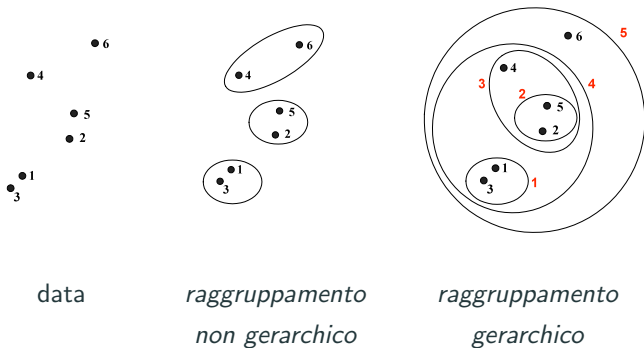


Cluster Analysis

Metodi di raggruppamento

R. Pappadà (rpappada@units.it)

Analisi dei dati per l'impresa, a.a. 25-26



Metodi non gerarchici Le tecniche di aggregazione non gerarchica sono finalizzate alla costruzione di una partizione dei dati in un numero di classi prefissato K .

Metodi gerarchici procedure sequenziali che restituiscono una partizione annidata di gruppi rappresentati da una struttura ad albero (dendrogramma)

Metodi non gerarchici: il metodo delle K medie

Date n unità su cui sono misurate p variabili e fissato un numero $K \geq 2$, l'obiettivo è trovare una partizione ottimale in K gruppi disgiunti C_1, \dots, C_K , tali che

- $C_1 \cup C_2 \cdots \cup C_K$ comprenda tutte le unità di indici $\{1, \dots, n\}$ (tutte le unità sono collocate in un gruppo)
- $C_j \cap C_{j'} = \emptyset$ for all $j \neq j'$ (ciascuna unità appartiene ad un solo gruppo)

Osservazione: una partizione *buona* è quella per cui la **variabilità all'interno del cluster** è più piccola possibile \rightarrow si vuole minimizzare la distanza tra le osservazioni all'interno di ciascun cluster

Metodo K -means: formulazione del problema

Indichiamo con $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ l' i -ma unità. Fissato il numero di gruppi K , il problema è quello di trovare una partizione tale che sia minima la *variabilità interna ai gruppi*. La quantità da minimizzare è

$$\sum_{k=1}^K W(C_k)$$

dove occorre definire meglio come calcoliamo la quantità $W(C_k)$ (**within-cluster variation**) per ciascun cluster. Si ricorre alla *distanza euclidea al quadrato* e si definisce

$$W(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2$$

dove $|C_k|$ è il numero di unità nel cluster k -esimo.

Il problema di ottimizzazione si può allora riscrivere come segue

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K W(C_k) = \min_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\} \quad (1)$$

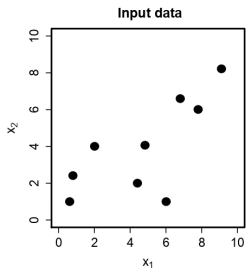
- l'esplorazione di tutte le possibili partizioni per scegliere la migliore non è percorribile
- si può dimostrare che l'algoritmo utilizzato e introdotto da MacQueen (1967) fornisce un ottimo locale al problema delle K -medie

L'algoritmo K-means

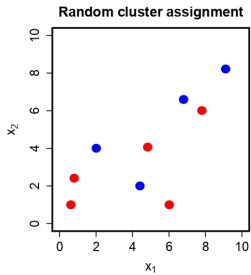
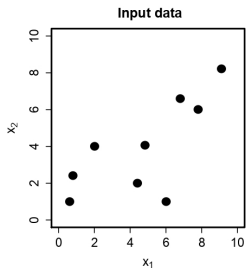
Fissato $K \geq 2$

1. **Assegnazione iniziale:** assegnare casualmente ogni unità ad un cluster da 1 a K
2. **Riassegnazione:** Ripetere fino a quando la configurazione non si stabilizza:
 - a. si calcola il *centroide* per ogni gruppo, cioè il vettore delle medie sulle p variabili con riferimento alle unità del gruppo
 - b. si assegna ciascuna unità al cluster che è più vicino (distanza euclidea calcolata rispetto al centroide)

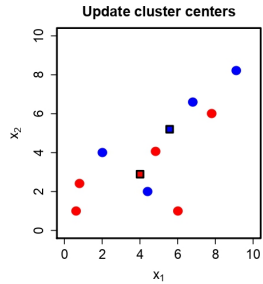
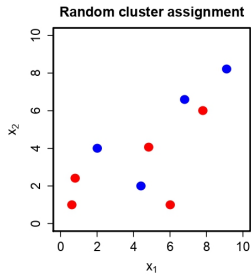
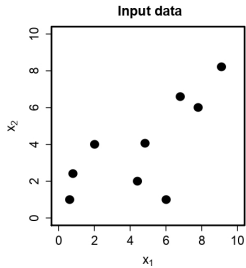
K-means: esempio



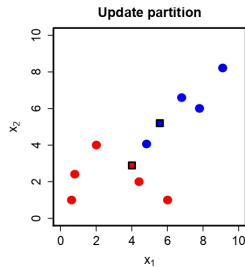
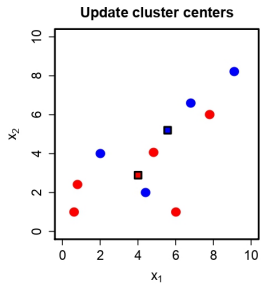
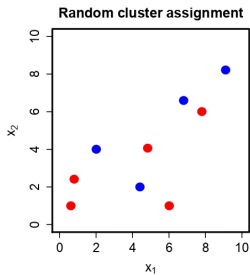
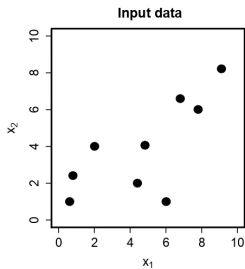
K-means: esempio



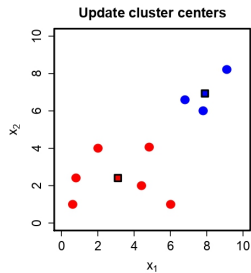
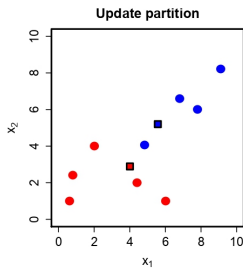
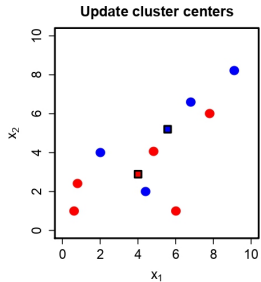
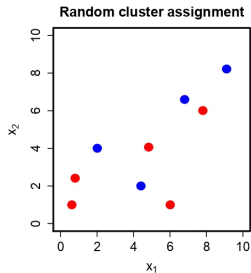
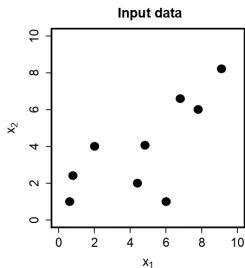
K-means: esempio



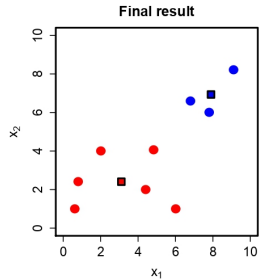
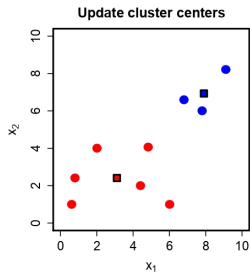
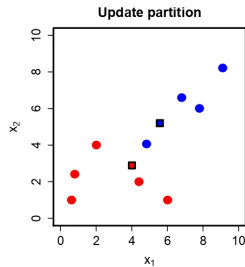
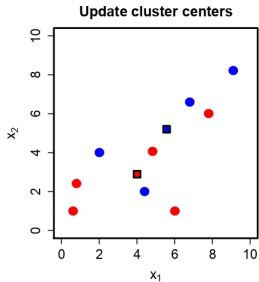
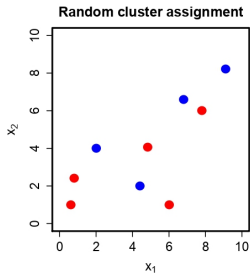
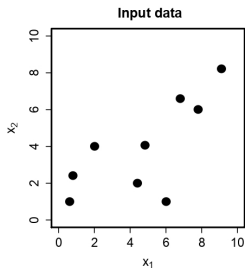
K-means: esempio



K-means: esempio



K-means: esempio



K-means: funzione obiettivo

L'algoritmo K-means garantisce che la funzione obiettivo nell'Eq.(1) diminuisca ad ogni step.

Infatti, si ha

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2,$$

dove $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ è la media per la variabile j -esima nel cluster C_k .

- In 2(a) le medie di gruppo sono le costanti che minimizzano gli scostamenti al quadrato
- In 2(b) le unità vengono riassegnate per migliorare l'assegnazione finale
- quando la configurazione rimane invariata è stato raggiunto un ottimo locale.

K-means: funzione obiettivo (cont)

Poiché l'algoritmo *K*-means trova un *ottimo locale* e non globale, i risultati ottenuti dipenderanno dall'assegnazione iniziale (casuale) del cluster di ciascuna osservazione nello **Step 1** dell'algoritmo.



Per questa ragione, nella pratica si esegue l'algoritmo più volte con diverse assegnazioni iniziali e poi si seleziona la soluzione migliore, cioè quella per cui la quantità

$$\sum_{k=1}^K W(C_k)$$

è più piccola

Il metodo *K*-means è molto utilizzato in svariati campi grazie alla sua semplicità, velocità ed efficienza.



Le principali limitazioni dell'algoritmo *K*-means sono:

- la convergenza ad un minimo locale può produrre risultati indesiderati
- è sensibile alla diversa assegnazione iniziale e conseguente scelte dei baricentri
- l'algoritmo *K*-means funziona male nel caso di cluster di dimensioni non omogenee o forme non sferiche
- una scelta inappropriata di *K* potrebbe dare scarsi risultati

Algoritmo PAM - Partitioning Around Medoids

Analogamente al metodo K -means, **PAM** è una procedura iterativa che assegna ciascun punto dato a uno e un solo gruppo da 1 a K



A differenza dell'algoritmo K -means, PAM individua una unità rappresentativa per ciascun gruppo tra le unità del dataset



Tali unità, dette **medoidi**, vengono utilizzate per assegnare le altre unità ad un gruppo sulla base della più piccola dissimilarità tra le unità e i medoidi



Un vantaggio importante di PAM è che è più robusto rispetto alla presenza di osservazioni anomale rispetto all'algoritmo K -means

K-means VS PAM (Esempio per $K = 2$)

Consideriamo i dati rappresentati nelle figure (a) e (b): i gruppi sono ottenuti rispettivamente con i metodi *K*-means e PAM. Si noti come i due raggruppamenti siano molto simili.

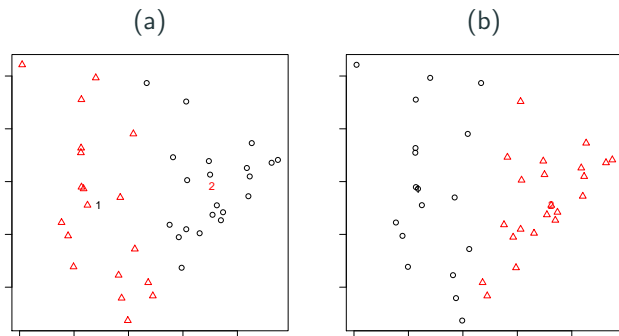


Figura 1: *K*-Means (a) e PAM (b): i centroidi e i medoidi in (a) e (b), rispettivamente, sono riportati con i numeri 1 e 2.

K -means VS PAM (Esempio per $K = 2$)

Il campione viene quindi contaminato e si può vedere che il raggruppamento ottenuto con K -means è cambiato sostanzialmente, non è così per il raggruppamento ottenuto con PAM.

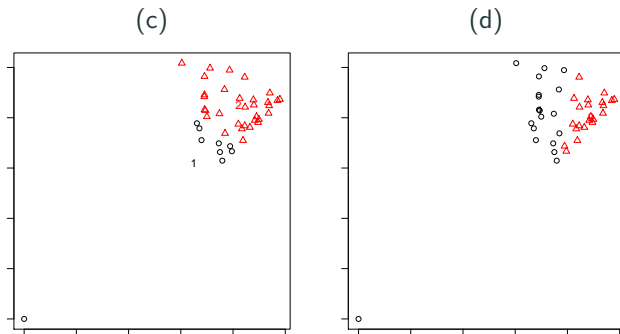


Figura 2: K -Means (c) e PAM (d) con dati contaminati

Valutazione di un raggruppamento

Vedremo nei laboratori che vi sono diversi indici che possono dare informazioni sulla *bontà* di una partizione, e quindi anche dare indicazioni sul K da scegliere.

In R, molti pacchetti contengono funzioni per gli indici più comuni in letteratura (si veda, ad esempio, il pacchetto `fpc`, `cluster`)



Esempio: la *Silhouette*: data una partizione in K gruppi di n unità basata sulla matrice di dissimilarità D , per l' i -ma unità nel cluster $C^{(i)}$ si calcola

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

$a(i)$ = dissimilarità media tra i e tutti gli altri punti nello stesso cluster;

$b(i) := \min_C d(i, C)$, dove $d(i, C)$ è la dissimilarità media tra i e tutti gli altri punti di C , cluster diverso da $C^{(i)}$

Se $s(i)$ è vicino a 1 l'unità è correttamente assegnata al cluster, un valore basso o negativo di $s(i)$ indica una criticità