



Università
degli Studi
di Palermo



Clustering

CORSO DI BIG DATA – MODULO ANALISI PER I BIG DATA

a.a. 2023/2024

Prof. Roberto Pirrone



Sommario

- Generalità
- Feature selection
- Clustering basato su prototipi rappresentativi
- Clustering gerarchico
- Clustering probabilistico
- Clustering per densità
- Misure di bontà del clustering

Generalità

- Il clustering è il processo di partizionamento di un data set in modo che ogni partizione (o *cluster*) contenga dati molto simili tra loro che rispetto agli appartenenti ad altre partizioni
- Rappresenta una classe di algoritmi di apprendimento non supervisionati che sono i più utilizzati nel *data mining*
 - Data summarization
 - Customer segmentation, collaborative filtering
 - Social network analysis
 - ...

Feature selection

- In generale non tutte le feature presenti in un campione sono rilevanti per il compito di clustering e, anzi, possono introdurre una certa «rumorosità» nei dati se prese in considerazione
- È necessario impiegare dei modelli di feature selection che cercano di individuare la «tendenza al clustering» di certi sottoinsiemi di feature
 - Modelli basati su filtro: viene valutata la tendenza di una feature o di un gruppo di feature a contribuire al clustering utilizzando uno score numerico
 - Modelli «wrapper»: approcci iterativi di clustering per tentativi su sottoinsiemi di feature
 - Computazionalmente onerosi, ma possono fornire una base per la scelta del miglior algoritmo di clustering per il problema sotto esame

Feature selection

- Modelli basati su filtro

$$\text{Term Strength} = P(t \in \bar{Y} | t \in \bar{X})$$

- Usata per analisi di dati testuali
- Viene calcolata come la frazione delle coppie di documenti simili in cui occorre un certo termine t , posto che t occorra nel primo dei due
- Determina «termini rilevanti» rispetto al data set

Feature selection

- Modelli basati su filtro

$$\text{Term Strength} = P(t \in \bar{Y} | t \in \bar{X})$$

- È una **similarità** che può essere estesa a dati multidimensionali rappresentati come vettori binari di attributi presenti/assenti
- Analogamente si possono usare misure di rilevanza rispetto a dati quantitativi confrontando la distanza tra singoli attributi di due campioni rispetto alla distanza complessiva dei due campioni

Feature selection

- Modelli basati su filtro
 - Dipendenza predittiva da un attributo
 - Assunto: gli attributi rilevanti per il clustering sono correlati
 - Come fare a trovare gli attributi rilevanti?

Feature selection

- Modelli basati su filtro
 - Dipendenza predittiva da un attributo
 - Usa modelli di classificazione/regressione (a seconda del tipo di attributo) per classificare il data set rispetto a tutte le feature tranne la i -esima che viene considerata come classe di appartenenza del campione
 - Spesso si usa la classificazione nearest neighbor
 - La misura di accuratezza della classificazione rispetto all'attributo i sarà la misura di rilevanza utilizzata per la feature selection

Feature selection

- Modelli basati su filtro
 - Entropia
 - I dati raggruppati in un cluster hanno entropia più bassa che quelli distribuiti uniformemente nello spazio di variazione
 - Affronteremo il problema di stabilire qual'è il subset delle k feature più rilevanti per clusterizzare i nostri campioni

Feature selection

- Modelli basati su filtro
 - Entropia
 - Discretizziamo i valori di ognuna delle feature in ϕ intervalli
 - Otteniamo $m = \phi^k$ regioni discrete nello spazio dei dati

Feature selection

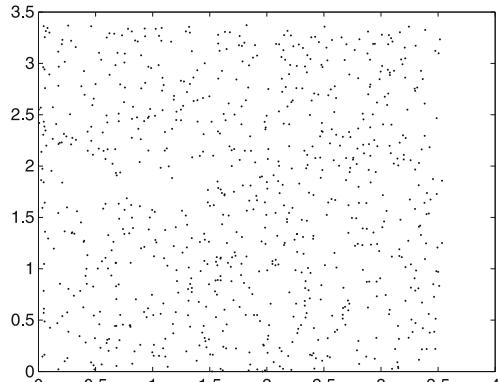
- Modelli basati su filtro
 - Entropia
- Si calcolano le «distanze punto a punto» tra i campioni lungo ogni feature e, se m sono gli intervalli di discretizzazione dei valori di distanza, si ha:

$$E = - \sum_{i=1}^m [p_i \log(p_i) + (1 - p_i) \log(1 - p_i)]$$

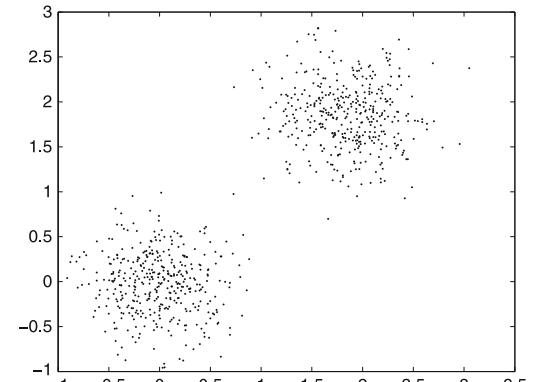
$$p_i = \frac{N_{d_i}}{N_d}, \quad d_i \in \text{i-th interval}, \quad i = 1, 2, \dots, m$$

Feature selection

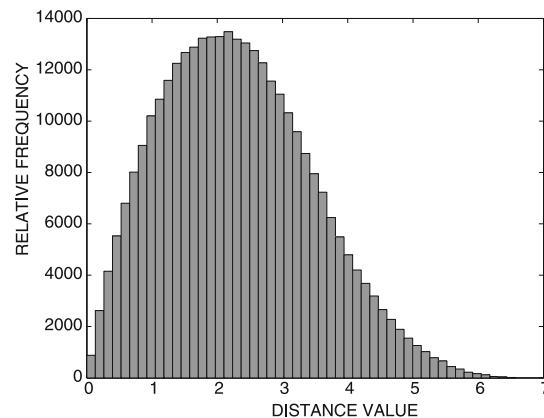
- Modelli basati su filtro
 - Entropia
 - Si scelgono le feature con entropia minore
 - Approccio greedy: si eliminano le feature che portano via via alla maggiore diminuzione di E



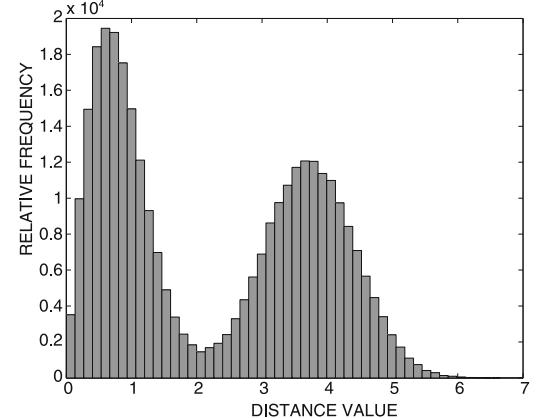
(a) Uniform Data



(b) Clustered data



(c) Distance distribution (uniform)



(d) Distance distribution (clustered)

Feature selection

- Modelli basati su filtro
 - Hopkins Statistic
 - Si generi un campione $R = \{r_1, \dots, r_r\} \subset \mathcal{D}$
 - e un campione sintetico $S = \{s_1, \dots, s_r\} \subset \mathbb{R}^d$, lo spazio dei dati
 - Siano $\alpha_i = \min_x \text{Dist}(r_i, x)$ e $\beta_i = \min_x \text{Dist}(s_i, x)$ $i = 1, \dots, r$ e $x \in \mathcal{D}$

Feature selection

- Modelli basati su filtro
 - Hopkins Statistic

$$H = \frac{\sum_{i=1}^r \beta_i}{\sum_{i=1}^r (\alpha_i + \beta_i)}$$

- H tende a 0.5 per dati distribuiti uniformemente perché α_i e β_i hanno comportamento simile, mentre tende a 1 per dati clusterizzati perché i coefficienti α_i saranno molto più piccoli dei β_i

Feature selection

- Modelli wrapper
 - Si effettua una serie di clustering su differenti sottoinsiemi di feature e si verifica di volta in volta la qualità ottenuta con una apposita *misura interna di bontà del clustering*
 - Computazionalmente oneroso
 - Necessita di un approccio greedy per la ricerca dei sottoinsiemi rilevanti nello spazio delle feature
 - Fortemente dipendente dalla scelta della misura di bontà del clustering

Feature selection

- Modelli wrapper
 - Alternativamente si effettua un clustering sul sottoinsieme di feature considerate e si utilizzano «artificialmente» le etichette dei cluster ottenute come etichette di classe
 - Si utilizza un *criterio di bontà di classificazione supervisionata* su tali etichette per valutare indirettamente la bontà del clustering
 - Si selezionano le prime k feature con il valore migliore dell'indice di qualità di classificazione

Clustering basato su prototipi rappresentativi

Algorithm *GenericRepresentative*(Database: \mathcal{D} , Number of Representatives: k)

begin

 Initialize representative set S ;

repeat

 Create clusters $(\mathcal{C}_1 \dots \mathcal{C}_k)$ by assigning each
 point in \mathcal{D} to closest representative in S
 using the distance function $Dist(\cdot, \cdot)$;

 Recreate set S by determining one representative \bar{Y}_j for
 each \mathcal{C}_j that minimizes $\sum_{X_i \in \mathcal{C}_j} Dist(\bar{X}_i, \bar{Y}_j)$;

until convergence;

return $(\mathcal{C}_1 \dots \mathcal{C}_k)$;

end

→ Passo di assegnazione

→ Passo di ottimizzazione

Clustering basato su prototipi rappresentativi

- I k cluster sono in genere un iperparametro determinato dall'utente
- Si devono determinare k prototipi rappresentativi \bar{Y}_j che minimizzino la funzione obiettivo

$$O = \sum_{j=1}^k \sum_{\overline{X_i} \in \mathcal{C}_j} Dist(\overline{X_i}, \overline{Y_j})$$

- I valori degli \bar{Y}_j in generale **non appartengono** al data set \mathcal{D}
- La scelta di diverse forme di $Dist(.,.)$ dà luogo a differenti algoritmi

Clustering basato su prototipi rappresentativi

- K-means

$$Dist(\overline{X_i}, \overline{Y_j}) = \|\overline{X_i} - \overline{Y_j}\|_2^2$$

- Si può mostrare che i valori ottimi degli \bar{Y}_j sono i valori medi dei punti in ogni singolo cluster \mathcal{C}_j

Clustering basato su prototipi rappresentativi

- Mahalanobis K-means

$$Dist(\overline{X_i}, \overline{Y_j}) = (\overline{X_i} - \overline{Y_j})\Sigma_j^{-1}(\overline{X_i} - \overline{Y_j})^T$$

- La distanza di Mahalanobis è opportuna per cluster di forma ellittica
- La covarianza locale Σ_j , calcolata sui punti del cluster \mathcal{C}_j :
 - fornisce una misura implicita di densità del cluster e quindi consente di gestire cluster a densità variabile
 - consente la gestione di cluster di forma variabile

Clustering basato su prototipi rappresentativi

- Il k-means è adatto per cluster di forma sferica anche con densità variabile (Mahalanobis k-means)

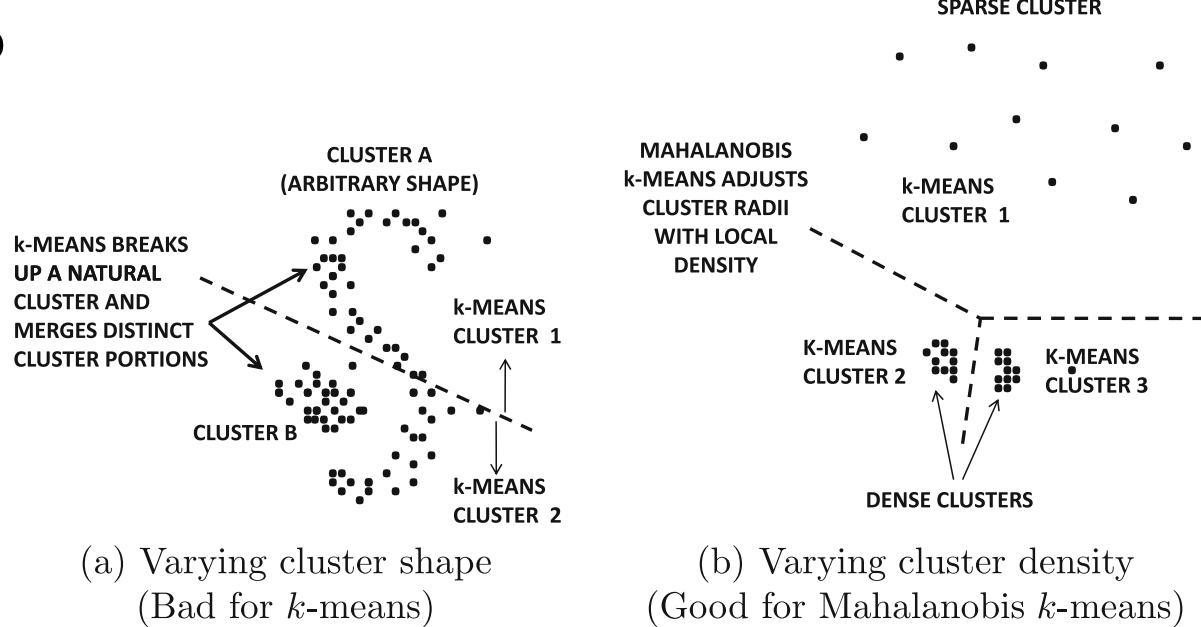


Figure 6.4: Strengths and weaknesses of k -means

Clustering basato su prototipi rappresentativi

- Per la gestione di cluster di forma qualunque, il k-means ricorre al cosiddetto *kernel trick*
 - Si consideri una funzione $\Phi(\cdot)$ che mappa un vettore in uno spazio a più alta dimensionalità
 - Si definisce «kernel» una qualunque funzione tale che:

$$K(\overline{X_i}, \overline{X_j}) = \Phi(\overline{X_i}) \cdot \Phi(\overline{X_j})$$

Clustering basato su prototipi rappresentativi

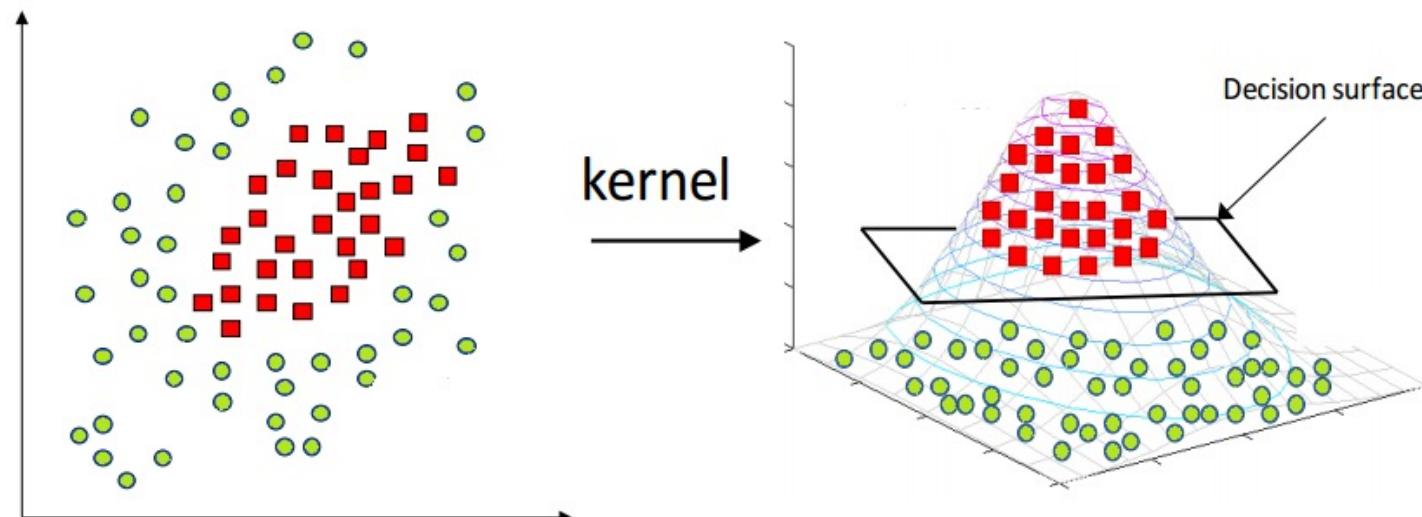
- Per la gestione di cluster di forma qualunque, il k-means ricorre al cosiddetto *kernel trick*
 - L'uso del kernel è un operatore lineare nello spazio trasformato e quindi consente l'uso di algoritmi lineari nello spazio definito da $\Phi(\cdot)$
 - Il kernel $K(\cdot, \cdot)$ viene definito direttamente in termini dei vettori dello spazio di ingresso e non è *mai* necessario calcolare il mapping $\Phi(\cdot)$

$$K(\overline{X_i}, \overline{X_j}) = \Phi(\overline{X_i}) \cdot \Phi(\overline{X_j})$$

Uso di *euristiche* e di *algoritmi di ML dedicati* per
selezionare il kernel K più adatto ad un dato
comitato

Clustering basato su prototipi rappresentativi

- Per la gestione di cluster di forma qualunque, il k-means ricorre al cosiddetto *kernel trick*



Clustering basato su prototipi rappresentativi

- Per la gestione di cluster di forma qualunque, il k-means ricorre al cosiddetto *kernel trick*

- La nuova funzione obiettivo, tenendo conto del mapping $\Phi(\cdot)$ è:

$$O = \sum_{j=1}^k \sum_{\overline{X_i} \in \mathcal{C}_j} \left\| \Phi(\overline{X_i}) - \overline{Y_j} \right\|_2, \quad \overline{Y_j} = \frac{\sum_{\overline{X_i} \in \mathcal{C}_j} \Phi(\overline{X_i})}{|\mathcal{C}_j|}$$

$$O = \Phi(\overline{X_i}) \cdot \Phi(\overline{X_i}) - 2 \frac{\sum_{\overline{X_l} \in \mathcal{C}_j} \Phi(\overline{X_i}) \cdot \Phi(\overline{X_l})}{|\mathcal{C}_j|} + \frac{\sum_{\overline{X_l}, \overline{X_k} \in \mathcal{C}_j} \Phi(\overline{X_l}) \cdot \Phi(\overline{X_k})}{|\mathcal{C}_j|^2} =$$

$$O = K(\overline{X_i}, \overline{X_i}) - 2 \frac{\sum_{\overline{X_l} \in \mathcal{C}_j} K(\overline{X_i}, \overline{X_l})}{|\mathcal{C}_j|} + \frac{\sum_{\overline{X_l}, \overline{X_k} \in \mathcal{C}_j} K(\overline{X_l}, \overline{X_k})}{|\mathcal{C}_j|^2}$$

Clustering basato su prototipi rappresentativi

- K-medians

$$Dist(\overline{X_i}, \overline{Y_j}) = \|\overline{X_i} - \overline{Y_j}\|_1$$

- Uso della distanza di Manhattan
- Robusto rispetto agli outlier a causa dell'uso dei mediani

Clustering basato su prototipi rappresentativi

- K-medoids
- Impone che i valori degli \bar{Y}_j appartengano a \mathcal{D}
- Adattabile a dati eterogenei con la giusta definizione dell'appropriata misura di distanza/similarità
- Computazionalmente oneroso

```
Algorithm GenericMedoids(Database:  $\mathcal{D}$ , Number of Representatives:  $k$ )
begin
    Initialize representative set  $S$  by selecting from  $\mathcal{D}$ ;
    repeat
        Create clusters  $(\mathcal{C}_1 \dots \mathcal{C}_k)$  by assigning
            each point in  $\mathcal{D}$  to closest representative in  $S$ 
            using the distance function  $Dist(\cdot, \cdot)$ ;
        Determine a pair  $\bar{X}_i \in \mathcal{D}$  and  $\bar{Y}_j \in S$  such that
            replacing  $\bar{Y}_j \in S$  with  $\bar{X}_i$  leads to the
            greatest possible improvement in objective function;
        Perform the exchange between  $\bar{X}_i$  and  $\bar{Y}_j$  only
            if improvement is positive;
    until no improvement in current iteration;
    return  $(\mathcal{C}_1 \dots \mathcal{C}_k)$ ;
end
```

Clustering gerarchico

- Il clustering gerarchico crea una tassonomia di cluster che vengono aggregati attraverso il calcolo di distanze tra essi
- Il calcolo di tali distanze utilizza in genere altri approcci come quelli basati sul concetto di densità o sull'utilizzo di grafi
- Approcci
 - Agglomerativi o bottom-up
 - Divisivi o top-down

Clustering gerarchico

- Approcci agglomerativi

```
Algorithm AgglomerativeMerge(Data:  $D$ )
begin
    Initialize  $n \times n$  distance matrix  $M$  using  $D$ ;
    repeat
        Pick closest pair of clusters  $i$  and  $j$  using  $M$ ;
        Merge clusters  $i$  and  $j$ ;
        Delete rows/columns  $i$  and  $j$  from  $M$  and create
            a new row and column for newly merged cluster;
        Update the entries of new row and column of  $M$ ;
    until termination criterion;
    return current merged cluster set;
end
```

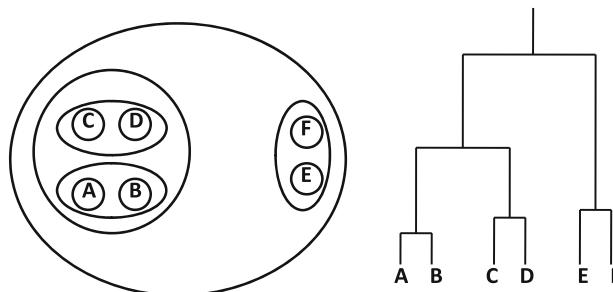
M è la struttura dati su cui lavorano tutti gli algoritmi gerarchici

- Due approcci:
1. Soglia massima sulla distanza tra due cluster da fondere
 2. Soglia minima sul numero di cluster

Clustering gerarchico

- Approcci agglomerativi

- Il calcolo delle distanze tra i cluster avviene incrementalmente via via che si aggiorna M
- In generale i cluster i e j contengono rispettivamente m_i e m_j elementi e necessitano di calcolare $m_i \times m_j$ distanze



(a) Dendrogram

Quale sarà la distanza tra i due cluster?

Si utilizzano delle statistiche per stimare la similarità tra gruppi di oggetti

Clustering gerarchico

- Stime di distanza per approcci agglomerativi, dopo aver fuso i cluster i e j
 - Single linkage – la distanza è la minima tra le $m_i \times m_j$ distanze possibili
 - Per ogni cluster $k \neq i,j$ si considerano $\min\{M_{ik}, M_{jk}\}$ per le righe e $\min\{M_{ki}, M_{kj}\}$ per le colonne di M
 - Complete linkage – la distanza è la massima tra le $m_i \times m_j$ distanze possibili
 - Per ogni cluster $k \neq i,j$ si considerano $\max\{M_{ik}, M_{jk}\}$ per le righe e $\max\{M_{ki}, M_{kj}\}$ per le colonne di M

Clustering gerarchico

- Stime di distanza per approcci agglomerativi, dopo aver fuso i cluster i e j
 - Group-average linkage – la distanza è la media tra le $m_i \times m_j$ distanze possibili
 - Per ogni cluster $k \neq i, j$ si considerano $(m_i M_{ik} + m_j M_{jk}) / (m_i + m_j)$ per le righe e $(m_i M_{ki} + m_j M_{kj}) / (m_i + m_j)$ per le colonne di M
 - Centroide più vicino – si fondono i cluster con i centroidi più vicini

Clustering gerarchico

- Approcci agglomerativi – varianza dei cluster
 - Si cerca di effettuare il merge quando la variazione della funzione obiettivo è la minima possibile per preservare le proprietà statistiche dell'agglomerato

$$SE_i = \sum_{r=1}^d \left(S_{ir}/m_i - F_{ir}^2/m_i^2 \right)$$

$S_{ir} = \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}_r^2$ Momento del II ordine di \mathcal{C}_i
sulla dimensione r

$F_{ir} = \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}_r$ Momento del I ordine di \mathcal{C}_i
sulla dimensione r

$$\text{Var} = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

Clustering gerarchico

- Approcci agglomerativi – varianza dei cluster
 - Si cerca di effettuare il merge quando la variazione della funzione obiettivo è la minima possibile per preservare le proprietà statistiche dell'agglomerato

$$SE_i = \sum_{r=1}^d \left(S_{ir}/m_i - F_{ir}^2/m_i^2 \right)$$

$$\Delta SE_{i \cup j} = SE_{i \cup j} - SE_i - SE_j$$

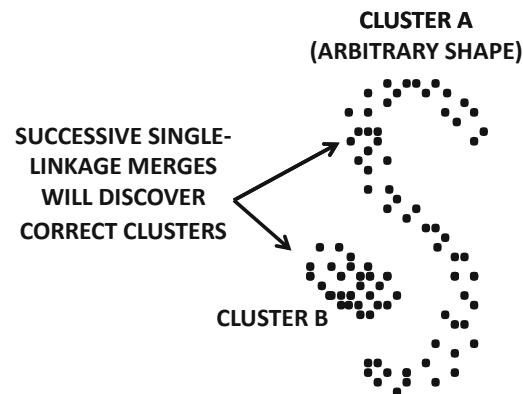
$$\Delta SE_{i \cup j \cup k} = SE_{i \cup j \cup k} - SE_{i \cup j} - SE_k, \quad k \neq i, j$$

$$S_{ir} = \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}_r^2 \text{ Momento del II ordine di } \mathcal{C}_i \text{ sulla dimensione } r$$

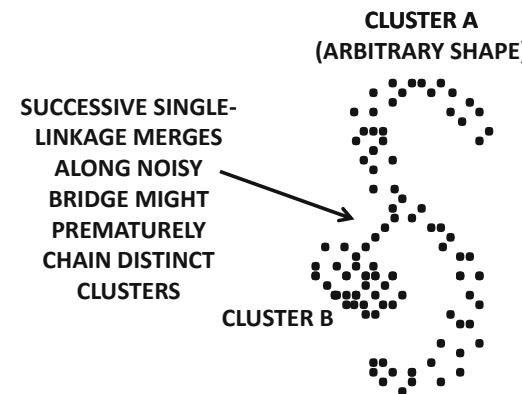
$$F_{ir} = \sum_{\mathbf{x} \in \mathcal{C}_i} \mathbf{x}_r \text{ Momento del I ordine di } \mathcal{C}_i \text{ sulla dimensione } r$$

Clustering gerarchico

- Approcci agglomerativi: limiti del clustering



(a) Good case with no noise



(b) Bad case with noise

Clustering gerarchico

- Approcci divisivi
 - \mathcal{A} può essere un algoritmo di clustering qualunque
 - Bisecting k-means: esegue un 2-means ad ogni iterazione
 - La scelta del nodo L può essere dettata da diversi criteri
 - Il nodo più vicino alla radice

Algorithm *GenericTopDownClustering*(Data: \mathcal{D} , Flat Algorithm: \mathcal{A})
begin

 Initialize tree \mathcal{T} to root containing \mathcal{D} ;

repeat

 Select a leaf node L in \mathcal{T} based on pre-defined criterion;

 Use algorithm \mathcal{A} to split L into $L_1 \dots L_k$;

 Add $L_1 \dots L_k$ as children of L in \mathcal{T} ;

until termination criterion;

end

Clustering probabilistico

- *Soft Clustering*: ogni punto ha una probabilità anche piccola, ma diversa da 0, di appartenere ad *ogni* cluster
- Considereremo un *modello generativo* \mathcal{M} che descrive la probabilità p_{data} di generare un punto appartenente ad un dato cluster
- Si assume che p_{data} sia una mistura di distribuzioni \mathcal{G}_i , che saranno i nostri cluster, con un insieme di prior $\alpha_i = P(\mathcal{G}_i)$

Clustering probabilistico

- Sia $f^i(\cdot)$ la forma funzionale di ogni \mathcal{G}_i
- La generazione di un punto dato il nostro modello è descritta da

$$f^{point}(\overline{X_j} | \mathcal{M}) = \sum_{i=1}^k \alpha_i \cdot f^i(\overline{X_j})$$

Clustering probabilistico

- Per l'intero data set si ha

$$f^{data}(\mathcal{D}|\mathcal{M}) = \prod_{j=1}^n f^{point}(\overline{X_j}|\mathcal{M})$$

Ipotesi i.i.d.

- Il clustering si ottiene massimizzando la log-likelihood di $f^{data}(\mathcal{D}|\mathcal{M})$

$$\mathcal{L}(\mathcal{D}|\mathcal{M}) = \log \left(\prod_{j=1}^n f^{point}(\overline{X_j}|\mathcal{M}) \right) = \sum_{j=1}^n \log \left(\sum_{i=1}^k \alpha_i f^i(\overline{X_j}) \right)$$

Clustering probabilistico

- La massimizzazione della likelihood si ottiene con un approccio iterativo a due passi detto *Expectation Maximization (EM)* che ci consente di stimare l'insieme Θ di tutti i parametri della mistura ed i prior α_i
- Usiamo lo schema del Teorema di Bayes

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}$$

Clustering probabilistico

- Expectation Maximization

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}$$

- Expectation: si effettua una stima di $P(\mathbf{w}|\mathcal{D})$ delle predizioni del modello fissati i parametri \mathbf{w}
- Maximization: si usa il nuovo valore dei parametri per eseguire uno step di massimizzazione di $P(\mathcal{D}|\mathbf{w})$

Clustering probabilistico

- Applichiamo l'algoritmo EM al nostro modello
- Expectation
 - Si stima col teorema di Bayes la probabilità a posteriori che un certo punto sia stato generato da una componente \mathcal{G}_i della mistura, dato Θ

$$P(\mathcal{G}_i | \overline{X_j}, \Theta) = \frac{P(\mathcal{G}_i) \cdot P(\overline{X_j} | \mathcal{G}_i, \Theta)}{\sum_{r=1}^k P(\mathcal{G}_r) \cdot P(\overline{X_j} | \mathcal{G}_r, \Theta)} = \frac{\alpha_i \cdot f^{i,\Theta}(\overline{X_j})}{\sum_{r=1}^k \alpha_r \cdot f^{r,\Theta}(\overline{X_j})}$$

Clustering probabilistico

- Applichiamo l'algoritmo EM al nostro modello
- Maximization:
 - Note le probabilità *stimate* di assegnazione di ciascun punto ad ogni cluster si ottengono i nuovi valori dei parametri α_i :

$$\alpha_i = P(\mathcal{G}_i) = \frac{\sum_{j=1}^n P(\mathcal{G}_i \mid \overline{X}_j, \Theta)}{n} \quad \text{ovvero} \quad \alpha_i = \frac{1 + \sum_{j=1}^n P(\mathcal{G}_i \mid \overline{X}_j, \Theta)}{k + n}$$

- Si massimizza la log-likelihood ponendo a 0 le derivate parziali di $\mathcal{L}(\mathcal{D} \mid \mathcal{M})$ rispetto a Θ e si calcola il nuovo valore quest'ultimo

Clustering probabilistico

- Si consideri una mistura di gaussiane in d dimensioni:

$$f^{i,\Theta}(\overline{X_j}) = \frac{1}{\sqrt{|\Sigma_i|}(2 \cdot \pi)^{(d/2)}} e^{-\frac{1}{2}(\overline{X_j} - \overline{\mu_i})\Sigma_i^{-1}(\overline{X_j} - \overline{\mu_i})}$$

- Si può mostrare che il passo di expectation in questo caso pesa i cluster in modo da assegnare i punti ai cluster **esattamente** come il Mahalanobis K-means

Clustering probabilistico

- Nel caso particolare in cui $\alpha_i = 1/k$ e tutte le componenti hanno deviazione standard uguale a σ :

$$f^{j,\Theta}(\overline{X}_i) = \frac{1}{(\sigma\sqrt{2\pi})^d} e^{-\left(\frac{\|\overline{X}_i - \overline{Y}_j\|^2}{2\sigma^2}\right)}$$

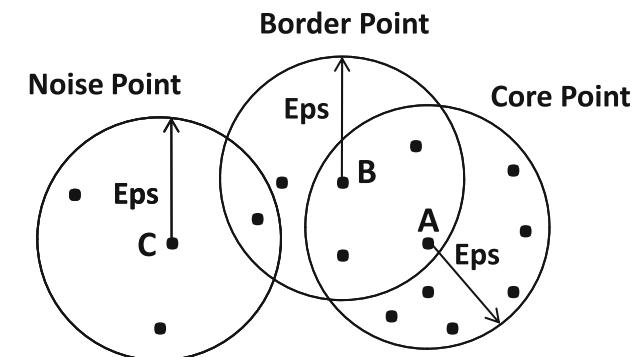
- I parametri sono σ e i vettori \overline{Y}_j e si ottiene il k-means classico
- In generale la forma della mistura $K_1 \cdot e^{-K_2 \cdot \text{Dist}(\overline{X}_i, \overline{Y}_j)}$ consente di usare EM per implementare tutti i clustering basati su prototipi rappresentativi

Clustering per densità

- In questi approcci si cerca di creare i cluster come componenti connesse di un grafo costituito da elementi strutturali che sono caratterizzati dal possedere un parametro di *densità* sopra una certa soglia
- Approcci density based propriamente detti: gli elementi strutturali sono punti il cui intorno contiene un numero di punti maggiore di una certa soglia
- Approcci grid-based: gli elementi strutturali sono ipercubi di una griglia che suddivide lo spazio dei dati: un ipercubo sarà «denso» se contiene un numero di punti maggiore di una certa soglia
 - p intervalli lungo ognuna delle d dimensioni $\rightarrow p^d$ celle della griglia

Clustering per densità

- Algoritmi density based – DBSCAN
 - DBSCAN si basa sulla definizione di tre tipologie di punti nel data set
 - Core points: punti che nel loro intorno di dato raggio Eps contengono almeno τ punti
 - Border points: punti che contengono meno di τ punti nel loro intorno, ma contengono un core point
 - Noise points: punti che nel loro intorno contengono meno di τ punti e non contengono core point



Clustering per densità

- Algoritmi density based – DBSCAN

Algorithm *DBSCAN*(Data: \mathcal{D} , Radius: Eps , Density: τ)

begin

Determine core, border and noise points of \mathcal{D} at level (Eps, τ) ;

Create graph in which core points are connected

if they are within Eps of one another;

Determine connected components in graph;

Assign each border point to connected component
with which it is best connected;

return points in each connected component as a cluster;

end

*Si devono calcolare le
distanze tra tutte le
coppie di punti!!*

Clustering per densità

- Algoritmi density based – DBSCAN
 - Il passo di determinazione del grafo dei core points è un approccio single linkage con distanza Eps
 - La ricerca dei core points è un problema $O(n^2)$ che può essere ridotto a $O(n \log n)$ con l'utilizzo di apposite strutture di indicizzazione
 - *Problemi legati all'elevata dimensionalità poiché tali strutture richiedono calcoli di distanze tra i punti che sono onerose e poco significative per elevati valori di d*

Clustering per densità

- Algoritmi density based – DBSCAN
 - Il fatto che τ (denominata in genere *MinPts* nella letteratura di settore) sia fissato comporta problemi nella gestione di cluster a densità variabile
 - Eps e τ sono in relazione tra loro: Eps può essere determinato come valore di cutoff della distanza di ogni punto dai suoi τ nearest neighbors

Clustering per densità

- Algoritmi grid based

Algorithm *GenericGrid*(Data: \mathcal{D} , Ranges: p , Density: τ)
begin

 Discretize each dimension of data \mathcal{D} into p ranges;

 Determine dense grid cells at density level τ ;

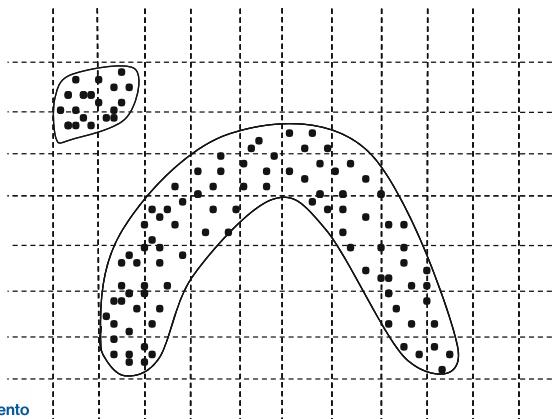
 Create graph in which dense grids are connected if they are adjacent;

 Determine connected components of graph;

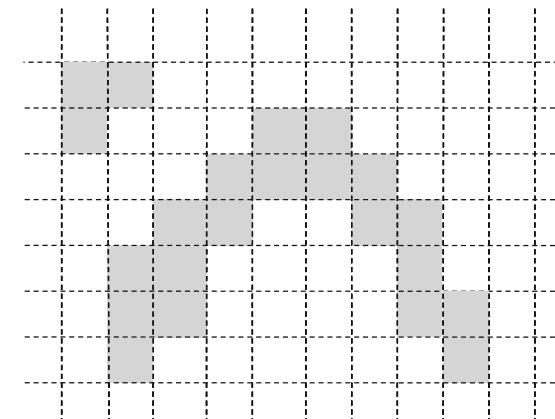
return points in each connected component as a cluster;

end

Le distanze non servono più: si calcolano i confronti con le posizioni discretizzate delle celle!!



(a) Data points and grid



(b) Agglomerating adjacent grids

Clustering per densità

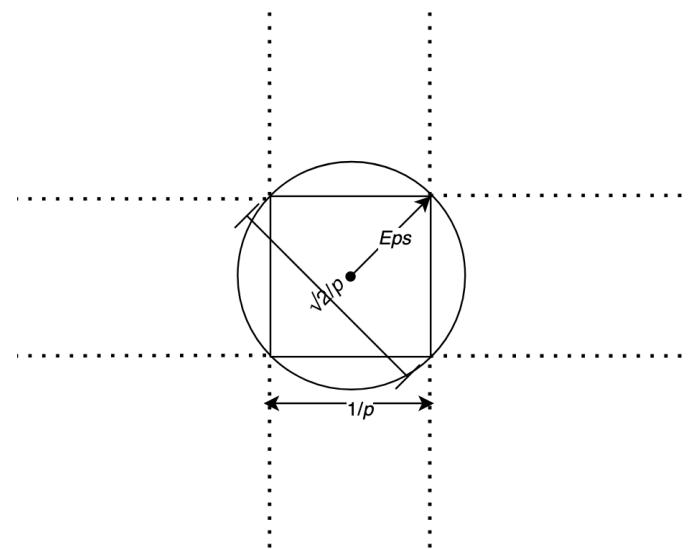
- Algoritmi grid based
 - Il clustering dipende dal numero delle celle p lungo le dimensioni della griglia e dal parametro di densità $MinPts$
 - Celle troppo grandi (p piccolo) possono contenere punti di diversi cluster, mentre celle troppo piccole danno luogo a un numero elevato di celle vuote
 - Un valore di $MinPts$ troppo basso fa collassare più celle in un cluster, mentre al contrario tende a spezzare i cluster
 - Problemi ad elevata dimensionalità perché il numero delle celle cresce esponenzialmente con d

Clustering per densità

- Algoritmi grid based
 - È possibile legare la dimensione della cella a al valore di Eps , a parità di scelta di $MinPts$, per ottenere forme dei cluster simili a quelle di DBSCAN
 - Assumiamo che i dati si trovino all'interno di un ipercubo unitario:

$$\frac{1}{p} \propto \frac{Eps}{\sqrt{2}}$$

Otteniamo un'adiacenza tra celle dense che raggruppa i punti in modo molto simile agli intorni dei core point



Misure di bontà del clustering

- Il clustering è una procedura non supervisionata e quindi non ci fornisce, in principio, informazioni utili alla validazione della bontà del task effettuato
- Si determinano comunque due tipologie di criteri
 - Criteri interni, basati sui dati
 - Criteri esterni, quando delle etichette di classe sono comunque disponibili per verificare se il clustering ha seguito il criterio di classificazione

Misure di bontà del clustering

- Criteri di validazione interni
 - Within Cluster Sum of Squares (WCSS)

$$WCSS = \sum_{j=1}^K \sum_{\overline{X}_i \in \mathcal{C}_j} (\overline{X}_i - \overline{Y}_j)^2$$

- Pensata principalmente per il k-means e per gli algoritmi basati su distanze e cluster di forma sferica
- Tende comunque a diminuire all'aumentare del numero di cluster

Misure di bontà del clustering

- Criteri di validazione interni
 - Rapporto delle distanze Intra-cluster/inter-cluster
 - Campioniamo r punti da \mathcal{D} di cui P appartengono ad un certo cluster e i rimanenti Q no
 - Il clustering è migliore quanto più $Intra/Inter$ è basso

$$Intra = \sum_{(\bar{X}_i, \bar{X}_j) \in P} \text{dist}(\bar{X}_i, \bar{X}_j) / |P|$$

$$Inter = \sum_{(\bar{X}_i, \bar{X}_j) \in Q} \text{dist}(\bar{X}_i, \bar{X}_j) / |Q|$$

Misure di bontà del clustering

- Criteri di validazione interni
 - Silhouette coefficient

$$Davg_i^{in} = \text{avg} (Dist (\overline{X}_k, \overline{X}_l)) , \quad \overline{X}_k, \overline{X}_l \in \mathcal{C}_i$$

$$Dmin_i^{out} = \min_{\mathcal{C}_m} \text{avg} (Dist (\overline{X}_k, \overline{X}_h)) , \quad \overline{X}_k \in \mathcal{C}_i, \overline{X}_h \in \mathcal{C}_m$$

$$S_i = \frac{Dmin_i^{out} - Davg_i^{in}}{\max \{ Dmin_i^{out}, Davg_i^{in} \}}$$

$$S = \frac{1}{n} \sum_i S_i$$

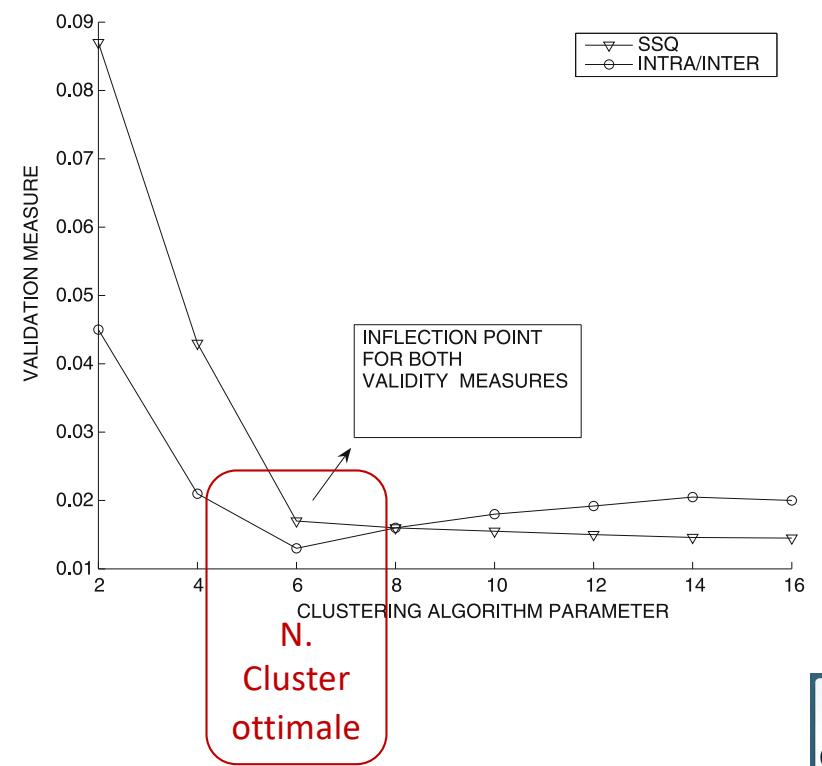
- S varia in $[-1, 1]$: valori grandi positivi indicano buona separazione tra i cluster, mentre valori negativi indicano mescolamento

Misure di bontà del clustering

- Criteri di validazione interni
 - Misure probabilistiche
 - Si assume che i dati siano descritti da una mistura di distribuzioni i cui centri delle componenti sono i centri dei cluster che sono stati trovati dall'algoritmo
 - Si utilizza solo il passo di massimizzazione dell'approccio *EM* calcolando i parametri rimanenti (le covarianze ad es.) tramite massimizzazione della log-likelihood della mistura
 - Il nuovo modello generativo viene usato per predire i dati da clusterizzare
 - La significatività della misura dipende strettamente dalla forma effettiva dei cluster ovvero la misura ha significato se si hanno delle informazioni a priori su come dovrebbe essere il clustering

Misure di bontà del clustering

- Criteri di validazione interni
 - Possono essere utilizzati anche per effettuare il tuning dei parametri e cioè stabilire il numero di cluster
 - Elbow method: sia WCSS sia *Intra/Inter* decrescono rapidamente fino al numero ottimale di cluster e poi si crea un plateau



Misure di bontà del clustering

- Criteri di validazione interni
 - Tutti questi criteri sottendono ciascuno un modello particolare di cluster che potrebbe non essere quello reale
 - Questo rende difficile anche il paragone tra i diversi criteri usati per testare la bontà di un dato algoritmo

Misure di bontà del clustering

- Criteri di validazione esterni
 - Si utilizzano le etichette di cluster, ove disponibili, da usare come etichette di classe
 - Benchmark data set
 - Oppure dati reali con etichette di classe che dipendono dall'applicazione di classificazione e *potrebbero non riflettere* le caratteristiche dei cluster naturali dei dati

Misure di bontà del clustering

- Criteri di validazione esterni

- Siano k_t le etichette di classe e k_d il numero di cluster determinati dall'algoritmo
- Se $k_t = k_d$ si può utilizzare la *matrice di confusione*

Valori predetti

		Valori predetti			
		1	2	3	4
Cluster Indices		1	2	3	4
Valori reali	1	97	0	2	1
	2	5	191	1	3
	3	4	3	87	6
	4	0	0	5	195

		Valori predetti			
		1	2	3	4
Cluster Indices		1	2	3	4
	1	33	30	17	20
	2	51	101	24	24
	3	24	23	31	22
	4	46	40	44	70

Misure di bontà del clustering

- Criteri di validazione esterni
 - In generale si usano degli indici numerici
 - Sia m_{ij} il numero di punti della classe i mappati nel cluster j , cioè l'elemento (i,j) della matrice di confusione

$$N_i = \sum_{j=1}^{k_d} m_{ij}, \quad \forall i = 1 \dots k_t$$

Somma dei valori lungo una riga:
l'effettivo numero di punti
appartenenti alla classe i

$$M_j = \sum_{i=1}^{k_t} m_{ij}, \quad \forall j = 1 \dots k_d$$

Somma dei valori lungo una colonna:
il numero di punti predetti
come appartenenti al cluster j

Misure di bontà del clustering

- Criteri di validazione esterni

- Purity: si assume che un buon cluster avrà quanti più possibili punti appartenenti tutti ad una stessa classe

$$\text{Purity} = \frac{\sum_{j=1}^{k_d} P_j}{\sum_{j=1}^{k_d} M_j}, \quad P_j = \max_i m_{ij} \quad \begin{matrix} \text{Numero di punti nella} \\ \text{classe dominante} \end{matrix}$$

- La Purity tende a 1 per un buon clustering

Misure di bontà del clustering

- Criteri di validazione esterni
 - Gini index: considera il contributo anche delle altre classi presenti nel cluster
 - È un buon cluster quello in cui i punti sono concentrati in poche classi e c'è poca dispersione
 - G_j tende a 0 per un buon clustering, mentre ha $1 - 1/k_t$ come limite superiore

$$G_j = 1 - \sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j} \right)^2$$

$$G_{average} = \frac{\sum_{j=1}^{k_d} G_j \cdot M_j}{\sum_{j=1}^{k_d} M_j}$$

% punti
del cluster
 j predetti
in classe i

Misure di bontà del clustering

- Criteri di validazione esterni
 - Entropia: l'indice di Gini G_j è legato all'entropia del cluster j E_j poiché m_{ij}/M_j è la probabilità frequentista che un punto appartenga a tale cluster

$$E_j = - \sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j} \right) \cdot \log \left(\frac{m_{ij}}{M_j} \right)$$

$$E_{\text{average}} = \frac{\sum_{j=1}^{k_d} E_j \cdot M_j}{\sum_{j=1}^{k_d} M_j}$$

Misure di bontà del clustering

- Criteri di validazione esterni
 - È possibile infine stabilire una misura di bontà del clustering analizzando la matrice di confusione con le misure tipiche della classificazione: Precision, Recall e F_1 -score (che vedremo più avanti)