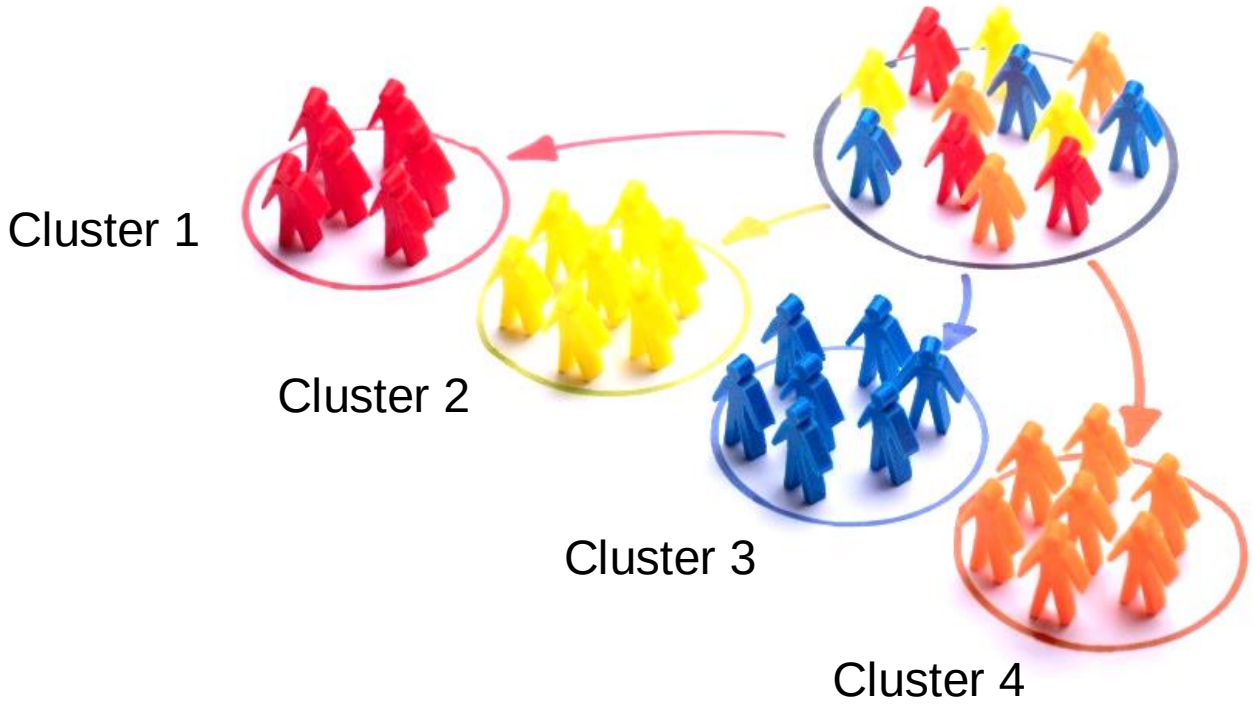
Cluster Analysis



L'analisi dei cluster (cluster analysis) è una tecnica statistica multivariata il cui obiettivo è raggruppare oggetti in base a un insieme di caratteristiche selezionate dall'utente.

L'analisi dei cluster consente di dividere un insieme di osservazioni in cluster (sottoinsiemi) in modo tale che:

➢ le osservazioni simili (o correlate) tra loro si trovano nello stesso gruppo
➢ le osservazioni dissimili (o non correlate) rispetto agli oggetti si trovano in altri gruppi

APPLICAZIONI:

Biologia: per raggruppare geni o proteine in base alla loro funzionalità o alla loro espressione. Questa segmentazione può facilitare la comprensione delle basi molecolari delle malattie e accelera lo sviluppo di terapie personalizzate, aprendo nuove vie nella lotta contro patologie complesse

Recupero di informazioni: clustering di documenti

Uso del suolo: identificazione di aree con uso del suolo simile in un database di osservazione della Terra

Marketing: aiuta a scoprire gruppi distinti all'interno di una database clienti e a utilizzare queste informazioni per sviluppare programmi di marketing mirati

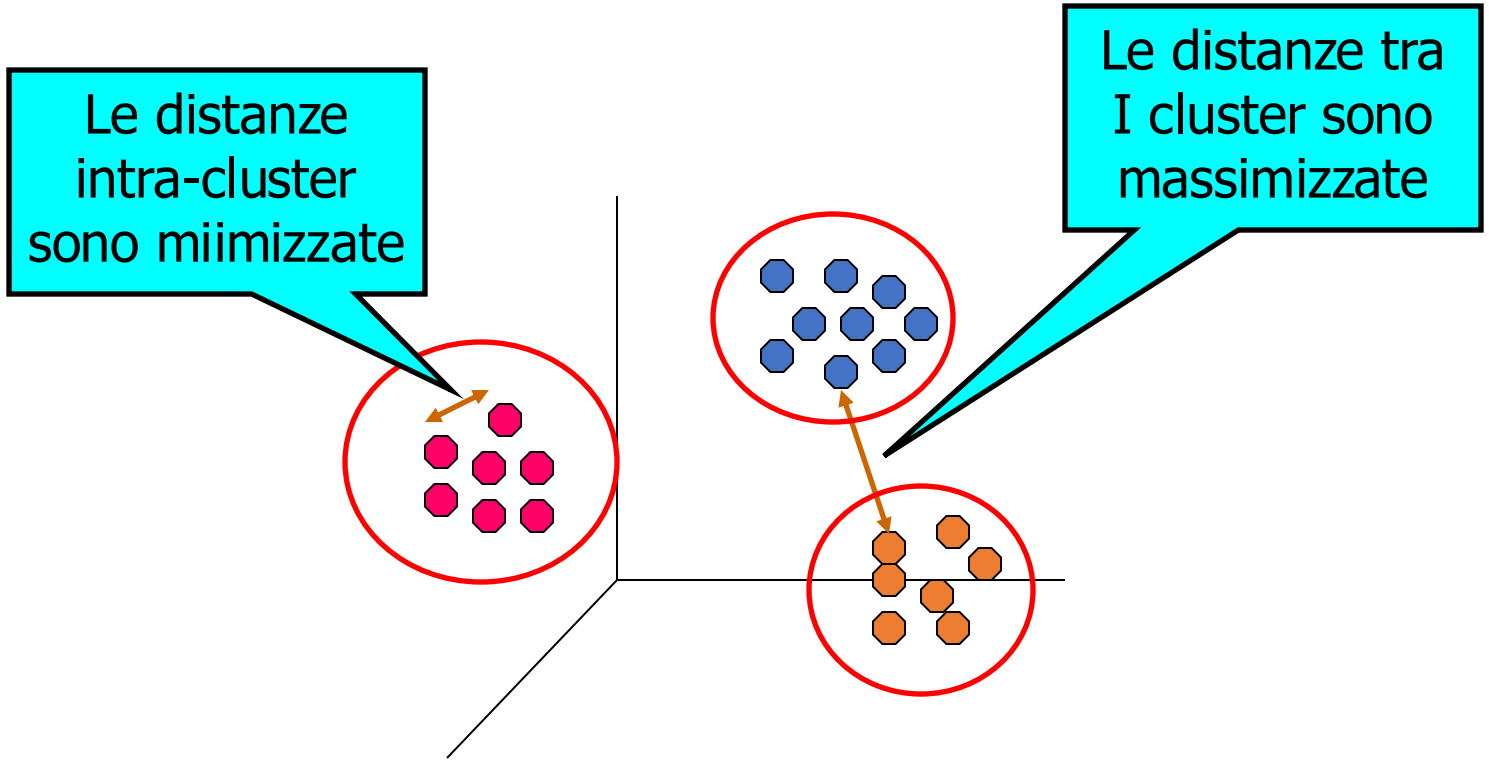
Pianificazione urbana: identificazione di gruppi di case in base al tipo, al valore e alla posizione geografica

Visione artificiale: gioca un ruolo chiave nella segmentazione di immagini, permettendo di distinguere e classificare componenti o oggetti significativi all’interno di un’immagine.

Come Funziona la Cluster Analysis?

Passaggi Principali:

- Selezione delle Caratteristiche: Si scelgono le variabili o caratteristiche da analizzare, ad esempio età + glicemia + BMI + ….
- Calcolo della Similarità: Si misura la similarità tra le osservazioni usando metriche come la distanza euclidea o la correlazione.
- Algoritmo di Clustering: Si applica un algoritmo per dividere i dati in cluster.


Le distanze
intra-cluster
sono miimizzate
Le distanze tra
I cluster sono
massimizzate

La Cluster analysis è un metodo non supervisionato: non ci sono classi predefinite:

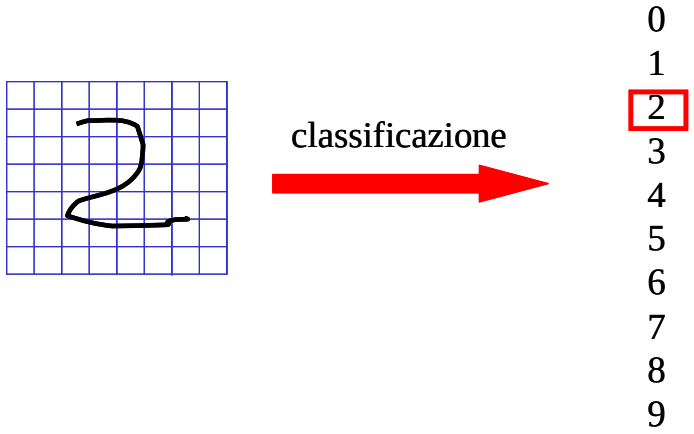
Classificazione supervisionata:

richiede campioni di riferimento precedentemente classificati (a priori) per addestrare il classificatore e successivamente classificare i dati sconosciuti

Esempio:

Pattern: numeri scritti a mano

10 classi (definite a priori): 0,1,2,3,4,5,6,7,8,9



Classificazione NON supervisionata:

NON richiede campioni di riferimento precedentemente classificati (a priori)

METODI DI CLUSTERING

- **Kmeans**: è un metodo che raggruppa i punti dati in un numero predefinito (k) di cluster basato sulla loro distanza dal centroide di ciascun cluster. Si tratta di un algoritmo iterative.
- **Clustering Gerarchico**: è un metodo che crea una gerarchia di cluster suddividendoli o unendoli ricorsivamente in base alle loro somiglianze.
- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise): è Un metodo che raggruppa i punti dati in base alla loro densità entro un raggio o una soglia di distanza definite
- **Spectral Clustering**
- …

Kmeans

Supponiamo di avere un datset composto da N misure (o osservazioni) e definito da questo vettore colonna (T: simbolo di trasporto):

$$[x_1, x_2, .., x_N\ ]^T$$

Scopo: dividere le N misure in K clusters (gruppi) con N> K

K : Noto e definito dall'operatore

Si definisce il vettore:

$$[\mu_1, \mu_2, .., \mu_K\ ]^T$$

$$\mu_i = centro\ del\ cluster\ i-esimo, o\ centroide\ o\ seed$$

Kmeans

Soluzione: assegniamo la i-esima misura x_i al cluster j-esimo il cui centroide ha distanza minima da x_i : $\min d(x_i, \mu_j)$

DISTANZA: In K-means, esistono diverse metriche di distanza che possono essere utilizzate per calcolare la distanza tra i punti dati e i centroidi dei cluster. Tuttavia, il K-means classico utilizza principalmente la distanza euclidea.

1. Distanza Euclidea: è la misura di distanza standard in K-means, calcolata come la radice quadrata della somma delle differenze al quadrato tra le coordinate. Viene utilizzata per cluster con distribuzione sferica e funziona bene con variabili quantitative.

$$d(x, \mu) = \sqrt{\sum_{i=1}^{K} (x_i - \mu_k)^2}$$

2. **Distanza di Manhattan** (o Distanza di Cityblock, o Taxicab geometry):
Calcolata come la somma del valore assoluto delle differenze tra le coordinate. Può essere usata in varianti di K-means e si adatta bene ai dati distribuiti in modo lineare o disposti su una griglia.

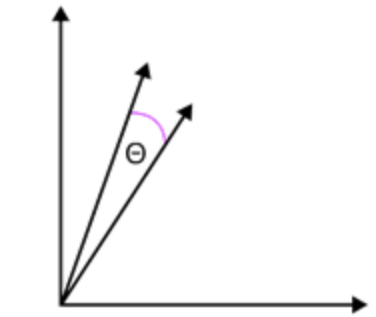
$$d(x, \mu) = \sum_{i=1}^{K} |x_i - \mu_k|$$

3. **Distanza di Minkowski**: è una generalizzazione delle distanze euclidea e di Manhattan

$$d(x, \mu) = \sqrt[p]{\sum_{i=1}^{K} (x_i - \mu_k)^p}$$

4. **Distanza Coseno**: misura l'angolo tra due vettori (es **A** e **B**) e risulta utile per dati di tipo vettoriale o documenti di testo. Non considera la magnitudine ma solo l'orientamento.

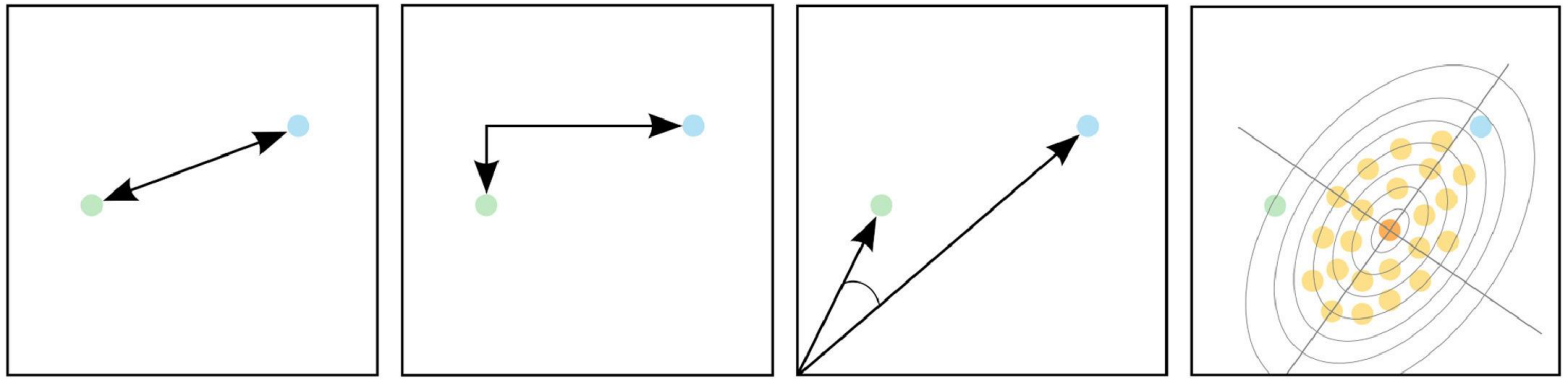
$$cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



5. **Distanza di Hamming**:
Calcola il numero di differenze tra le coordinate e viene usata per dati binari o categorici, specialmente per varianti di clustering su dati non numerici.

6. **Distanza Mahalanobis**:
Tiene conto della correlazione tra le variabili e viene usata in casi in cui le variabili sono correlate tra loro, ma è meno comune nel K-means standard.

…



(a) Euclidean (b) Manhattan (c) Cosine (d) Mahalanobis

ALGORITMO: il K-means classico utilizza principalmente la distanza euclidea. Quindi vediamo come procede il metodo usando questa distanza.

Si procede minimizzando la seguente funzjone obiettivo:

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\left\|x_n - \mu_k\right\|^2$$

Dove:

$$r_{nk} \in \{0,1\} \quad k = 1,\ldots,K$$

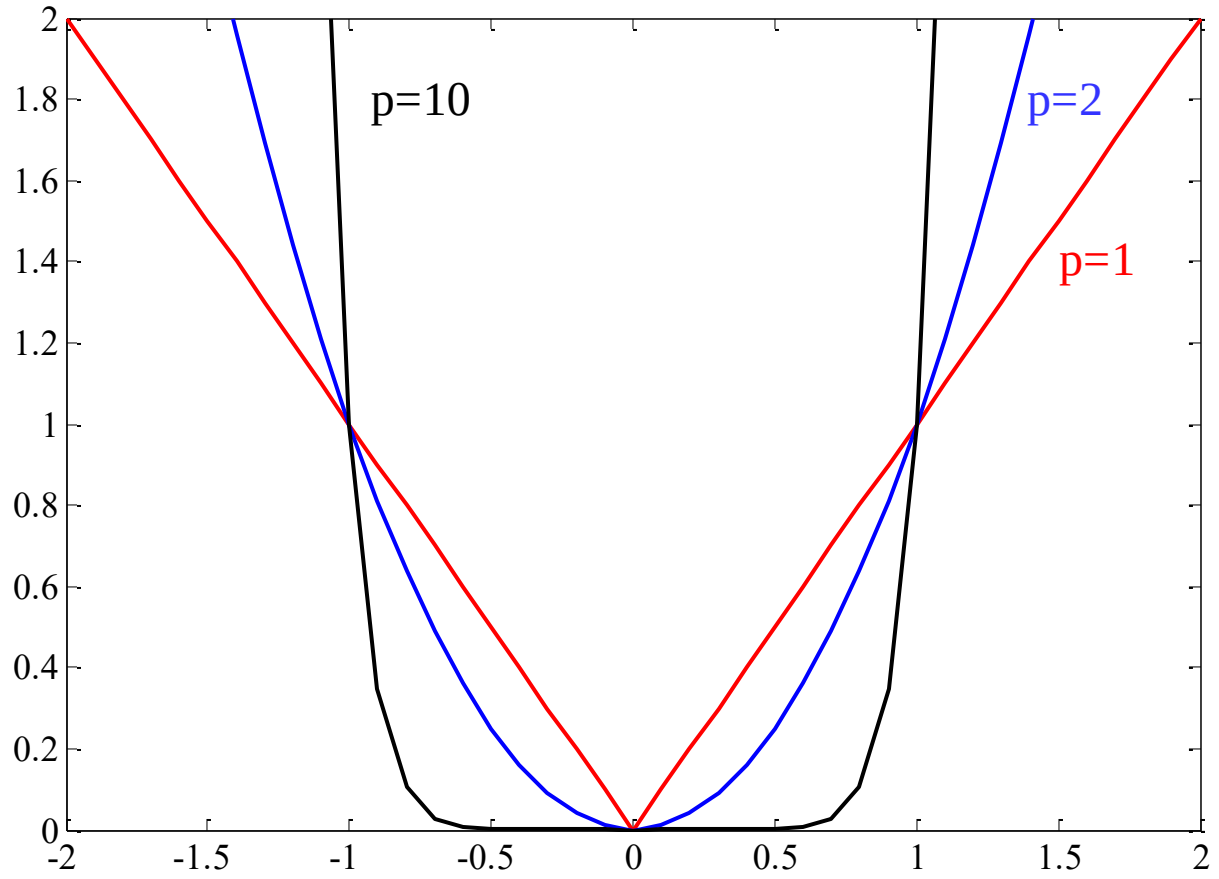
Se il dato x_n è assegnato al cluster k-esimo:

$$r_{nk} = 1$$
$$r_{nj} = 0 \quad j \neq k$$

Più precisamente vogliamo trovare i valori di r_{nk} e μ_k che minimizzano la funzione obiettivo J

Maggiore è p più la funzione obiettivo riceve informazioni dai punti/dati che sono più distanti dal centroide.

$|x_i - \mu_k|^p$



$x_i - \mu_k$

$$d(x, \mu) = \sqrt[p]{\sum_{i=1}^{K} (x_i - \mu_k)^p}$$

ALGORITMO:

Primo step: i valori iniziali di μ_k sono calcolati dividendo le N misure/valori in K sotto-insiemi non nulli e calcolando le media dei valori per ogni K sotto-insieme

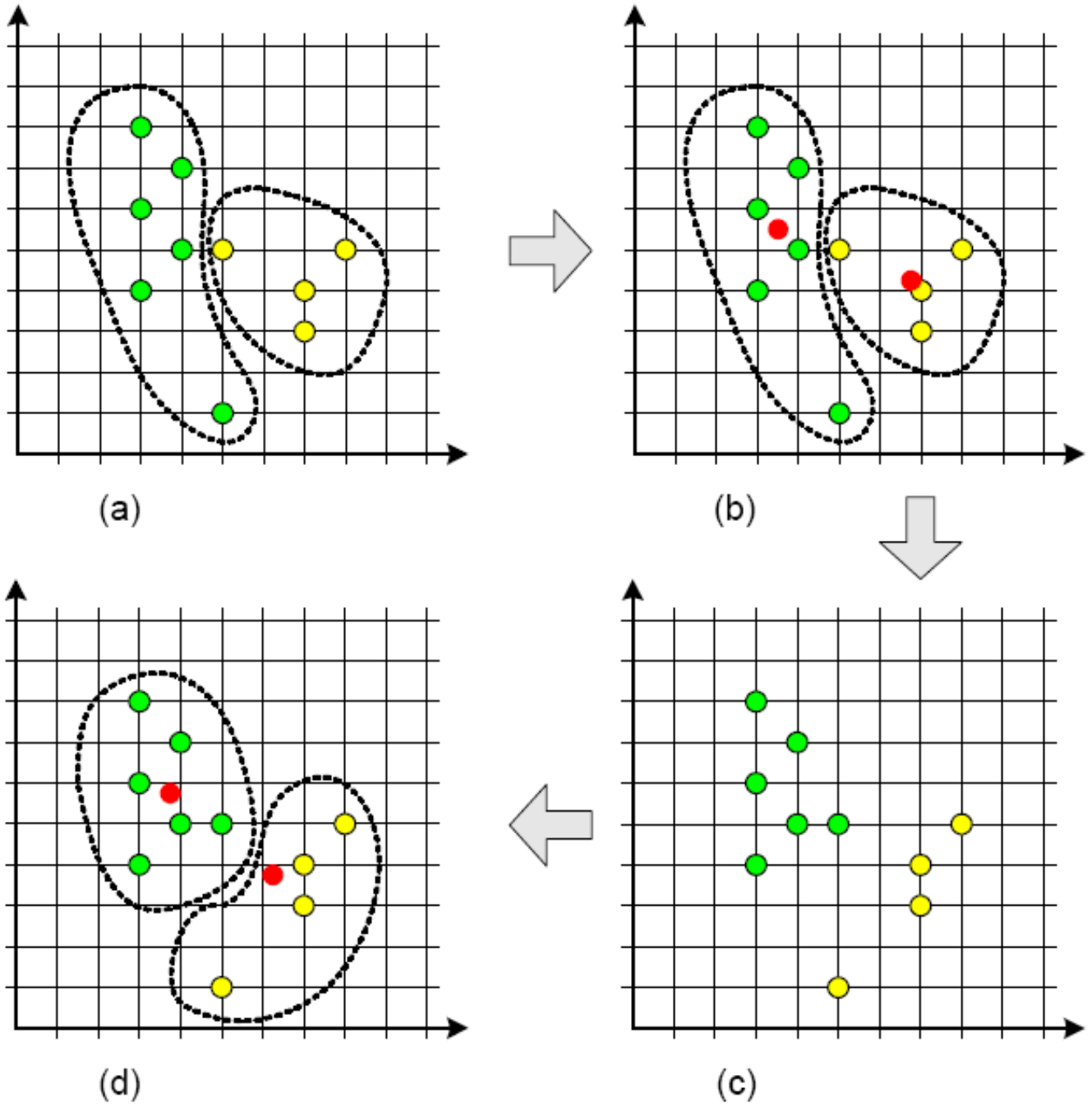
iterazione

Ciascun valore x è assegnato al cluster il cui centroide è il più vicino (il meno distante) da x, ossia r_{nk} sono stimati minimizzando J e mantenendo fissi i centroidi μ_k

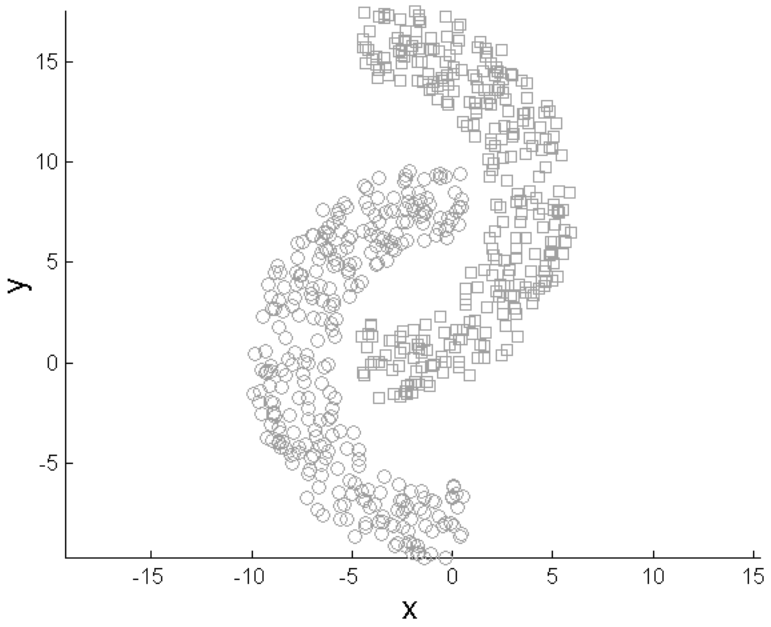
Calcola nuovamente i centroidi dei cluster della suddivisione corrente μ_k, i.e. μ_k sono stimati minimizzando J e mantenendo fisse le stime r_{nk}

Output: termina quando l'assegnazione non cambia o si raggiunge il numero massimo di iterazioni.

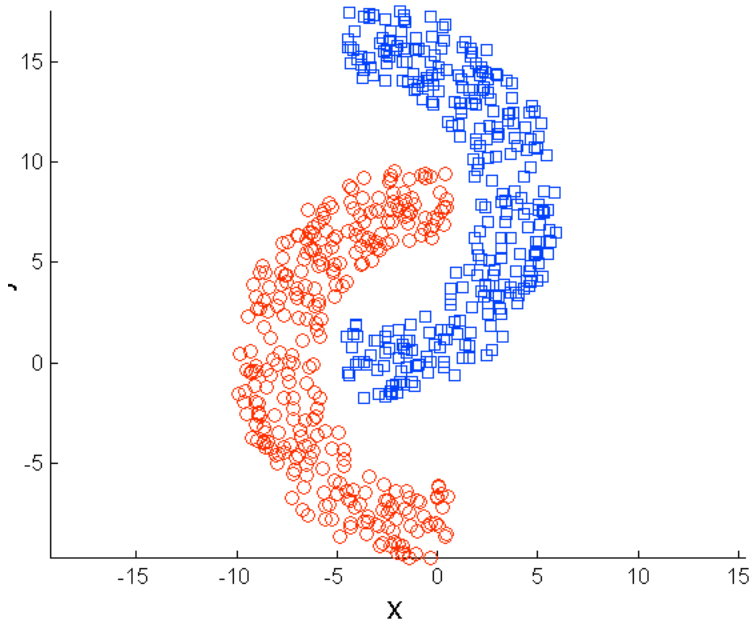
ESEMPIO:



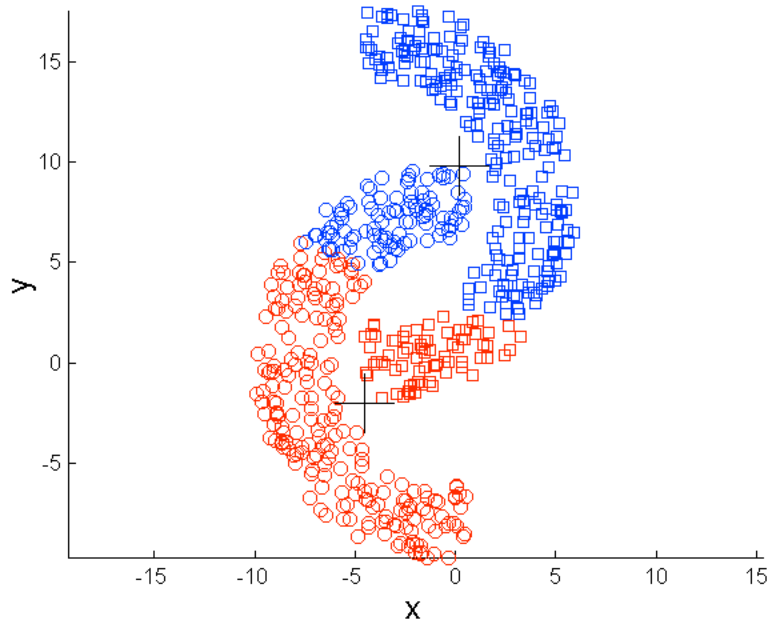
Limiti del K-means: forme non-globulari Shapes



Valori originali



Possibile suddivisione in 2 cluster



K-means (2 Clusters)

VALUTARE L'AFFIDABILITA' E LA VALIDITA'

- Effettua un'analisi sugli stessi dati utilizzando diverse misure di distanza. Confronta i risultati tra le diverse misure per determinare la stabilità delle soluzioni.

- Dividi i dati casualmente in due metà. Esegui il clustering separatamente su ciascuna metà. Confronta i centroidi dei cluster tra i due sotto-campioni.

- Elimina variabili casualmente. Esegui il clustering basato sul set ridotto di variabili. Confronta i risultati con quelli ottenuti utilizzando l'intero set di variabili.

- **La soluzione può dipendere dall'inizializzazione dell'algoritmo. Esegui più iterazioni utilizzando suddivisioni diverse dei dati fino a quando la soluzione si stabilizza**.

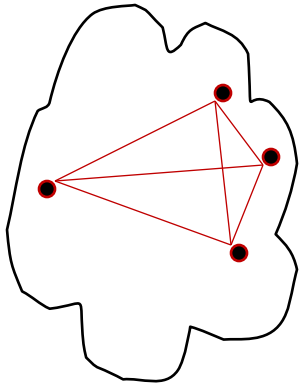
COESIONE E SEPARAZIONE DEI CLUSTER

Coesione del cluster:

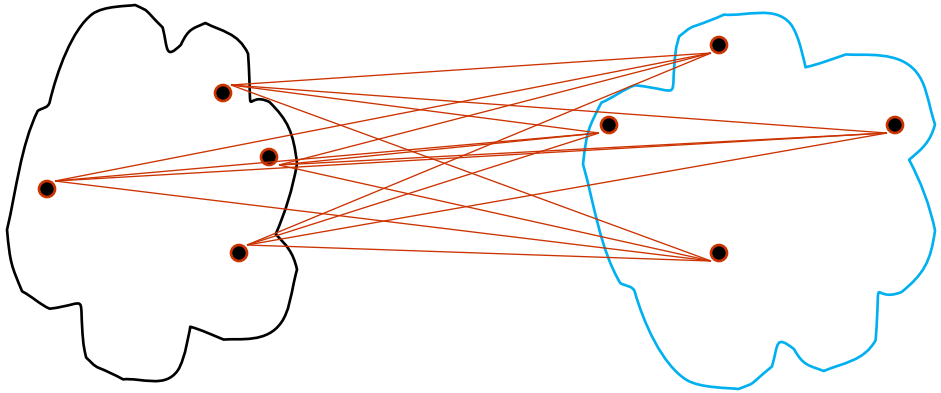
misura quanto sono strettamente vicini gli oggetti in un cluster
La coesione del cluster è la somma del peso di tutti i collegamenti all'interno di un cluster

Separazione dei cluster:

misura quanto sono ben separati/distinti i cluster tra di loro.
La separazione del cluster è la somma delle distanze tra gli elementi nel cluster i-esimo e gli elementi degli altri cluster



coesione



separazione

INDICE DI SILHOUETTE

L'indice di silhouette misura quanto bene un punto è assegnato al proprio cluster rispetto a un altro cluster. Varia tra -1 e +1:

- ✓ Un valore vicino a +1 indica che il punto è ben assegnato al proprio cluster.
- ✓ Un valore vicino a 0 indica che il punto è ai confini tra due cluster.
- ✓ Un valore vicino a -1 indica che il punto sarebbe più adatto in un altro cluster (*patologia*).

La **silhouette media** di tutti i punti dà una **misura generale** della bontà del clustering

Il coefficiente di silhouette combina idee di coesione e separazione

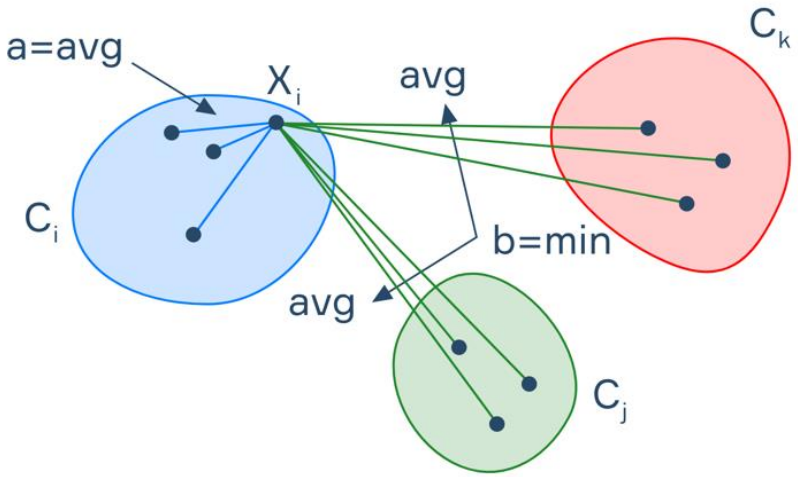
INDICE DI SILHOUETTE

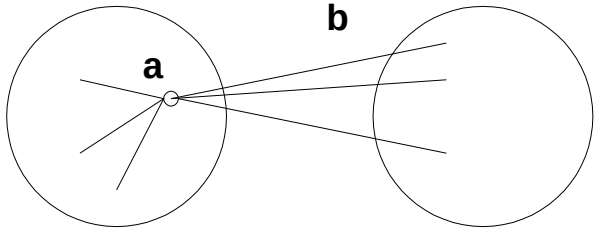
Consideriamo il dato x_i:

a = distanza media di x_i con gli altri dati del suo cluster

b = min (per ogni cluster j a cui x_i NON appartiene, calcolare la distanza media di x_i con i dati del cluster j)

Il coefficiente di Silhuette S per x_i è dato allora da: $S = (b - a)/\max(a, b)$





Avendo S per ogni dato, si può anche calcolare il valore medio di tutti gli S

CONTRO

L’algoritmo può fermarsi ad un minimo locale

Il numero di cluster deve essere fissato a priori

E’ sensibile agli outliers. La presenza di outliers può distorcere la distribuzione dei dati

PRO

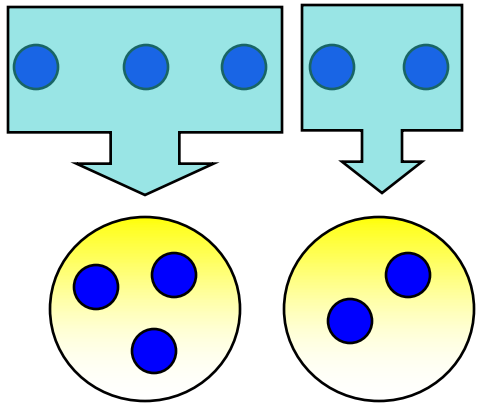
Semplice e veloce

Flessibile: i cluster sono aggiornati ad ogni step

K-means effettua un'assegnazione deterministica dei dati (assegnazione rigida o esclusiva): ogni dato x_i è assegnato a uno e un solo cluster

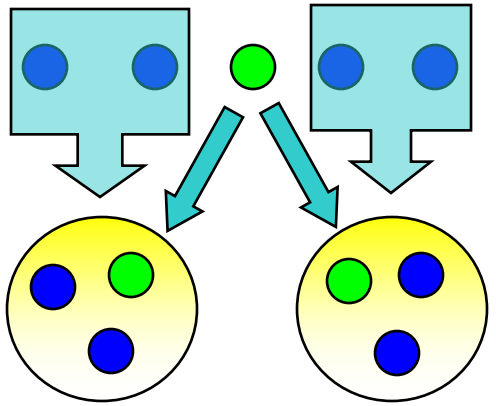
Assegnazione deterministica (hard)

– Ciascun dato è assegnato a solo un cluster



Assegnazione probabilistica (soft)

– Ciascun dato è assegnato a più cluster



Metodi di partizione soft: fuzzy C-means

Dato:

$x_1, x_2, \ldots, x_N$ dati, misure

K fissato, N>K

$\mu_1, \mu_2, \ldots, \mu_K$ centroidi

distanza euclidea

La funzione obiettivo ora è:

$$J_m = \sum_{n=1}^{N} \sum_{k=1}^{K} p_{nk}^{m} \left\| x_n - \mu_k \right\|^2 \quad 1 < m < \infty$$

m è un numero reale (solitamente è un intero >1 ma <5) chiamato ‘fuzzy index’

dove:

p_{nk} = probabilità che il dato x_n appartenga al cluster k

$$p_{nk} \in [0,1] \quad k = 1,\ldots,K$$

$$\sum_{k=1}^{K} p_{nk} = 1$$

Esempio con K=2

$$\mathbf{p} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ & \\ p_{N1} & p_{N2} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0 & 1 \\ & \\ 0.9 & 0.1 \end{bmatrix}$$

ALGORITMO: STEP iniziale

Scelta di m e dei valori iniziali p_{nk}

ad esempio: suddividere casualmente N misure in K cluster (circa N/K misure per ogni cluster). La misura generale x_n, assegnata al cluster k, ha probabilità:

$$p_{nk} = 1$$
$$p_{nj} = 0 \quad j = 1,2,\dots,K \quad j \neq k$$

STEP 2: si calcola il centroide di ciascun cluster come media ponderata

$$\mu_k = \frac{\sum_{i=1}^{N} p_{ik}^m \cdot x_i}{\sum_{i=1}^{N} p_{ik}^m}$$

STEP 3: Aggiornare la probabilità basandosi sull'idea che minore è la distanza tra misura e cluster, maggiore è la probabilità corrispondente

$$p_{nk} = \frac{1}{\sum_{j=1}^{K} \left(\frac{\|x_n - \mu_k\|}{\|x_n - \mu_j\|} \right)^{\frac{2}{m-1}}}$$

STEP 4: ripetere dallo step 2 fino a quando le variazioni dei valori di probabilità diventano minori di un valore selezionato oppure fino a quando non si supera un numero massimo di iterazioni

STOP

Si può decidere di assegnare deterministicamente una misura al cluster per il quale la probabilità di appartenenza è più alta oppure si può usare un criterio di soglia.

ad esempio associare x_n ai cluster per i quali x_n ha una probabilità>80%, altrimenti non tenere conto di x_n.

Clustering Gerarchico

Il Clustering Gerarchico Agglomerativo è una tecnica di clustering utilizzata per organizzare i dati in una struttura ad albero, chiamata dendrogramma, che rappresenta le relazioni di similarità tra i dati.

Caratteristiche principali

- Approccio Bottom-Up: Inizia con ogni punto dati come un singolo cluster e li unisce gradualmente in cluster più grandi.

- Distanza e Similarità: La fusione tra cluster è basata su una misura di distanza, come la distanza Euclidea o Manhattan, o su una misura di similarità.

- Dendrogramma: La struttura gerarchica creata aiuta a visualizzare i cluster e a scegliere il numero ottimale di gruppi.

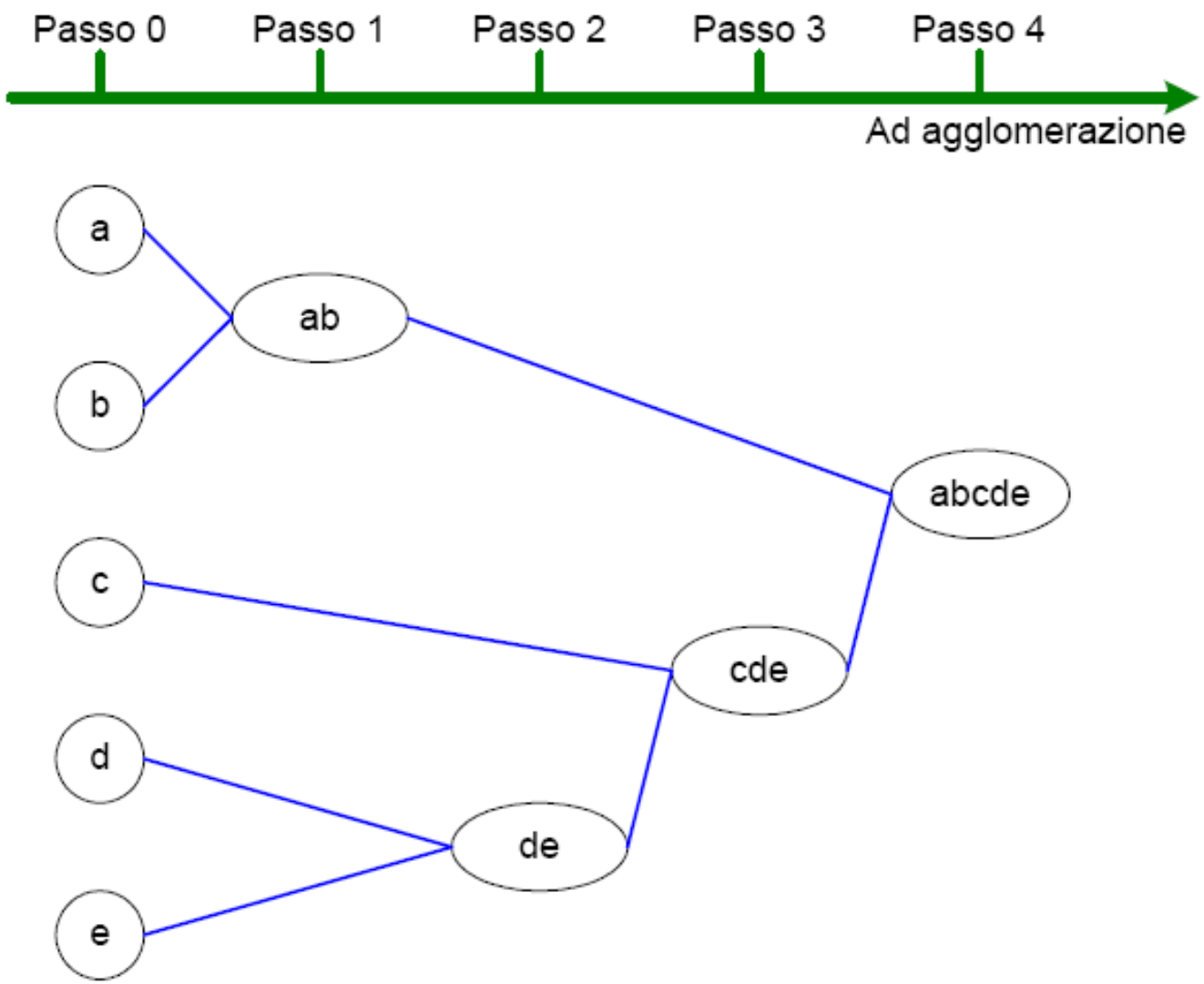
Clustering Gerarchico

Vantaggi e Svantaggi

Vantaggi: Non richiede di specificare il numero di cluster in anticipo e genera una rappresentazione gerarchica dei dati.

Svantaggi: Richiede molto tempo di calcolo per dataset grandi e sensibile alla scelta della metrica di distanza.

Clustering Gerarchico



STEP 1

Ogni misura è un cluster.

Per ogni coppia di misure x_i, x_j (N(N-1)/2 = numero di coppie con N numero totale di misure) calcolare la loro distanza. Spesso si usa la distanza euclidea.

Si forma il cluster sulla base della misura di distanza e del criterio di linkage scelto ➔

Criterio di Linkage

Single Linkage (Collegamento Singolo): calcola la distanza minima tra i punti di due cluster. La distanza tra due cluster è data dalla distanza più breve tra i singoli punti in ciascun cluster. È sensibile al rumore e tende a formare cluster "a catena".

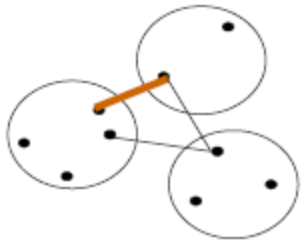
Complete Linkage (Collegamento Completo): calcola la distanza massima tra i punti di due cluster. La distanza tra due cluster è quindi data dalla distanza maggiore tra i punti nei due cluster. Questo metodo tende a creare cluster più compatti e di forma simile.

Average Linkage (Collegamento Medio): misura la distanza media tra tutti i punti di un cluster e tutti i punti di un altro cluster. Spesso è una scelta bilanciata per ottenere cluster di forma uniforme.

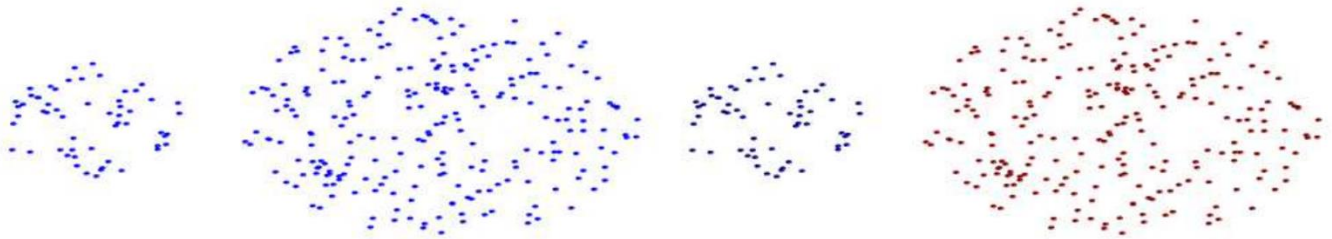
Centroid Linkage (Collegamento del Centroid): calcola la distanza tra i centroidi (o medie) di ciascun cluster. È sensibile alle variazioni nei centroidi e funziona bene se i cluster hanno distribuzioni simmetriche.

Ward’s Linkage: cerca di minimizzare la varianza interna di ciascun cluster. Questo metodo tende a produrre cluster di dimensioni simili ed è molto usato perché cerca di ridurre la dispersione interna tra i dati, creando cluster più omogenei.

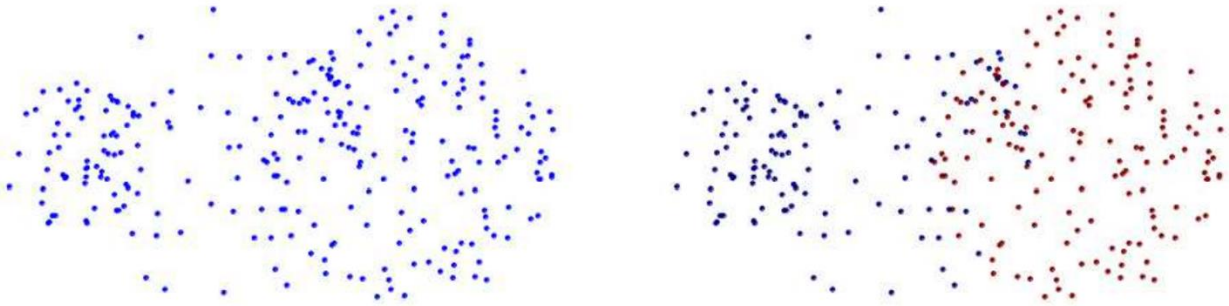
– **Single linkage** (nearest neighbour): the distance between two clusters is the smallest distance between two objects in the considered clusters;



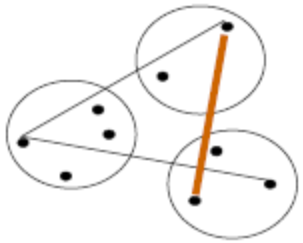
- Permette di gestire anche cluster non sferici



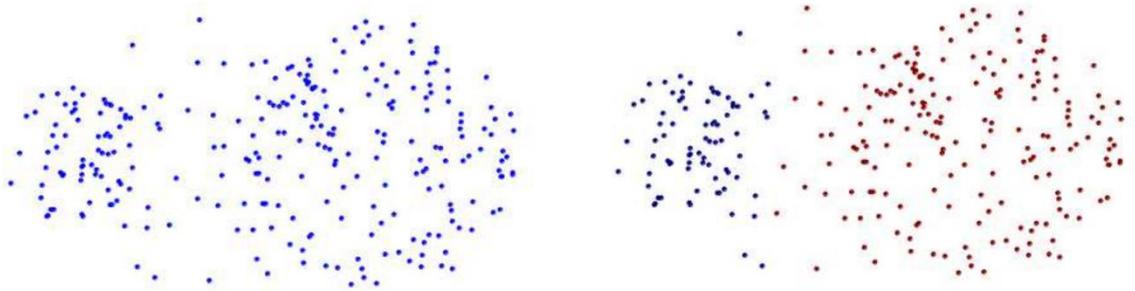
- E' soggetto a outlier e rumori



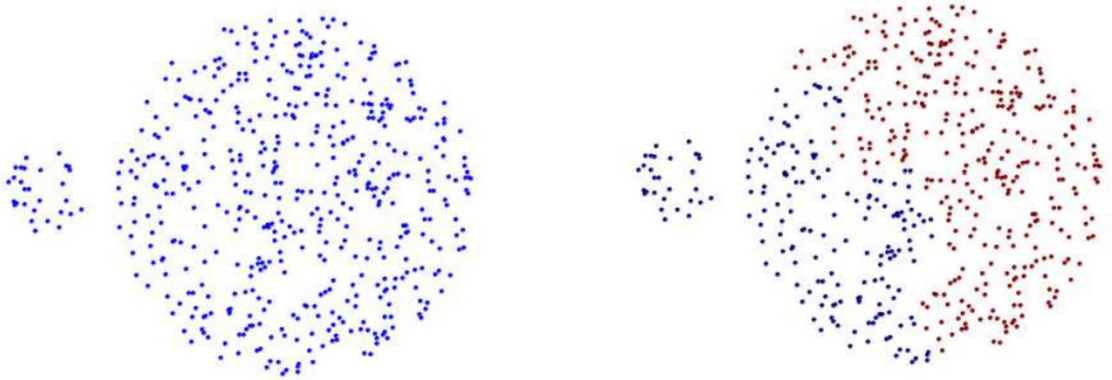
– **Complete linkage** (furthest neighbour): the distance between two clusters is the largest distance between two objects in the considered clusters;



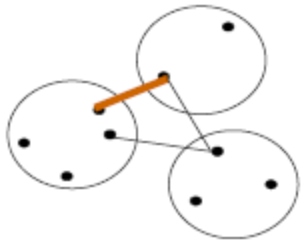
- Meno suscettibile al rumore



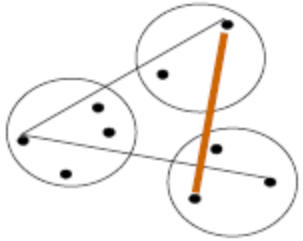
- Tende a separare cluster grandi
- Privilegia cluster globulari



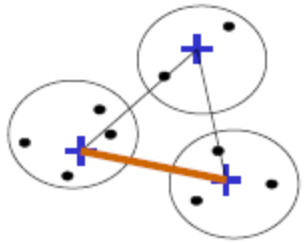
– **Single linkage** (vicino più prossimo): la distanza tra due cluster è la distanza minima tra due oggetti nei cluster considerati;



– **Complete linkage** (vicino più lontano): la distanza tra due cluster è la distanza massima tra due oggetti nei cluster considerati;



– **Centroid linkage:** La distanza tra due cluster è la distanza tra i loro centroidi

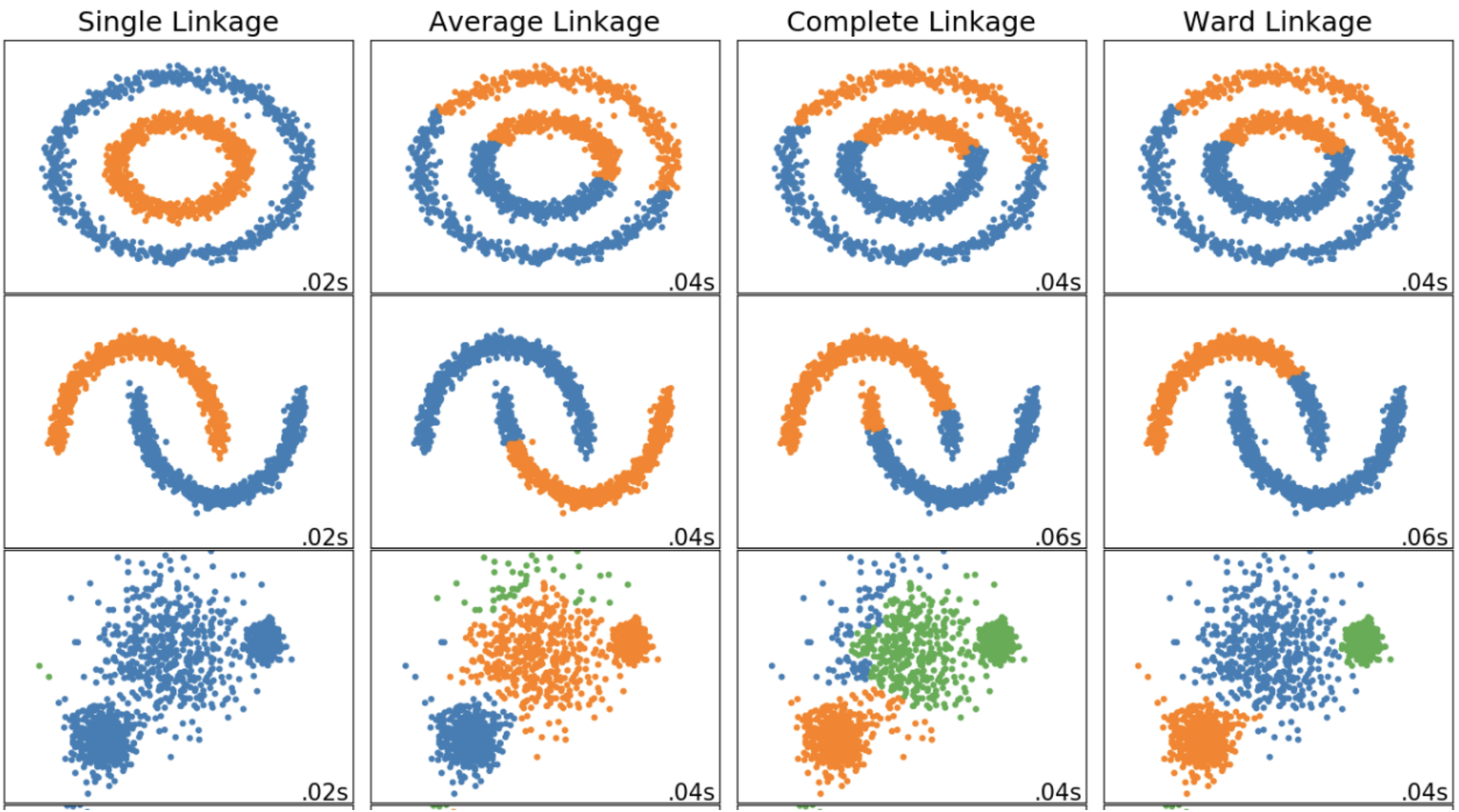


Ward's Linkage: cerca di minimizzare la varianza interna di ciascun cluster. Questo metodo tende a produrre cluster di dimensioni simili ed è molto usato perché cerca di ridurre la dispersione interna tra i dati, creando cluster più omogenei.

Dati due cluster A e B la formula usata per calcolare la distanza tra A e B è:

$$d(A,B) = \frac{N_A N_B}{N_A + N_B} \|C_A - C_B\|^2 = \Delta(A,B)$$

Dove C_A e C_B sono i centroidi . Questa distanza è pari a quanto la varianza aumenterà nel nuovo cluster dato dalla combinazione delle misure comprese nel cluster A e B.; è anche detta mergin cost dovuto alla combinazione del cluster A con B


Single Linkage
Average Linkage
Complete Linkage
Ward Linkage
.02s
.04s
.04s
.04s
.02s
.04s
.06s
.06s
.02s
.04s
.04s
.04s

STEP 3

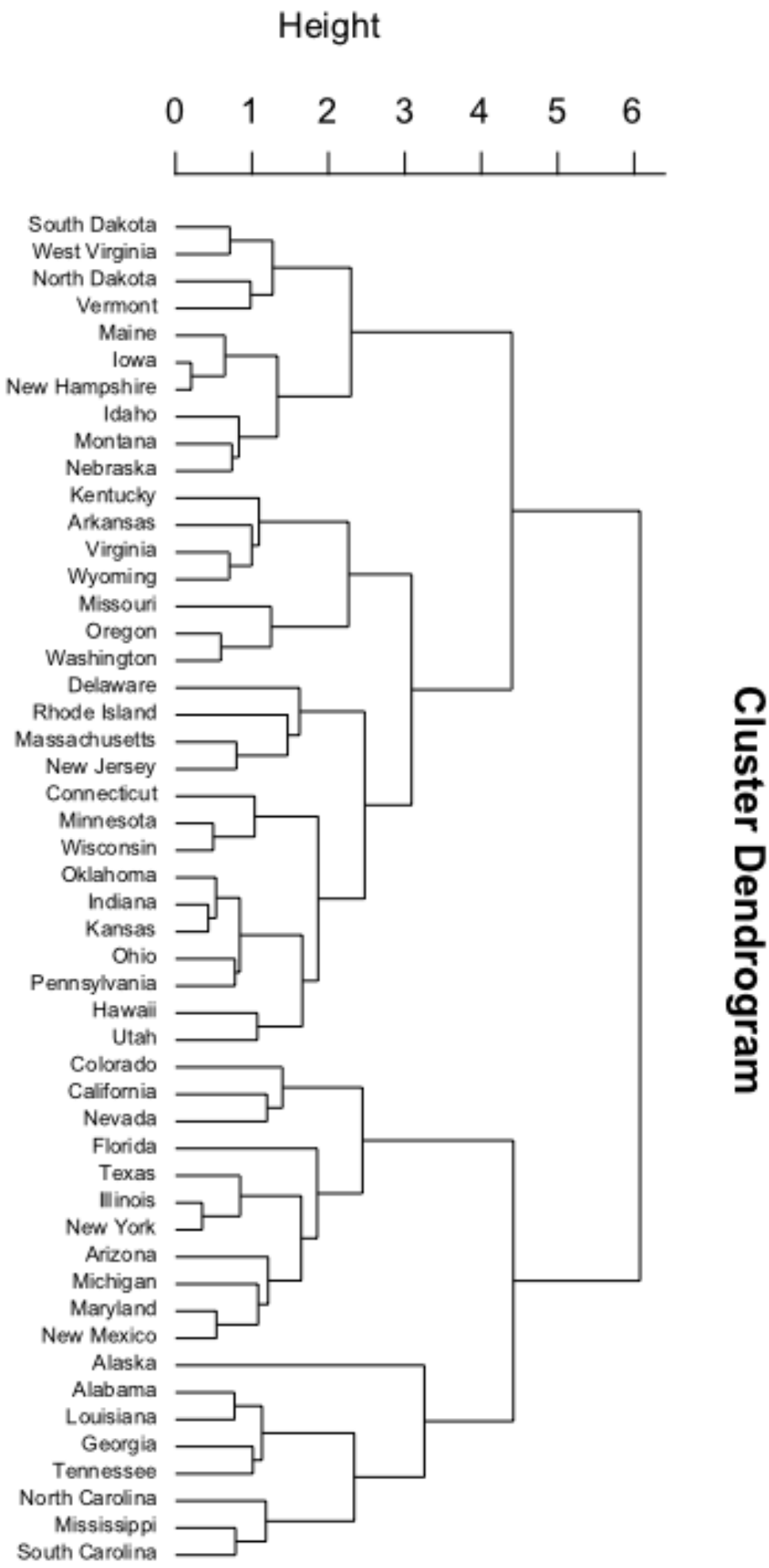
Unisci i due cluster più vicini: questi cluster vengono raggruppati e il numero di cluster diminuisce di un'unità (da K=N a K=N-1

STEP 4

Aggiorna le distanze tra i cluster K=N-1 secondo il criterio di collegamento scelto..

STEP 5

Ripeti gli step 2, 3 e 4 fino ad ottenere un solo cluster


Height
0
1
2
3
4
5
6
South Dakota
West Virginia
North Dakota
Vermont
Maine
Iowa
New Hampshire
Idaho
Montana
Nebraska
Kentucky
Arkansas
Virginia
Wyoming
Missouri
Oregon
Washington
Delaware
Rhode Island
Massachusetts
New Jersey
Connecticut
Minnesota
Wisconsin
Oklahoma
Indiana
Kansas
Ohio
Pennsylvania
Hawaii
Utah
Colorado
California
Nevada
Florida
Texas
Illinois
New York
Arizona
Michigan
Maryland
New Mexico
Alaska
Alabama
Louisiana
Georgia
Tennessee
North Carolina
Mississippi
South Carolina
Cluster Dendrogram

Per **tagliare un albero generato** dal clustering gerarchico agglomerativo e ottenere un determinato numero di cluster, esistono diversi metodi:

1. Soglia di Distanza (o Dissimilarità) Si seleziona una soglia di distanza e si taglia l'albero in modo da interrompere l'unione dei cluster quando la distanza tra essi supera quella soglia. I nodi al di sotto di questa distanza formano i cluster finali. Questo metodo è utile quando si ha una buona idea della distanza massima accettabile tra i punti all'interno di un cluster.

2. Numero Fisso di Cluster. Si decide a priori quanti cluster si desiderano e si taglia l'albero a un livello tale da ottenere esattamente quel numero di cluster. Ad esempio, se si vogliono 3 cluster, si taglia l'albero in modo da interrompere la fusione al punto in cui si generano esattamente 3 gruppi. Questo approccio è semplice, ma può non essere sempre ottimale se la struttura dell'albero non supporta quel numero specifico di cluster in modo naturale.

3. Distanza Massima tra Cluster (Metodo di Ward). Utilizzato in combinazione con il criterio di Ward, taglia l'albero in modo da minimizzare l'aumento della varianza intra-cluster risultante da ogni fusione. Si sceglie di interrompere la fusione dei cluster quando l'aumento della varianza diventa troppo elevato.

4. Altezza Massima dei Rami nell'Albero (Dendrogramma) L'altezza rappresenta la distanza tra i cluster, quindi si può scegliere di tagliare l'albero a una determinata altezza. Questo è un approccio visivo che può essere usato osservando il dendrogramma, dove si taglia in corrispondenza di un punto significativo per ottenere cluster ben distinti.

5. Metodo della Silhouette Si può utilizzare l'indice di silhouette per determinare il livello di taglio ottimale. Si esegue il clustering a diversi livelli (cioè diversi numeri di cluster) e si calcola la silhouette media per ciascuno. Il numero di cluster con la silhouette media più alta rappresenta il punto in cui l'albero dovrebbe essere tagliato per ottenere la migliore coesione e separazione tra cluster.

6. Gap Statistic. Confronta la dispersione intra-cluster per il clustering gerarchico reale con quella di un clustering generato da dati casuali. Si taglia l'albero nel punto in cui la differenza tra la dispersione reale e quella attesa dai dati casuali è massima, individuando un numero di cluster "ottimale".

Verifica delle dissimilarità/similarità: indice cofrenetico

- L'indice cophenetico è una misura che valuta quanto bene un dendrogramma rappresenta le distanze originali tra i punti.

- Si basa sulla matrice delle distanze cophenetiche, che contiene le distanze calcolate dal dendrogramma per ogni coppia di punti.

- Confronta queste distanze con quelle della matrice delle distanze originali (ad esempio, euclidea)

- Si calcola come correlazione tra due misure: la distanza tra i dati e le altezze a cui si uniscono i dati stessi (altezze dell'albero):

Formula completa dell'indice cophenetico

$$c = \frac{\sum_{i<j}(d_{ij} - \bar{d})(t_{ij} - \bar{t})}{\sqrt{\sum_{i<j}(d_{ij} - \bar{d})^2 \sum_{i<j}(t_{ij} - \bar{t})^2}}$$

- c: indice cophenetico, che varia tra -1 e 1, con valori vicini a 1 che indicano una buona fedeltà del dendrogramma.
- d_{ij}: distanza originale tra i punti i e j nella matrice delle distanze iniziali (ad esempio, distanze euclidee o altro criterio scelto).
- t_{ij}: distanza cophenetica tra i punti i e j, derivata dal dendrogramma.
- $\bar{d}$: media delle distanze nella matrice d_{ij}.
- $\bar{t}$: media delle distanze cophenetiche t_{ij}.

Come si calcola la **distanza cophenetica** (t_{ij})?

La **distanza cophenetica** per una coppia di punti i e j è la distanza al livello del dendrogramma in cui i due punti vengono uniti in un cluster per la prima volta.

Passaggi per calcolare t_{ij}:

1. **Costruisci il dendrogramma** usando un metodo di linkage (ad esempio: single linkage, complete linkage, average linkage, Ward).
2. Per ogni coppia di punti i, j:
    - Trova l'altezza del nodo nel dendrogramma in cui i due punti si uniscono per la prima volta.
    - Quell'altezza è la **distanza cophenetica** t_{ij}.

ESEMPIO:

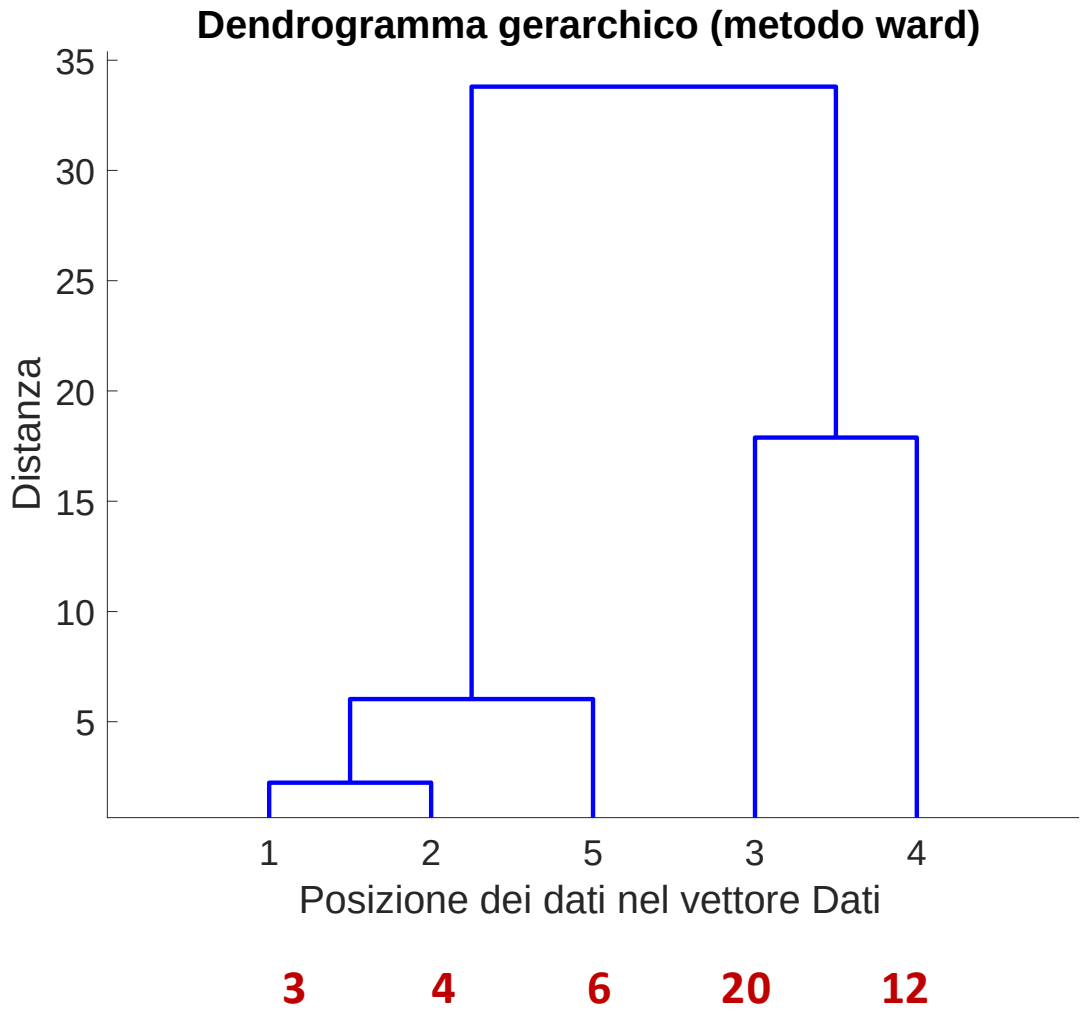
Dati= [3, 4, 20, 12, 6];

Distanze tra le coppie di dati (radice del quadrato):
[1, 17, 9, 3, 16, 8, 2, 8, 14, 6]

Distanze cofrenetiche:
[2.2, 33.8, 33.8, 6.02, 33.8, 33.8, 6.02, 17.89, 33.8, 33.8]

C = corr(Distanze tra le coppie di dati, Distanze cofrenetiche,'type','spearman')

Correlazione cophenetica: 0.79523



Interpretazione dell’indice cophenetico

Valori prossimi a 1:

- Il dendrogramma rappresenta fedelmente le distanze originali.
- Clustering affidabile.

Valori bassi (vicini a 0):

- Scarsa rappresentazione delle distanze originali.
- Necessità di rivedere il metodo di clustering o le metriche di distanza.

Esempio pratico:
Se l'indice cophenetico è 0.9, significa che il dendrogramma spiega il 90% delle variazioni delle distanze originali.

Vantaggi e limiti dell’indice cophenetico

Vantaggi:

- Fornisce un’indicazione numerica della qualità del clustering gerarchico.
- Aiuta a scegliere il metodo di linkage più adatto.

Limiti:

- Non valuta la qualità dei cluster finali (ma solo la fedeltà al modello delle distanze).
- Sensibile alla scelta della metrica di distanza iniziale.

Lo si può ritenere un indizio di come la struttura gerarchica dei dati, se presente, potrebbe essere riprodotta dal dendrogramma dal quale sono state ricostruite le distanze.

La correlazione cofenetica è tendenzialmente positiva (perché riferita ad ordinamenti simili) ed in genere elevata anche per ricostruzioni molto approssimative.

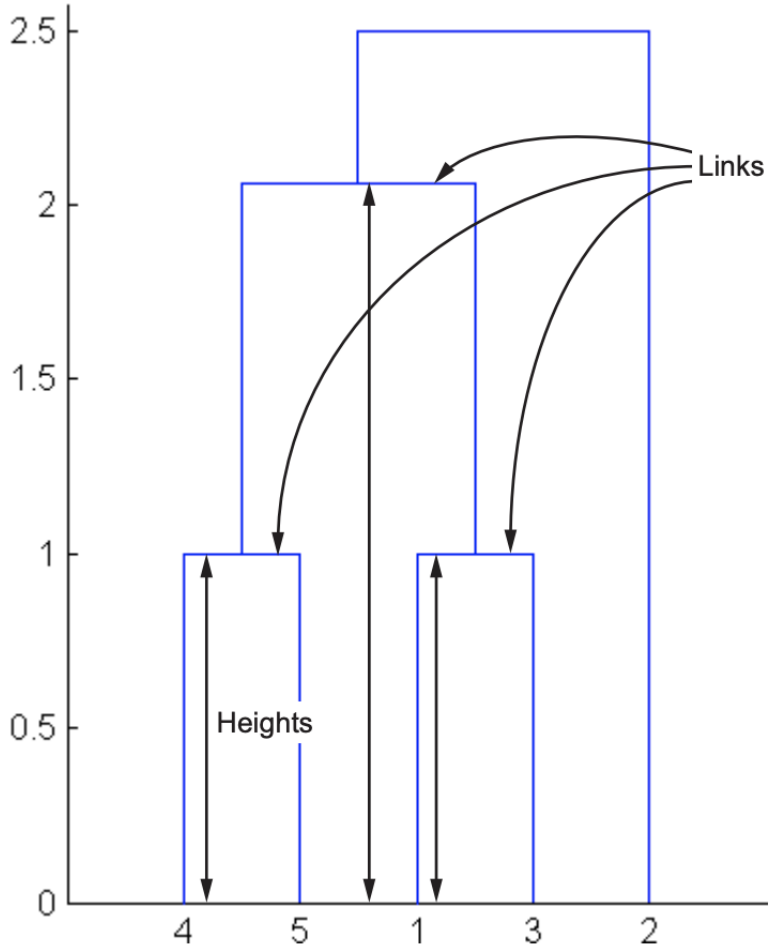
Solo i valori molto alti, diciamo superiori a 0.90 (meglio 0.95) possono essere considerati realmente significative

Valori ridotti, diciamo inferiori a 0.70, rendono contestabile la riproduzione della matrice delle distanze o dissimilarità e, forse, il data set non ha clusters o non ha gruppi nella forma che il legame adoperato è in grado di riconoscere.

Conclusione:
L'indice cophenetico è un utile strumento per validare il clustering gerarchico,
ma dovrebbe essere usato insieme ad altri metodi di valutazione.

Verifica delle dissimilarità/similarità: indice di Inconsistenza

- L'indice di inconsistenza è una misura utilizzata per valutare quanto una fusione in un dendrogramma sia coerente con le fusioni vicine.
- Si calcola confrontando l'altezza (h) di una fusione con la media e la deviazione standard delle altezze delle fusioni vicine.
- Serve a individuare punti di fusione "anomali", dove due cluster vengono uniti a una distanza significativamente maggiore rispetto alle precedenti.



Per una fusione k che combina due cluster:

$$\text{Inconsistency Index}_k = \frac{h_k - \text{mean}(\{h\})}{\text{std}(\{h\})}$$

Dove:

- h_k = altezza della fusione corrente.
- $\{h\}$ = altezze delle fusioni vicine (fino a un certo livello gerarchico).

l valore della soglia per l'indice di inconsistenza dipende dal dataset e dall'obiettivo dell'analisi. Non esiste un valore universale, ma ci sono alcune linee guida comuni:

Valori comuni: Tra 0.5 e 2.0.

Soglia ≈ 1.0	è un buon punto di partenza, che spesso individua cluster ben separati.
Soglia<1.0	Permette di rilevare cluster più piccoli e dettagliati, ma potrebbe creare gruppi rumorosi.
Soglia>1.5	Individua cluster più grandi e meno dettagliati

Esempio di strategia

- *Imposta una soglia iniziale di 1.0.*
- *Osserva il numero di cluster risultanti e la loro coerenza.*
- *Adatta la soglia aumentando o diminuendo per ottenere un risultato significativo per la tua analisi.*

ESEMPIO:

Dati= [3, 4, 20, 12, 6];

Indice di inconsistenza = 1