CLUSTER GERARCHICO

Questi tipi di algoritmi si basano sull'uso della dissimilarity matrix come criterio per formare i cluster. Non richiede l'uso del parametro K, ma necessitano la definizione di una stop condition.

AGGLOMERATIVO

Partendo con ogni oggetto appartenente a un singolo cluster per arrivare a tutti gli oggetti nel medesimo cluster

DIVISIVO

Esattamente l'opposto. Da tutti nello stesso cluster a ognuno nel proprio.

CONNETTIVITY MATRIX

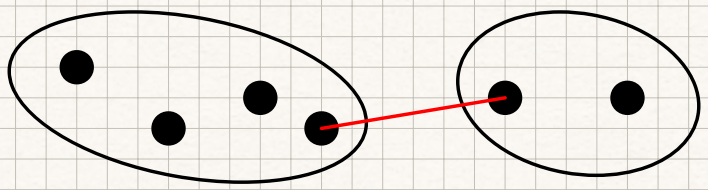
Struttura dati per decidere se fondere o splittare due cluster.

LINKAGE METRIC

Per fare il merge o lo split di un insieme di punti, bisogna considerare la distanza fra due sottoinsiemi di punti. Questa metrica è detta LINKAGE ed è costruita a partire dalla CONNETTIVITY MATRIX.
Ci sono diversi tipi di linkage metric, e questi danno risultati diversi nella ricerca dei cluster.

1) Linkage single link

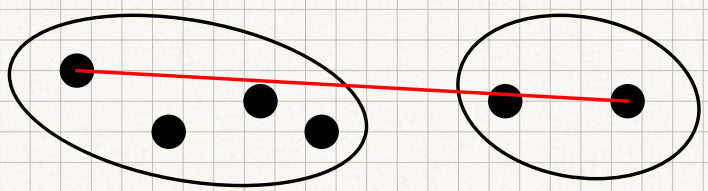
È la distanza fra i due oggetti più vicini appartenenti a due cluster diversi



$$d(C_1, C_2) = \min \{ dist(x, y) \mid x \in C_1 \; y \in C_2 \}$$

2) Complete Link

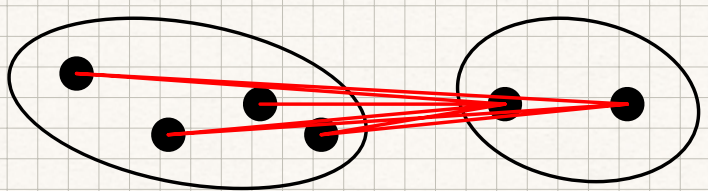
Questa volta valutiamo la distanza più grande fra due punti di cluster diversi



$$d(C_1, C_2) = \max \{ dist(x, y) \mid x \in C_1 \; y \in C_2 \}$$

3) Pair-Group Average

Questa volta consideriamo la distanza media fra tutte le coppie di oggetti dei due cluster.

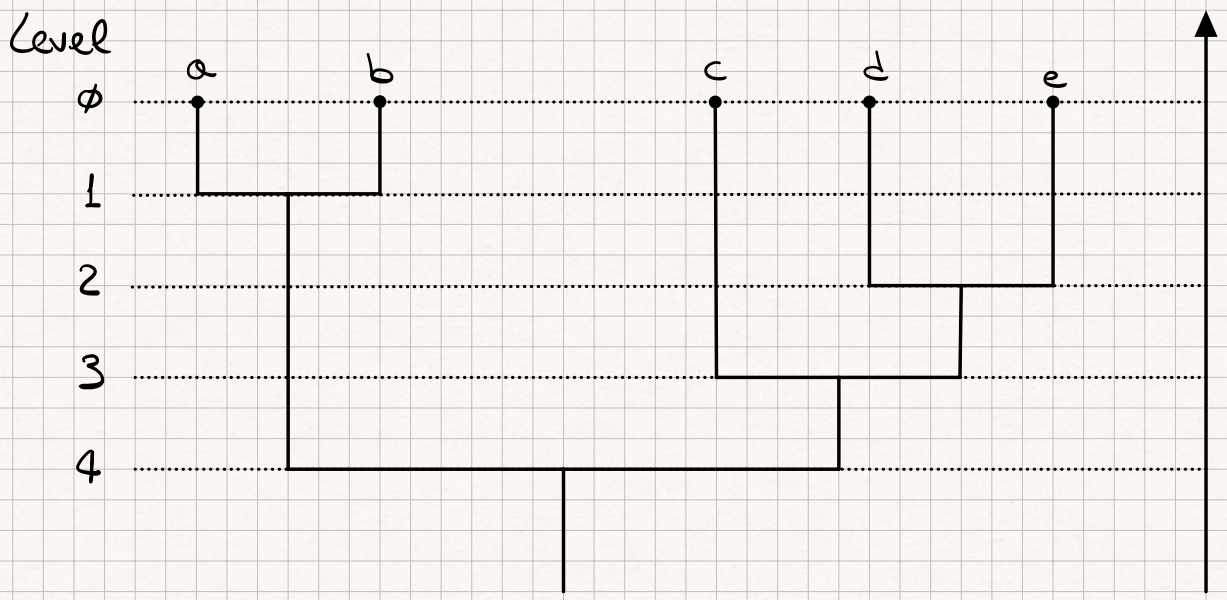


$$d(C_1, C_2) = avg \{ dist(x, y) \mid x \in C_1 \; y \in C_2 \}$$

$$d(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{\substack{p \in C_i \\ p \in C_j}} |p - p'|$$

DENDOGRAM

Struttura dati ad albero utile a visualizzare i passi della creazione di un cluster gerarchico.



AGNES

È un algoritmo di cluster gerarchico agglomerativo.

1) Ogni punto è assegnato a un cluster distinto così che inizialmente ogni cluster ha un solo punto
2) Merge procedure. Usando una procedura di merge si mettono insieme i cluster più simili. Può essere una delle metriche Link-age viste.
   Fa il merge dei due cluster che hanno linkage minore.
3) Si ripete potenzialmente finché tutti i punti non appartengono a un solo cluster.

Esempio

Dissimilarity Matrix

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

La distanza più piccola nella matrice è 1 per AB e CD:

AUB
CUD

Ora bisogna ricalcolare la dissimilarity matrix

	AB	CD	E
AB	0	2	3
CD	2	0	3
E	3	3	0

Uniamo AB ∪ CD

	ABCD	E
ABCD	0	3
E	3	0

Uniamo ABCD ∪ E

DIANA

È l'inverso di AGNES, e opta per un approccio divisivo, partendo da un solo grande cluster con tutto D dentro.

L'intuizione dietro DIANA è quella di dividere il cluster preso in considerazione in due, partendo dal cercare il gruppo di punti che ha la maggiore dissimilarità e lo etichetta come splitter group. Poi l'algoritmo riassegna i punti più simili allo splitter group. Risultato: due cluster diversi

ALGORITMO

1) Trova l'oggetto in D con la maggiore dissimilarità media con gli altri oggetti. Questo oggetto inizializza lo splinter group (SG)

2) $\forall o_i \notin SG$ calcola:

$$D_i = [AVG\{dist(i,j) \mid j \notin SG\}] - [AVG\{dist(i,j) \mid j \in SG\}]$$

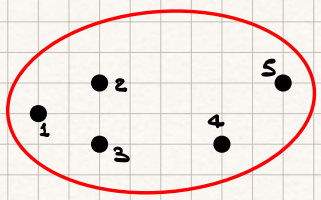
3) Trovare l'oggetto h per cui D_h è la maggiore.
if $(D_h > \emptyset)$ then h è in media più vicino a SG.

4) Ripetere finché tutte le $D_h < \emptyset$. Il dataset è diviso così in due cluster.

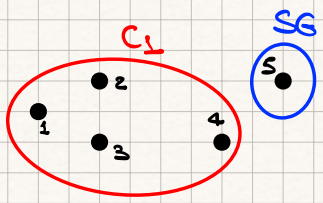
5) Selezionare ora il cluster con diametro maggiore dove con diametro si intende la distanza maggiore fra due oggetti dello stesso cluster. Ripetere i punti 1-4 su questo cluster.

6) Ripetere il tutto finché ogni punto avrà un suo cluster.

ESEMPIO



Iniziamo con un solo cluster. Dobbiamo cercare l'oggetto mediamente più dissimile, quello che ha distanza media dagli altri maggiore, e lo mettiamo nello splitter group (SG).



$\forall p_i \in C_1$ calcoliamo D_i:

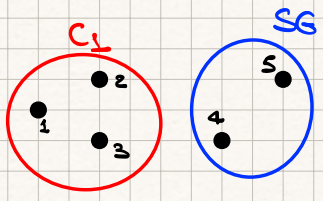
$$D_1 = \frac{1+1+3}{3} - \frac{4}{1} = -2.33$$

$$D_2 = \frac{1+1+2.5}{3} - \frac{3}{1} = -1.5$$

$$D_3 = \frac{1+1+2}{3} - \frac{3.5}{1} = -2.17$$

$$D_4 = \frac{3+2.5+2}{3} - \frac{1}{1} = +1.5$$

Mettiamo P_4 nello SG.



Calcoliamo $D_{i=1,2,3}$

$D_1 = 1 - 3.5 = -2.5$

$D_2 = 1 - 2.75 = -1.75$

$D_3 = 1 - 2.75 = -1.75$

Tutti $D_i < 0$. Abbiamo trovato due cluster distinti.

PRO CLUSTER GERARCHICI

1) Nessun bisogno di specificare K. Per avere K cluster basta tagliare il (K-1)-esimo link più lungo.

2) In generale escono fuori cluster migliori rispetto agli algoritmi come K-means.

CONTRO CLUSTER GERARCHICO

1) Non si può ritornare indietro una volta fatta una iterazione

2) Non è scalabile, e ha una complessità $O(n^2)$