottenere i centri iniziali; un'alternativa è l'aggiunta dinamica di cluster prototipo durante l'esecuzione per trattare quei punti che non sono ben gestiti normalmente[29], un'altra possibilità è la modifica della nozione di prossimità per trattare le eccezioni[30].
Infine possiamo considerare un'implementazione possibilistica dell'algoritmo partizionale, basata cioè sull'assunzione per cui l'appartenenza ad una categoria consiste nella compatibilità dei punti con l'archetipo di classe che andiamo a considerare; questa interpretazione proposta da Krishnapuram e Keller[31] porta quindi la matrice da $u_{ij} = 1$ a $u_{ij} > 0$; ponendo quindi le condizioni:

$$u_{ij} \in [0;1] \; con \; 1 \le i \le N \; ; 1 \le j \le C$$

$$\exists j, u_{ij} > 0 \; \forall \; i$$

$$0 < \sum_{i=1}^{N} u_{ij} < N \; con \; 1 \le j \le C$$

Entro queste condizioni è quindi possibile un'interpretazione possibilistica di questo algoritmo.
FCM risultano vantaggiosi rispetto a K-Means nella trattazione di dati incompleti o molto rumorosi, nell'analisi di informazioni provenienti da media diversi, nell'operare in relazione ad un'interazione umana e infine risultano più rapidi nel proporre una soluzione approssimata; d'altro canto, però il tempo associato ad una computazione completa risulta decisamente peggiore rispetto K-Means che quindi risulta operare nel complesso più rapidamente[32].

4.3. Ulteriori Metodi di Clustering

Fino ad ora abbiamo discusso le principali famiglie di metodi di clustering; è però vero che abbiamo trascurato alcuni gruppi importanti, come ad esempio il clustering su grafi o il clustering basato su modelli oppure il clustering su distribuzioni; di seguito elenchiamo alcuni di questi gruppi in modo da proporre una trattazione più completa; molto spesso questi algoritmi partono da concetti affini a metodi gerarchici o a metodi partizionali, questi concetti di base vengono poi modificati nell'applicazione; venendo utilizzati per studiare particolari strutture matematiche, oppure approcciandosi al tema in modo differente, come nel caso delle dinamiche collaborative o dei metodi di sovrapposizione, che risultato in tutto e per tutto metodi partizionali.

4.3.1.Clustering di Grafi

Il clustering su grafi è un metodo alternativo di rappresentazione dei dati, in cui si visualizzano le istanze del dataset come nodi di un grafo, e si vuole accorpare i dati in modo che i collegamenti siano massimi tra membri dello stesso gruppo, e minimi verso gli altri agglomerati.

Un esempio di algoritmo che opera su queste strutture è *MST*[33]*;* questo è un algoritmo che si basa sul minimo albero ricoprente, cioè un grafo formato da tutti i nodi del grafo iniziale e dall'insieme minimale di collegamenti che però permetta di aver un solo percorso per raggiungere da ogni nodo; esistono anche metodi alternativi, ad esempio basati sul *limited neighbourhood*[34], il comportamento però risulta comune tra questi metodi, che risultano molto utili per avere una rappresentazione dei dati sotto forma di grafo, ma risultano deboli nel trattare elementi estremanti e nel gestire la sovrapposizione di cluster.

Solitamente per valutare le soluzioni viene utilizzato un modello a blocchi stocastici; in cui si suppone di dover scomporre un set di n nodi in l gruppi; ognuno formato da n/l nodi; ognuno di questi sarà collegato con un nodo interno al cluster con una certa probabilità p e con un nodo esterno con una probabilità $r<p$. Risulta che se $p - r \geq n - \frac{1}{2} + \varepsilon$ con ε costante allora avremo una soluzione ottimale con una probabilità $1 - \exp\left(-n\Theta(\varepsilon)\right)$[35]. Questi algoritmi permettono di fatto di separare una rete in sottoreti, noi solitamente poi selezioniamo le scomposizioni che portano a sottoreti dense.

Gli algoritmi di clustering che operano su grafi hanno permesso lo sviluppo di una teoria molto potente, cioè gli algoritmi di clustering spettrale, questo metodo infatti si basa sull'utilizzo di proprietà caratteristiche degli autovalori di alcune matrici che possiamo costruire a partire dai dati, queste matrici traggono origine da un grafo che valuta la similarità tra i dati, per questo i metodi di analisi spettrale risultano legati agli algoritmi definiti sui grafi.

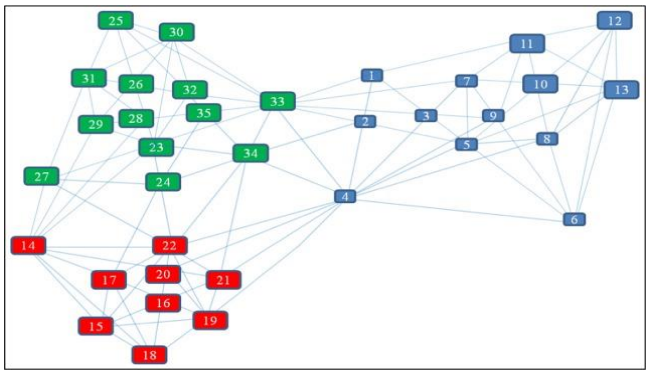


Figura 16: Potenziale risultato prodotto da un algoritmo di clustering applicato ad un grafo[1].

4.3.2.Clustering tramite Griglia

Gli algoritmi che operano su una griglia vanno a scomporre il dataset in una fitta rete di celle; dopo aver affibbiato i dati alle rispettive celle viene valutata la densità di ciascuna cella, eliminando quelle sotto una certa soglia, i dati vengono poi accorpati considerando le celle vicine. Questi algoritmi risultano molto utili in quanto non necessitano del calcolo delle distanze e quindi eccellono nella rapidità di calcolo, in più risulta molto semplice definire quali cluster sono vicini. Tra i metodi di questo tipo un esempio è *STING (statistical information on grid approach)*[36] questo algoritmo va a dividere la classificazione in vari livelli, ogni cella viene infatti a sua volta divisa in sotto-celle nel livello successivo, e di ogni cella vengono calcolati attraverso alcune distribuzioni vari parametri statistici; un'alternativa risulta *CLIQUE*[37] il quale riesce in automatico a trovare i sottospazi di spazi di dati a dimensione maggiore che permettano la miglior classificazione dei dati.

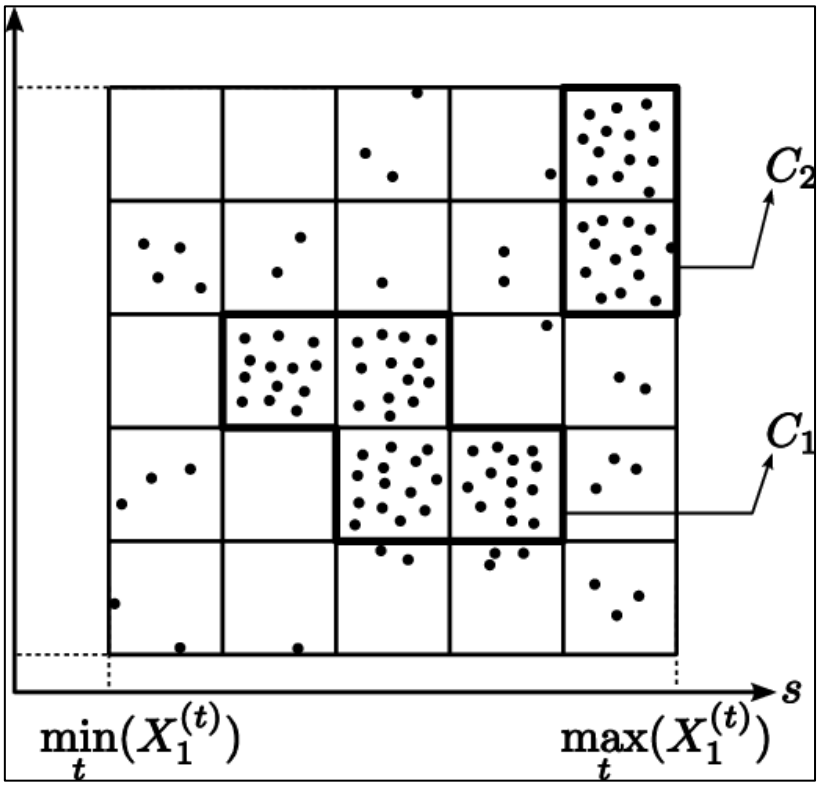


Figura 17: Esempio di clustering tramite griglia bidimensionale[38]

4.3.3.Spectral Clustering (normalizzato e non)

Come detto in precedenza l'analisi spettrale dei dati è uno strumento molto potente quando applicabile, non richiedendo assunzioni stringenti sulla forma dei cluster rendendoli più applicabili, al contrario di altri algoritmi ottimali su cluster formati da insiemi convessi nello spazio; d'altro canto però questo tipo di algoritmi richiede che sia prodotta una matrice di similarità rappresentativa del set, e per farlo è necessario costruire un grafo di similarità che risulta essere un procedimento non triviale, che se ben scelto rende il problema del raggruppamento dei dati un semplice problema lineare; evitando quindi la possibilità di bloccare l'algoritmo in un punto ottimale locale ed evitando anche la necessità di testare l'algoritmo con varie inizializzazioni per valutare gli iper-parametri.

Questi algoritmi[39] si basano sulla costruzione di una matrice di similarità: $s_{ij} = sim(x_i, x_j)$ dove *sim* risulta una qualsiasi funzione di similarità; questa matrice risulta simmetrica e non negativa per le proprietà delle misure di similarità; una volta costruita questa matrice supponendo di voler scomporre in *k* cluster il mio dataset andrò a costruire il mio grafo, scegliendo tra alcune possibilità quella che meglio si addice al mio problema, da questo grafo sarà poi necessario estrapolare la matrice laplaciana[40] di cui calcoleremo i primi *k* autovettori (generalizzati o meno a seconda si stia usando l'algoritmo normalizzato o no) $u_{1,\dots,}u_k$ con cui possiamo costruire una matrice $R^{n\times k}$ che ha per colonne i nostri autovettori consideriamo ora y_i il vettore di R^k associato alla riga *i*; andiamo ora ad raggruppare usando K-Means i punti rappresentati da questi y_i ottenendo i cluster $C_1, \dots, C_k$; e come output otterremo i cluster $A_1, \dots, A_k \; con \; A_i = \{j | y_j \in C_i\}$. Abbiamo descritto come i due metodi (normalizzato e non) differiscano dalla necessità di utilizzare nel primo caso gli autovalori generalizzati, questo significa considerare quei vettori u che soddisfino l'equazione $Lu = \lambda Du$ invece che il classico $Lu = \lambda u$; in questa espressione la matrice *D* indica la matrice di grado[41] cioè la matrice diagonale che associa ad ogni vertice il suo grado, ossia il numero di archi a cui si collega; questa matrice come la matrice laplaciana si possono estrapolare dal grafo.

4.3.4.Clustering Basato su Modelli

Il clustering basato su modelli risulta uno strumento molto efficace per analizzare i dati quando questi sono aderenti con il modello di riferimento; in questo caso si tende a far aderire al concetto di cluster il significato di classe o concetto; i metodi induttivi di questo tipo più usati sono gli alberi decisionali e le reti neurali:

i) *Alberi Decisionali:*

Gli alberi decisionali sono strutture gerarchiche in cui ogni foglia rappresenta una categoria e genera una descrizione probabilistica di quella categoria, esistono numerosi algoritmi che permettono di modellizzare dati non etichettati in modo da fargli assumere questa forma ad albero; il più usato risulta essere *COBWEB* in cui ogni categoria o concetto è composto da un insieme di oggetti, i quali sono descritti come lista di proprietà binarie; lo scopo di questo algoritmo è generare una struttura sia in grado di predire bene i valori nominali delle variabili nel caso sia associata ad un cluster; il problema di questo algoritmo è però la sua non applicabilità su dataset troppo grandi[44].

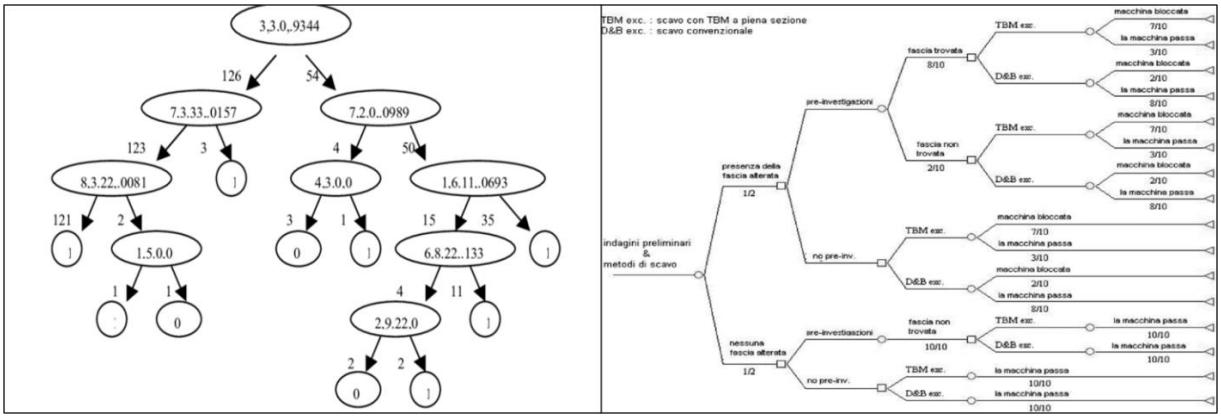


Figura 18:In figura vediamo due esempi di alberi decisionali, l'elemento a sinistra è un grafico costruito su dati relativi al tumore al seno[42], mentre a destra abbiamo un albero costruito con dati relativi alle analisi preliminari per la costruzione i gallerie[43].

ii) *Reti Neurali:*

Nelle reti neurali i cluster sono rappresentati da neuroni, collegati su livelli con altri neuroni che formano l'input e a cui il livello passa l'output; le connessioni sono pesate con valori inizializzati a random, ma che vengono modificati tramite apprendimento adattivo.

L'algoritmo di clustering più popolare che si appoggi al concetto di rete neurale è *SOM (Self Organizing Map)*[45]; questo algoritmo produce una rete neurale su singolo livello e basa l'apprendimento su un metodo "*winner takes all*" i neuroni, infatti, competono per l'istanza in atto, il neurone che ha il vettore di peso associato più vicino al valore dell'istanza in analisi risulta vincitore; in questo modo il vincitore e i vettori vicini vengono modificati imparando. Questo algoritmo trova molti utilizzi; in particolare nella quantizzazione vettoriale, nella visualizzazione e nell'estrazione di proprietà se usato nell'ambito dell'analisi attraverso il clustering.

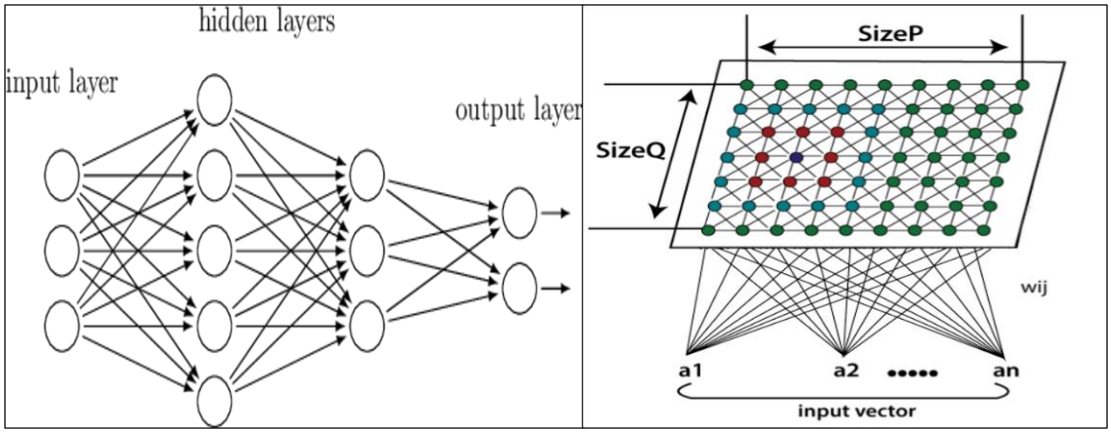


Figura 19:Due esempi di reti neurali, a sinistra abbiamo una generica struttura a livelli nascosti, mentre a destra possiamo vedere una SOM

Questo algoritmo risulta vantaggioso in quanto non dipende nell'apprendimento dall'ordine dei dati in input e in più risulta ottimo nell'approssimazione della densità dei dati di input; nonostante ciò però comporta alcuni problemi, in primis il fatto che come per K-Means questo algoritmo necessita in inizializzazione di conoscere il numero di cluster che vanno a formare il lattice della rete, in più risulta anche che un *SOM* completamente trainato può errare nella rappresentazione della densità dei dati in input[46] in particolare le zone più dense potrebbero risultare sottorappresentate e quelle meno dense sovra rappresentate[47].

4.3.5.Clustering Multi-Obbiettivo

Nel clustering multi-obbiettivo, più approcci vengono portati avanti contemporaneamente, ad esempio se consideriamo l'algoritmo *MOCK*[48,49] avvengono contemporaneamente l'ottimizzazione della compattezza dei cluster, che risulta l'obbiettivo primario, e della connettività tra i cluster, come obbiettivo secondario; nel caso invece di *Pareto*[50] questi due approcci sono portati avanti con egual peso; infine, nell'algoritmo *MACE*[51] viene usato l'algoritmo *MOCK* insieme ad un particolare operatore di crossover che permette il clustering complessivo.

4.3.6.Clustering Basato su Distribuzioni

È stato descritto in vari articoli[47,52] come sia possibile trattare il clustering supponendo che i dati siano estrapolati da qualsivoglia distribuzione di probabilità e anche supponendo che la distribuzione totale dei dati sia l'insieme di varie distribuzioni differenti; in quest'ottica i punti possono essere ottenuti da varie distribuzioni di probabilità, oppure dalla stessa ma con parametri modificati; lo scopo di questi algoritmi è quindi trovare i cluster di punti provenienti dalla stessa distribuzione e la distribuzione stessa.
Esistono numerose proposte per risolvere il problema, a partire da *AUTOCLASS*[53] che risulta descrivere bene varie distribuzioni (gaussiana, bernoulli, poisson, etc), in alternativa troviamo *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*[54] che descrive cluster di forma arbitraria e risulta efficace per database molto estesi; troviamo altre proposte ma tra tutte l'algoritmo *EM (Expectation Maximization)*[55] risulta il più usato; per questo algoritmo definiamo una funzione *log-like* che andrà massimizzata:

$$ln\,(p(X|\Theta)) = ln\left(\sum_{Y} p(X,Y|\Theta)\right)$$

dove indichiamo con *X* il set di tutti i dati osservati, *Y* risulta il set di tutte le variabili latenti, il set completo risulta quindi composto dalla coppia *(X,Y)*; la distribuzione $p(X,Y|\Theta)$ sarà pilotata da un determinato insieme di parametri Θ.
Il procedimento parte con l'inizializzare i parametri Θ^{old}; successivamente viene valutata $p(X,Y|\Theta^{old})$ da cui vengono poi ristimati i parametri Θ^{new}, infine viene valutata la convergenza, che se non è raggiunta comporta la ripetizione dei passaggi precedenti.
Il maggior limite di questo algoritmo risulta essere la forte sensibilità ai parametri di inizializzazione; infatti, l'inizializzazione sbagliata per esempio di una singola matrice di covarianza può inficiare fortemente la convergenza dell'algoritmo; portandola ad un massimo locale e rallentandola.

4.3.7.Clustering Basato su Metodi Evolutivi

La dinamica evolutiva viene applicata nell'ambito dei metodi di clustering attraverso alcune famiglie di processi: programmazione evolutiva, strategie d'evoluzione, ottimizzazione degli sciami particellari e ottimizzazione a colonia di formiche sono solo alcuni esempi; di fatto però il metodo più diffuso risulta essere quello degli algoritmi genetici (*GA*).
Questi metodi condividono in realtà un approccio comune, essi infatti partono col definire una popolazione di potenziali soluzioni, solitamente generate casualmente, in cui ogni istanza rappresenta una partizione in *k* gruppi del dataset, viene poi associata al problema una funzione che dia una valutazione al risultato, solitamente si sceglie una funzione inversamente proporzionale al valore di errore quadrato; vengono poi applicati gli operatori evolutivi: ricombinazione, selezione e mutazione per cercare la miglior soluzione; questo processo viene ripetuto finché non si raggiunge una precisione soddisfacente.

Gli algoritmi genetici risultano i metodi evolutivi più usati nel clustering, soprattutto perché producono un risultato trattabile in stringa binaria; in questo tipo di algoritmi sono fondamentali gli operatori evolutivi, in primo luogo la selezione, che permette anche di dare un taglio probabilistico al problema; le soluzioni con fitness più alto possono con probabilità maggiore essere trasmesse alla generazione seguente.
Il maggior problema di questo tipo di algoritmi è la dipendenza dai parametri di tuning iniziali come dimensione della popolazione, rateo o tipo di crossover, probabilità di mutazione etc.; anche piccole variazioni di questi parametri possono inficiare fortemente la bontà della convergenza ottenuta, e il tempo richiesto per raggiungere una soluzione.

4.3.8.Clustering Basato su Ricerca

Gli algoritmi di ricerca sono solitamente usati per ricercare un valore ottimale di una funzione nota come funzione obbiettivo, che funziona da criterio per definire i cluster.
Esistono due tipi principali di algoritmo di ricerca, i metodi che si appoggiano su concetti evolutivi e i metodi totalmente deterministici i quali garantiscono una soluzione ottimale portando avanti un'enumerazione esaustiva.
Esistono metodi stocastici di ricerca sia in forma sequenziale che in forma parallela; mentre per i metodi evolutivi risulta prediletta la forma parallela. *SA (Simulated Annealing)* è un particolare algoritmo che segue un approccio stocastico, è strutturato appositamente per evitare soluzioni locali e anzi recuperare nel caso ci si trovasse in una situazione del genere, questo viene garantito dalla possibilità di accettare, con una certa probabilità, una soluzione peggiore come iterazione successiva, probabilità e governata da un parametro critico chiamato temperatura, in analogia con il fenomeno della ricottura nei metalli. Questo parametro permette all'algoritmo *SA* di essere molto preciso nella soluzione ma rende questi metodi molto dispendiosi, in quanto la temperatura va abbassata lentamente da iterazione a iterazione.
L'algoritmo *Tabu Search*[56], risulta un metodo di clustering basato sulla ricerca stocastica, che, come *SA*, permette di superare i limiti ottimizzazione locale, rilasciando e imponendo vincoli ed esplorando così tutto lo spazio delle soluzioni, evitando di bloccarsi su punti ottimali locali[57,58].

4.3.9.Clustering di Sovrapposizione

Solitamente il clustering partizionale indica sia metodi esclusivi che metodi che considerano la possibilità di sovrapposizione; si parla di metodi esclusivi (come K-Means) nel caso in cui ogni oggetto appartenga a un solo cluster; nel caso in cui un oggetto possa essere associato a più di un solo cluster si parla invece di algoritmi di sovrapposizione (come fuzzy C-Means).

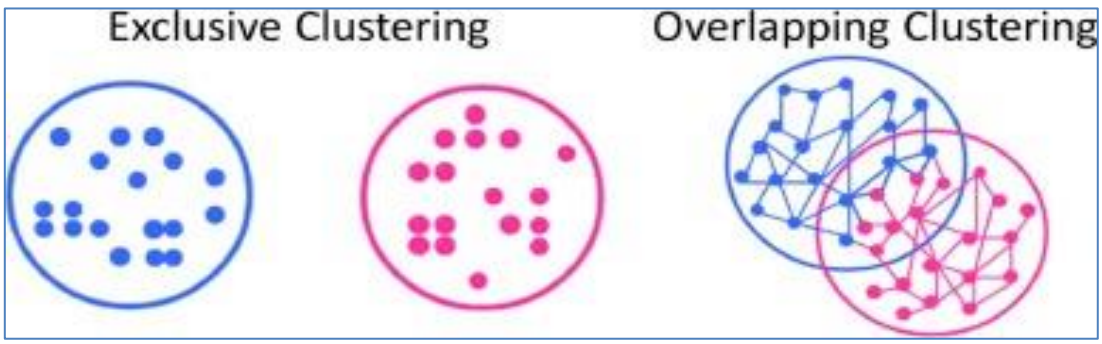


Figura 20: In figura[59] vediamo la differenza tra clustering esclusivo e clustering di sovrapposizione

Questi algoritmi stanno trovando largo impiego al giorno d'oggi in quanto è sempre più importante il processo di *community detection*, che permette di scomporre una rete in una serie di sottoreti in modo che i collegamenti siano più densi all'interno di una comunità e meno densi verso l'esterno, per una rete di nodi è semplice visualizzare questo concetto ma nella realtà molto spesso un elemento appartiene a più categorie, ad esempio un individuo ha contatti frequenti sia con la famiglia che con svariati gruppi di amici; per implementare ciò vengono usati i cosiddetti *overlapping nodes*[60] trovare questi nodi risulta essere l'obbiettivo del processo di *overlapping community detection*, insieme a visualizzare le comunità a cui si associano.
I metodi per gestire questo problema possono essere divisi in due macrocategorie: i metodi basati sui nodi e i metodi basati sui collegamenti; i primi tendono a separare direttamente i nodi presenti in comunità, i metodi basati sui collegamenti, invece, si appoggiano sull'idea per cui ogni collegamento sia unico nella rete; perciò, questi algoritmi vanno a raggruppare prima gli archi tra i nodi per poi mappare i nodi in questa struttura costruita per gli archi, raccogliendo tutti i nodi toccati dai collegamenti di uno stesso cluster[61]. Sviluppi recenti hanno mostrato come i metodi basati sui collegamenti si dimostrino superiori nell'analisi di comunità multi-scala; nonostante però comportino una grande complessità computazionale e presentino di fatto alcuni bias nella scoperta delle comunità. Un'implementazione recente risulta *GaoCD*[60], un algoritmo basato sull'idea di unicità degli archi, che parte quindi da un'idea *link-based*, che però fa uso di un algoritmo genetico per produrre un risultato; questo algoritmo è stato testato sia con reti artificiali che reali mostrando ottimi risultati.

4.3.10. Clustering Basato su Collaborazione Fuzzy

Questa famiglia di algoritmi risulta uno sviluppo abbastanza recente, in particolare si basa sulla possibilità di scomporre i dataset in sottoinsiemi differenti con lo scopo di trovare una struttura a partire da uno di questi sottoinsiemi che risulti simile o quantomeno compatibile con gli altri sottoinsiemi; in particolare identifichiamo due principali famiglie di algoritmi collaborativi: gli algoritmi verticali e gli algoritmi orizzontali; la differenza tra queste due tipologie risulta essere l'approccio con cui si pongono verso la divisione del dataset, in particolare risulta che per gli algoritmi verticali la scomposizione è fatta in modo che ogni subset di feature contenga tutti i pattern; per l'approccio verticale invece i sottoinsiemi di pattern sono prodotti in modo che i pattern di ogni sottoinsieme siano dotati di tutte le possibili feature.

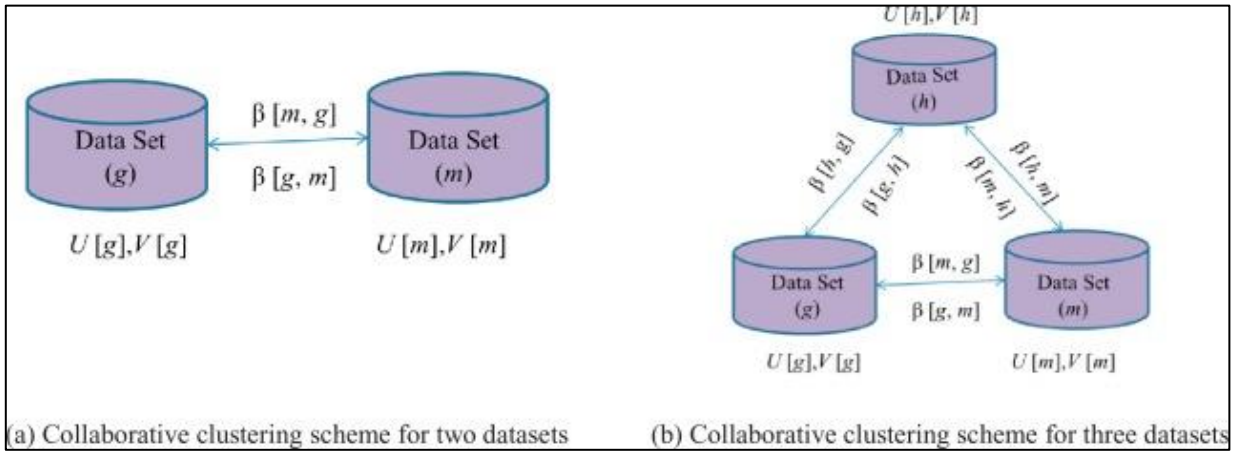


Figura 21: In figura notiamo una rappresentazione del funzionamento della collaborazione nel clustering[1]

Concentrandoci sui metodi orizzontali possiamo definire la funzione obbiettivo come:

$$Q[l] = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^2[l]d_{ij}^2[l] + \sum_{\substack{m=1 \\ m\neq l}}^{p} \beta[l,m] \sum_{i=1}^{N}\sum_{j=1}^{n} \{u_{ij}[l] - u_{ij}[m]\}^2 d_{ij}^2[l]$$

Dove usiamo β per indicare un parametro definito dall'utente, positivo ed è definito come proprietà del dataset, $\beta[l,m]$ indica quindi il coefficiente di collaborazione per i sottoinsiemi l e m; c risulta invece il numero di cluster, *p* rappresenta il numero di sottoinsiemi che consideriamo, N rappresenta il numero di pattern del dataset, *n* il numero di features, d rappresenta una distanza euclidea tra pattern e prototipi di cluster e u la matrice di partizione. Il processo di soluzione del problema si basa quindi su una prima soluzione locale, che porta alla ricerca di una matrice *u* per ottimizzare localmente il sottoinsieme considerato, questa viene poi messa in collaborazione globalmente con le altre soluzioni locali, per ottenere una soluzione globale.

4.4. APPLICAZIONI

Gli algoritmi di clustering sono strumenti molto potenti in quanto permettono di trattare in modo più semplice insieme di dati che risulterebbero altrimenti ingestibili; di fatto i campi in cui questi metodi vengono sfruttati sono svariati, i principali ambienti di applicazione spaziano dalla bioinformatica, alle analisi di mercato; il principale vantaggio di questi algoritmi è infatti la grande adattabilità che garantiscono, in quanto per poterli applicare basta solamente che i dati siano rappresentabili tramite proprietà, descrivibili numericamente o meno; permettendo quindi un'applicabilità teoricamente infinita e come sappiamo la possibilità di raggruppare i dati che abbiamo in categorie in base alla somiglianza o meno, sotto determinati criteri, è uno strumento potentissimo in un'era in cui la possibilità di raccogliere dati è così grande.

4.4.1.Applicazioni nella Ricerca: Omics

I recenti sviluppi della ricerca nell'analisi del codice genetico hanno dato il via ad una collaborazione estesa tra informatica e biologia.

La bioinformatica trova infatti ampio utilizzo in numerose branche della biologia, genericamente indicate come *"-omiche"* (più note con il nome inglese *omics*); questi ambiti di ricerca hanno in comune la necessità di operare con librerie di dati molto estese e quindi necessitano l'applicazione di algoritmi e metodi per semplificare l'analisi, come ad esempio gli algoritmi di clustering; i dati necessari provengono da ambiti molto estesi a partire dalla genomica, che si propone di studiare la struttura, la funzione e l'evoluzione del nostro codice genetico; fino a ottenere una potenziale mappatura del codice, il Progetto Genoma Umano è stato il primo passo in quest'ambito e ha prodotto un'enorme quantità di dati che ancora va ad espandersi; similmente troviamo anche la proteomica, cioè la ricerca di una categorizzazione precisa delle proteine, a partire dalle relazioni tra struttura e funzione, in questo caratterizzati da quantità di dati, questi dati sono poi stati raccolti in numerosi database e vengono costantemente studiati, tramite svariati metodi di clustering e non, producendo tra le altre cose *PPIN (Protein Protein Interaction Networks)* ossia mappe delle interazioni tra le proteine.

Tra le *omics* troviamo principalmente ricerche che si appoggiano sulla biologia, ma metodi simili a quelli proposti dalle materie sopraelencate hanno portato a far coincidere con questa macrocategoria anche materie lontane dalla biologia, come la textomica, ossia la volontà di raggruppare e categorizzare la letteratura scientifica, utilizzando anche in questo caso metodi informatici, come gli algoritmi di clustering.

4.4.2.Applicazioni nell'Analisi Immagini

Un altro ambito di applicazione importante è l'analisi delle immagini, anche in questo caso gli usi sono molteplici, partendo dalla ricerca medica in ambito di *Image Segmentation,* fino all'analisi di immagini col fine di riconoscere oggetti oppure dell'analisi di dati spaziali, come quelli del GIS; in questi tre casi l'analisi dell'immagine parte scomponendo la stessa figura in una serie di vettori descritti da determinate proprietà.
Le differenze principali sono nello scopo che si vuole raggiungere, se nell'analisi di dati spaziali la funzione è l'estrapolazione di informazioni da immagini GPS o da strumenti simili, nel riconoscimento di oggetti, invece, si sfruttano immagini di un oggetto da vari punti di vista per far riconoscere ad un sistema automatico gli oggetti 3D che si trova davanti.
Nell'*Image Segmentation*, si ha la necessità di scomporre le immagini riconoscendo pattern simili all'interno della stessa figura, per fare ciò esistono svariati metodi, tra cui anche alcuni algoritmi basati sul clustering; questo strumento risulta molto potente, possiamo vedere una particolare applicazione, ad esempio, nella combinazione di queste tecniche con i risultati delle Risonanze Magnetiche; l'*image segmentation* può essere usata infatti in questo campo per separare tessuti diversi all'interno dell'immagine permettendo quindi una diagnosi semplificata.

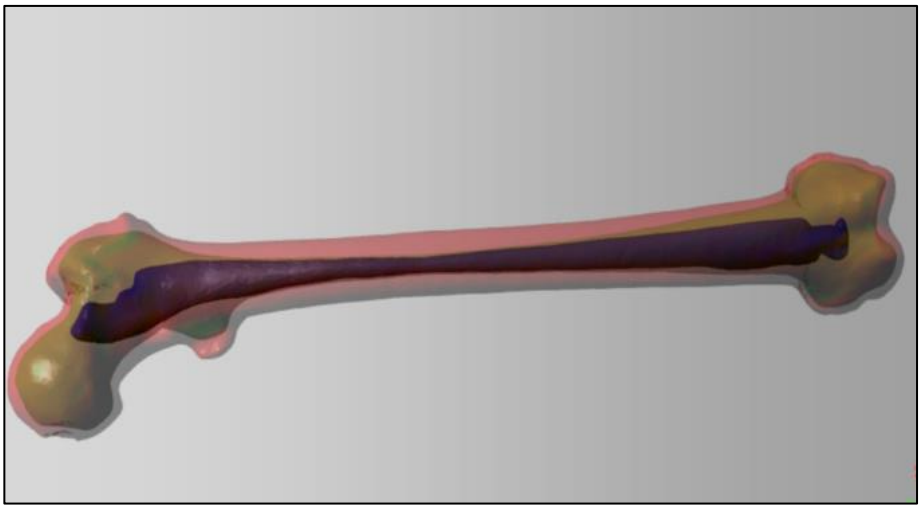


Figura 22: In figura il modello di una potenziale segmentazione di un femore, evidenziati in colore diversi i differenti tipi di tessuto osseo[62].

4.4.3.Applicazioni Esterne alla Ricerca

Infine abbiamo una terza macrocategoria di applicazioni, cioè quegli ambiti esterni alla ricerca in cui però è comunque necessario trattare database estesi al fine di estrapolarne informazioni; sotto questa categoria cade qualsiasi tipo di algoritmo applicato alla raccolta dati, dall'analisi dei big data (*data* e *big data mining* e anche *data reduction*); in particolare la gestione dei *big data*, sta trovando sempre più spazio negli ambiti più svariati; dalla gestione governativa, che si tratti di sicurezza o di innovazione, fino alle necessità dei media odierni, essi infatti necessitano di un modo per gestire il grande quantitativo di dati a cui hanno accesso, in primis, per aver accesso alle informazioni in modo da permettere ai giornalisti di proporre approfondimenti innovativi, ma anche per poter garantire ai consumatori i servizi migliori in base alle preferenze individuali, un ulteriore esempio è rappresentato dal sistema di catalogazione dei libri nelle biblioteche, questo infatti è basato su una determinata codifica andando a raccogliere in modo gerarchico i testi di temi collegati; abbiamo poi che, similmente a come già detto per i media, anche in ambito commerciale algoritmi di questo genere trovano sempre più uso, sia per produrre analisi di tendenze di mercato, che per permettere di definire un obiettivo specifico per le operazioni di marketing, permettendo di ottimizzare le vendite; infine processi basati su algoritmi di clustering risultano fondamentali anche nell'estrazione di informazioni da testi, come ad esempio gli algoritmi atti al riconoscere il plagio nei testi.
Un ambito in cui questi algoritmi trovano molto utilizzo è l'analisi dati volta all'economia, in particolare avremo analisi di tendenze di mercato per evidenziare quando produrre determinati prodotti, e anche per proporre ai clienti i servizi che risultano più desiderati.

4.5. Conclusioni

Una volta definiti tutti questi diversi algoritmi la domanda che sorge spontanea è quale algoritmo risulta il migliore? Quale possiamo applicare in ogni occasione?
La risposta risulta nessuno, u primo esempio è il teorema "No Free Lunch"[63] per cui qualsiasi algoritmo sviluppato per avere prestazioni ottimali in una determinata situazione risulta surclassato in situazioni differenti da algoritmi pensati per quelle specifiche situazioni. Ogni algoritmo, di fatto, risulta ottimale in una particolare situazione, o risulta vantaggioso sotto un determinato punto di vista, mentre, ovviamente comporta svantaggi in altri casi; ad esempio, possiamo avere situazioni in cui il numero di cluster è fissato per necessità dall'utente, e quindi algoritmi in cui la necessità di fissare il numero di cluster in input era stata vista come limite risultano ottimali; può succedere invece che il numero di cluster venga stimato con vari metodi euristici, stocastici o addirittura con metodi evolutivi, richiamando quindi alcuni degli algoritmi che abbiamo citato in precedenza; ragionamenti simili possono essere fatti anche sul tema del data mining, la letteratura si divide e sono presenti numerose proposte alternative di potenziali algoritmi da utilizzare[64–66] ognuna comporta vantaggi e svantaggi su set di dati differenti.
Si conclude quindi che non esiste effettivamente un algoritmo migliore degli altri ma è necessario considerare il problema che si sta affrontando e provare sperimentalmente quale metodo risulta il più efficace nella situazione che si sta trattando.

5. BIBLIOGRAFIA

1. Saxena A, Prasad M, Gupta A, et al. A review of clustering techniques and developments. *Neurocomputing*. 2017;267:664-681. doi:10.1016/j.neucom.2017.06.053
2. Rokach L, Maimon O. Clustering Methods. In: Maimon O, Rokach L, eds. *Data Mining and Knowledge Discovery Handbook*. Springer US; 2005:321-352. doi:10.1007/0-387-25465-X_15
3. Jain AK. Data clustering: 50 years beyond K-means. *Pattern Recognit Lett*. 2010;31(8):651-666. doi:10.1016/j.patrec.2009.09.011
4. Irani J, Pise N, Phatak M. Clustering Techniques and the Similarity Measures used in Clustering: A Survey. *Int J Comput Appl*. 2016;134:9-14. doi:10.5120/ijca2016907841
5. Irani J, Pise N, Phatak M. Clustering Techniques and the Similarity Measures used in Clustering: A Survey. *Int J Comput Appl*. 2016;134(7):9-14. doi:10.5120/ijca2016907841
6. Bravais A. *Analyse mathématique sur les probabilités des erreurs de situation d'un point*. Impr. Royale; 1844.
7. VII. Mathematical contributions to the theory of evolution.—III. Regression, heredity, and panmixia | Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character. Accessed August 22, 2021. https://royalsocietypublishing.org/doi/10.1098/rsta.1896.0007
8. Strehl E, Ghosh J, Mooney R. Impact of Similarity Measures on Web-page Clustering. *Workshop Artif Intell Web Search AAAI 2000*. Published online March 28, 2001.
9. Sørensen TJ. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons.* I kommission hos E. Munksgaard; 1948.
10. Dice LR. Measures of the Amount of Ecologic Association Between Species. *Ecology*. 1945;26(3):297-302. doi:10.2307/1932409
11. Gower JC. A General Coefficient of Similarity and Some of Its Properties. *Biometrics*. 1971;27(4):857-871. doi:10.2307/2528823
12. dos Santos TRL, Zárate LE. Categorical data clustering: What similarity measure to recommend? *Expert Syst Appl*. 2015;42(3):1247-1260. doi:10.1016/j.eswa.2014.09.012
13. Rendon E, Abundez I, Arizmendi A, Quiroz EM. Internal versus external cluster validation indexes. *Int J Comput Commun*. 2011;5(1):27-34.
14. Fortier JJ, Solomon H. Clustering procedures. *Multivar Anal*. 1966;62.
15. Fraley C, Raftery AE. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *Comput J*. 1998;41(8):578-588. doi:10.1093/comjnl/41.8.578
16. Quintano C, Castellano R, Rocca A. (233) Claudio QUINTANO, Rosalia CASTELLANO, Antonella ROCCA - L'EVOLUZIONE DEL PART-TIME IN ITALIA NEGLI ANNI DELLA FLESSIBILITÀ DEL LAVORO: LE DIFFERENZE TRA OCCUPAZIONE MASCHILE E FEMMINILE E TRA LE DIVERSE REGIONI DEL PAESE - Quintano C. (a cura di) (2007), Scritti di Statistica Economica 14, pp- 199 - 244 Quaderni di discussione, Dipartimento di Statistica e Matematica per la Ricerca Economica, Università degli Studi di Napoli "Parthenope", n. 30, Napoli. Published online January 1, 2007.
17. Besozzi M. Statistica e grafica con R: Analisi dei gruppi (clustering gerarchico e non gerarchico). Statistica e grafica con R. Published October 3, 2019. Accessed September 15, 2021. https://impararfacendo.blogspot.com/2019/10/analisi-dei-gruppi-metodi-gerarchici-e.html

18. Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. In: *SIGMOD '96*. ; 1996. doi:10.1145/233269.233324
19. Guha S, Rastogi R, Shim K. Rock: A robust clustering algorithm for categorical attributes. *Inf Syst*. 2000;25(5):345-366. doi:10.1016/S0306-4379(00)00022-3
20. Guha S, Rastogi R, Shim K. Cure: an efficient clustering algorithm for large databases. *Inf Syst*. 2001;26(1):35-58. doi:10.1016/S0306-4379(01)00008-4
21. Karypis G, Han E-H, Kumar V. Chameleon: hierarchical clustering using dynamic modeling. *Computer*. 1999;32(8):68-75. doi:10.1109/2.781637
22. Lloyd S. Least squares quantization in PCM. *IEEE Trans Inf Theory*. 1982;28(2):129-137. doi:10.1109/TIT.1982.1056489
23. Diagramma di Voronoi. In: *Wikipedia*. ; 2020. Accessed September 15, 2021. https://it.wikipedia.org/w/index.php?title=Diagramma_di_Voronoi&oldid=111727968
24. Lam D, Wunsch DC. Chapter 20 - Clustering. In: Diniz PSR, Suykens JAK, Chellappa R, Theodoridis S, eds. *Academic Press Library in Signal Processing*. Vol 1. Academic Press Library in Signal Processing: Volume 1. Elsevier; 2014:1115-1149. doi:10.1016/B978-0-12-396502-8.00020-6
25. Bustamam A, Tasman H, Yuniarti N, Frisca, Mursidah I. *Application of K-Means Clustering Algorithm in Grouping the DNA Sequences of Hepatitis B Virus (HBV)*. Vol 1862.; 2017. doi:10.1063/1.4991238
26. Dunn JC. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J Cybern*. 1973;3(3):32-57. doi:10.1080/01969727308546046
27. Bezdek JC. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer US; 1981. doi:10.1007/978-1-4757-0450-1
28. Yager RR, Filev DP. Approximate clustering via the mountain method. *IEEE Trans Syst Man Cybern*. 1994;24(8):1279-1284. doi:10.1109/21.299710
29. Gath I, Geva AB. Unsupervised optimal fuzzy clustering. *IEEE Trans Pattern Anal Mach Intell*. 1989;11(7):773-780. doi:10.1109/34.192473
30. Hathaway RJ, Bezdek JC, Hu Y. Generalized fuzzy c-means clustering strategies using L/sub p/ norm distances. *IEEE Trans Fuzzy Syst*. 2000;8(5):576-582. doi:10.1109/91.873580
31. Krishnapuram R, Keller JM. A possibilistic approach to clustering. *IEEE Trans Fuzzy Syst*. 1993;1(2):98-110. doi:10.1109/91.227387
32. Ghosh S, Dubey SK. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *Int J Adv Comput Sci Appl IJACSA*. 2013;4(4). doi:10.14569/IJACSA.2013.040406
33. Zahn CT. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Trans Comput*. 1971;C-20(1):68-86. doi:10.1109/T-C.1971.223083
34. Urquhart R. Graph theoretical clustering based on limited neighbourhood sets. *Pattern Recognit*. 1982;15(3):173-187. doi:10.1016/0031-3203(82)90069-3
35. Condon A, Karp RM. Algorithms for Graph Partitioning on the Planted Partition Model. In: Hochbaum DS, Jansen K, Rolim JDP, Sinclair A, eds. *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*. Springer Berlin Heidelberg; 1999:221-232.
36. Wang W, Yang J, Muntz R. STING: A Statistical Information Grid Approach to Spatial Data Mining. :10.
37. Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic subspace clustering of high dimensional data for data mining applications. In: *Proceedings of the 1998 ACM SIGMOD*

International Conference on Management of Data. SIGMOD '98. Association for Computing Machinery; 1998:94-105. doi:10.1145/276304.276314
38. Bandaru S, Deb K. Automating Discovery of Innovative Design Principles through Optimization. Published online September 17, 2021.
39. von Luxburg U. A tutorial on spectral clustering. *Stat Comput*. 2007;17(4):395-416. doi:10.1007/s11222-007-9033-z
40. Weisstein EW. Laplacian Matrix. Accessed September 8, 2021. https://mathworld.wolfram.com/LaplacianMatrix.html
41. Weisstein EW. Degree Matrix. Accessed September 8, 2021. https://mathworld.wolfram.com/DegreeMatrix.html
42. Chandra B, Mazumdar S, Arena V, Parimi N. *Elegant Decision Tree Algorithm for Classification in Data Mining.*; 2002:169. doi:10.1109/WISEW.2002.1177859
43. Peila D. INDAGINI PRELIMINARI NELLA COSTRUZIONE DI GALLERIE: ANALISI DELLA LETTERATURA TECNICA PRELIMINARY INVESTIGATION FOR TUNNELLING: TECHNICAL REVIEW. *Geoingeneria Ambient E Mineraria*. 2009;47:47-58.
44. Fisher DH. Knowledge acquisition via incremental conceptual clustering. *Mach Learn*. 1987;2(2):139-172. doi:10.1007/BF00114265
45. T. Kohonen. The self-organizing map. *Proc IEEE*. 1990;78(9):1464-1480. doi:10.1109/5.58325
46. Haykin S, Haykin SS, HAYKIN SA. *Neural Networks: A Comprehensive Foundation*. Prentice Hall; 1999.
47. Xu R, Wunsch D. Survey of Clustering Algorithms. *Neural Netw IEEE Trans On*. 2005;16:645-678. doi:10.1109/TNN.2005.845141
48. Forestier G, Gançarski P, Wemmert C. Collaborative clustering with background knowledge. *Data Knowl Eng*. 2010;69(2):211-228. doi:10.1016/j.datak.2009.10.004
49. J. Handl, J. Knowles. An Evolutionary Approach to Multiobjective Clustering. *IEEE Trans Evol Comput*. 2007;11(1):56-76. doi:10.1109/TEVC.2006.877146
50. Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: A tutorial. *Spec Issue - Genet Algorithms Reliab*. 2006;91(9):992-1007. doi:10.1016/j.ress.2005.11.018
51. Faceli K, de Souto MCP, de Araújo DSA, de Carvalho ACPLF. Multi-objective clustering ensemble for gene expression data analysis. *Neurocomputing*. 2009;72(13-15):2763-2774. doi:10.1016/j.neucom.2008.09.025
52. Xu R, Wunsch DC. Clustering algorithms in biomedical research: a review. *IEEE Rev Biomed Eng*. 2010;3:120-154. doi:10.1109/RBME.2010.2083647
53. Stutz J, Cheeseman P. Autoclass — A Bayesian Approach to Classification. In: Skilling J, Sibisi S, eds. *Maximum Entropy and Bayesian Methods*. Springer Netherlands; 1996:117-126.
54. Jörg S, Martin E, Hans-Peter K, Xiaowei X. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Min Knowl Discov*. 1998;2(2):169-194. doi:10.1023/A:1009745219419
55. Ng S, Krishnan T, Mclachlan G. The EM algorithm. *Handb Comput Stat Concepts Methods*. Published online January 1, 2004. doi:10.1007/978-3-642-21551-3_6
56. Tabu search. In: *Wikipedia*. ; 2021. Accessed September 9, 2021. https://en.wikipedia.org/w/index.php?title=Tabu_search&oldid=1022267754

57. Al-Sultan KS. A Tabu search approach to the clustering problem. *Pattern Recognit*. 1995;28(9):1443-1451. doi:10.1016/0031-3203(95)00022-R
58. Glover F. Future paths for integer programming and links to artificial intelligence. *Comput Oper Res*. 1986;13(5):533-549. doi:10.1016/0305-0548(86)90048-1
59. Khanmohammadi S, Adibeig N, Shanehbandy S. An improved overlapping k-means clustering method for medical applications. *Expert Syst Appl*. 2017;67:12-18. doi:10.1016/j.eswa.2016.09.025
60. Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*. 2005;435:814-818.
61. Ahn Y-Y, Bagrow JP, Lehmann S. Link communities reveal multiscale complexity in networks. *Nature*. 2010;466(7307):761-764. doi:10.1038/nature09182
62. Simplified Generation of Biomedical 3D Surface Model Data for Embedding into 3D Portable Document Format (PDF) Files for Publication and Education. *PLOS ONE*. 2013;8(11):e79004. doi:10.1371/journal.pone.0079004
63. D. H. Wolpert, W. G. Macready. No free lunch theorems for optimization. *IEEE Trans Evol Comput*. 1997;1(1):67-82. doi:10.1109/4235.585893
64. Bensmail H, Celeux G, Raftery AE, Robert CP. Inference in model-based cluster analysis. *Stat Comput*. 1997;7(1):1-10. doi:10.1023/A:1018510926151
65. C. Fraley, A. E. Raftery. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *Comput J*. 1998;41(8):578-588. doi:10.1093/comjnl/41.8.578
66. Saxena A, Pal NR, Vora M. Evolutionary Methods for Unsupervised Feature Selection Using Sammon's Stress Function. *Fuzzy Inf Eng*. 2010;2(3):229-247. doi:10.1007/s12543-010-0047-4