

Facoltà di Scienze MM. FF. NN.

Università di Verona

A.A. 2010-11

# **Teoria e Tecniche del Riconoscimento**

## **Clustering**

# Sommario

- ⇒ Tassonomia degli algoritmi di clustering
- ⇒ Algoritmi partizionali:
  - ⇒ clustering sequenziale
  - ⇒ center-based clustering: K-means e varianti
  - ⇒ model based clustering: Mixture of Gaussians (EM)
- ⇒ Algoritmi gerarchici agglomerativi: complete link, single link
- ⇒ (Altri: Fuzzy clustering, Neural Networks clustering, ...)
  
- ⇒ LABORATORIO: implementazione di alcuni semplici algoritmi di clustering

# Tassonomia

## Nota preliminare

- ⇒ Esistono moltissimi algoritmi di clustering
- ⇒ Non esiste un'unica tassonomia
  - ⇒ esistono diverse suddivisioni
- ⇒ In questo corso si adotta il punto di vista di Jain
  - ⇒ Jain, Dubes, Algorithms for clustering data, 1988
  - ⇒ Jain et al., Data Clustering: a review, ACM Computing Surveys, 1999

# Classi di approcci

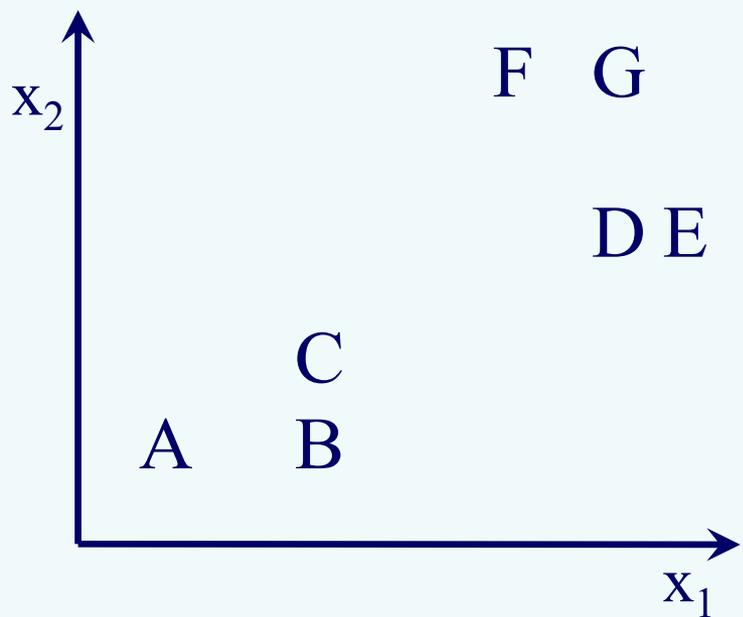
A seconda del punto di vista possiamo avere differenti classi:

- ⇒ Gerarchico vs partizionale
- ⇒ Hard clustering vs soft clustering
- ⇒ Agglomerativo vs divisivo
- ⇒ Seriale (sequenziale) vs simultaneo
- ⇒ Monothetic vs polythetic
- ⇒ Graph Theory vs matrix algebra
- ⇒ Incrementale vs non incrementale
- ⇒ Deterministico vs stocastico

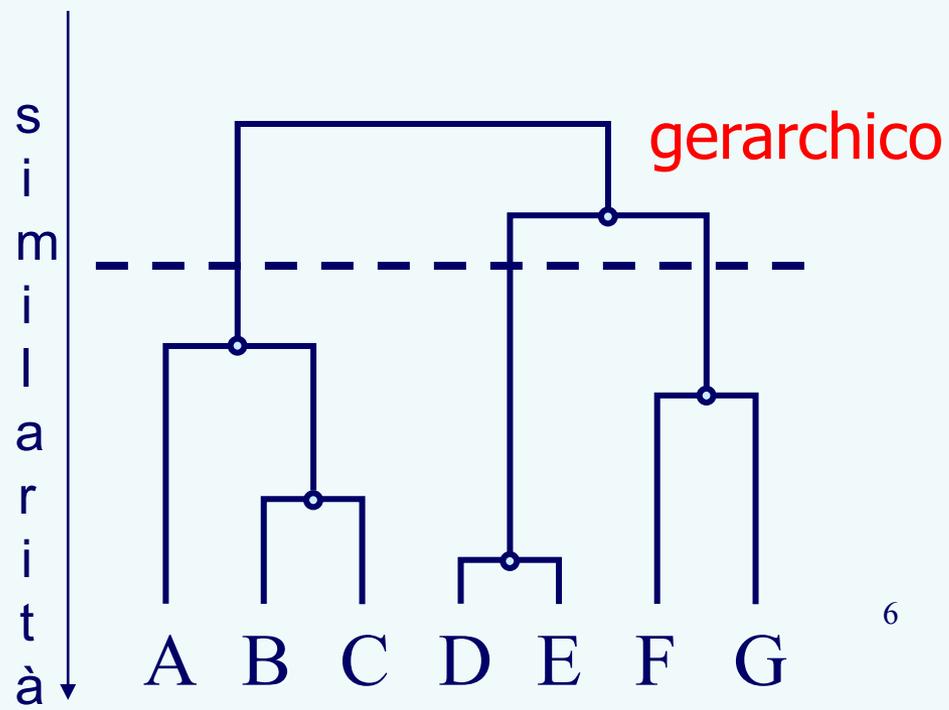
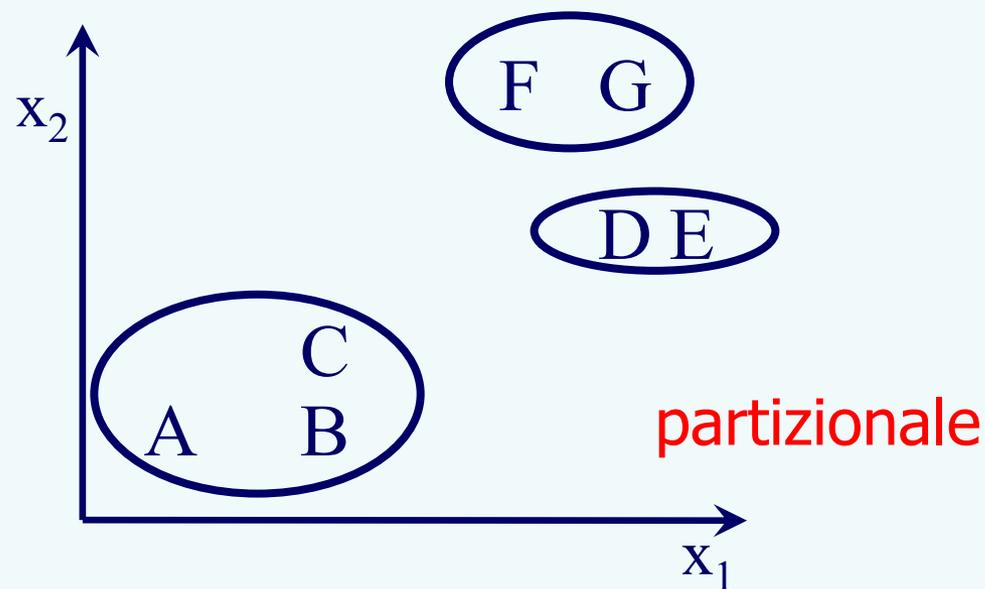
# Gerarchico vs partizionale

PUNTO DI VISTA: il tipo di risultato dell'operazione di clustering

- ⇒ Clustering Partizionale: il risultato è una singola partizione dei dati (tipicamente il numero di cluster deve essere dato a priori)
  - ⇒ mira ad identificare i gruppi naturali presenti nel dataset
  - ⇒ tipicamente richiede che i dati siano rappresentati in forma vettoriale
  - ⇒ genera una partizione (insieme di cluster disgiunti la cui unione ritorna il data set originale)
- ⇒ Clustering Gerarchico: il risultato è una serie di partizioni innestate (un "dendrogramma")
  - ⇒ mira ad evidenziare le relazioni tra i vari pattern del dataset
  - ⇒ tipicamente richiede una matrice di prossimità



problema originale



# Gerarchico vs partizionale

## Ulteriori dettagli

### ⇒ Partizionale:

- ⇒ ottimo per dataset grandi
- ⇒ scegliere il numero di cluster è un problema (esistono metodi per determinare in modo automatico il numero di cluster)
- ⇒ tipicamente il clustering è il risultato di un procedimento di ottimizzazione, definito sia localmente (in un sottoinsieme dei pattern) che globalmente (in tutti i pattern)
- ⇒ Esempi: K-means (e sue varianti), PAM, ISODATA,...

### ⇒ Gerarchico

- ⇒ non è necessario settare a priori il numero di cluster
- ⇒ più informativo del partizionale, è improponibile per dataset grandi
- ⇒ Esempi: Complete Link, Single Link, Ward Link,...

# Hard clustering vs soft clustering

PUNTO DI VISTA: la natura dei cluster risultanti

⇒ Hard clustering:

- ⇒ un pattern viene assegnato ad un unico cluster
  - ⇒ sia durante l'esecuzione dell'algoritmo che nel risultato
- ⇒ detto anche clustering "esclusivo"

⇒ Soft clustering:

- ⇒ un pattern può essere assegnato a diversi clusters
- ⇒ detto anche "fuzzy clustering" o "clustering non esclusivo"
- ⇒ ci può essere una funzione di "membership"
- ⇒ può essere trasformato in hard guardando la massima membership

⇒ ESEMPIO:

- ⇒ raggruppare persone per età è esclusivo
- ⇒ raggrupparle per malattia è non esclusivo

# Agglomerativo vs divisivo

PUNTO DI VISTA: come vengono formati i cluster

⇒ Agglomerativo:

- ⇒ costruisce i cluster effettuando operazioni di "merge"
- ⇒ inizia con un cluster per ogni pattern, e successivamente fonde cluster assieme fino al raggiungimento di una determinata condizione

⇒ Divisivo:

- ⇒ costruisce i cluster effettuando operazioni di "split"
- ⇒ inizia con un unico cluster contenente tutti i dati, e successivamente divide i cluster fino al raggiungimento di una determinata condizione

⇒ Commenti:

- ⇒ tipo di procedura piuttosto che tipo di clustering
- ⇒ è naturalmente applicabile al clustering gerarchico, in linea di principio funziona anche per il clustering partizionale

# Sequenziale vs simultaneo

PUNTO DI VISTA: in che modo vengono processati i pattern

⇒ Sequenziale: i pattern vengono processati uno alla volta

⇒ Simultaneo: i pattern vengono processati tutti assieme

⇒ ESEMPIO sequenziale: prende un pattern alla volta e lo assegna ad un cluster

# Monothetic vs polythetic

PUNTO DI VISTA: come vengono utilizzate le features

- ⇒ Monothetic: viene utilizzata una feature alla volta per fare clustering
- ⇒ Polythetic: vengono utilizzate tutte le features simultaneamente per fare clustering
- ⇒ La maggior parte delle tecniche di clustering sono polythetic.

# Monothetic vs polythetic

⇒ Esempio di algoritmo monothetic [Anderberg 73]

⇒ si divide il data set in due clusters utilizzando una sola feature

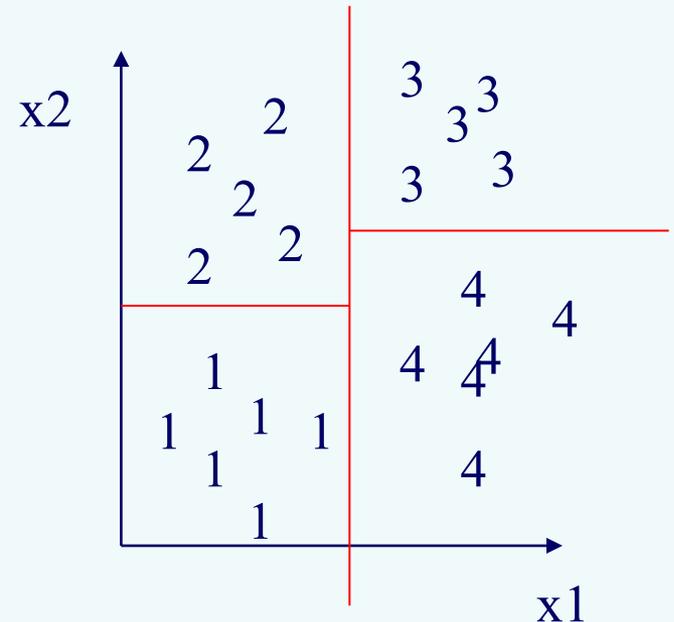
⇒ ognuno di questi data sets viene poi diviso in due utilizzando la seconda feature

⇒ si procede così fino alla fine

⇒ Svantaggi:

⇒  $d$  features  $2^d$  clusters

⇒  $d$  grande, troppo frammentato



# Graph Theory vs matrix algebra

PUNTO DI VISTA: come viene formulato matematicamente l'algoritmo

- ⇒ graph theory: gli algoritmi sono formulati in termini di teoria dei grafi, con l'utilizzo di definizione e proprietà dei grafi (ad esempio connettività)
- ⇒ matrix algebra: gli algoritmi sono espressi in termini di formule algebriche (ad esempio l'errore quadratico medio)

# Incrementale vs non incrementale

PUNTO DI VISTA: cosa succede se arrivano nuovi dati

- ⇒ Incrementale: il clustering può essere “aggiornato” (è costruito in modo incrementale)
- ⇒ Non incrementale: all’arrivo di nuovi dati occorre riesaminare l’intero data set
- ⇒ Caratteristica cruciale in questi anni: database sempre più grossi e sempre in espansione!

# Deterministico vs stocastico

PUNTO DI VISTA: come viene ottimizzata la funzione di errore

- ⇒ deterministico: ottimizzazione classica (discesa lungo il gradiente)
- ⇒ stocastico: ricerca stocastica nello spazio degli stati della soluzione (Monte Carlo)
- ⇒ Tipico problema nel clustering partizionale che deve ottimizzare una funzione di errore (come lo scarto quadratico medio)

# Clustering partizionale

## ⇒ Classi di approcci:

### ⇒ clustering sequenziale:

- ⇒ approccio di clustering molto semplice e intuitivo
- ⇒ tipicamente i pattern vengono processati poche volte
- ⇒ in generale, il risultato finale dipende dall'ordine con cui vengono presentati i pattern
- ⇒ funzionano bene per cluster convessi

### ⇒ center-based clustering:

- ⇒ ogni cluster è rappresentato da un centro
- ⇒ metodi efficienti per clusterizzare database grandi
- ⇒ l'obiettivo è minimizzare una funzione di costo
- ⇒ funzionano bene per cluster convessi

# Clustering partizionale

⇒ search based clustering

⇒ l'idea è quella di minimizzare la funzione di costo in modo "globale"

⇒ model based clustering

⇒ l'idea è quella di creare dei modelli per i dati (tipicamente probabilistici)

⇒ tipicamente si assume che i dati siano generati da una mistura di distribuzioni di probabilità in cui ogni componente identifica un cluster

# Clustering sequenziale

BSAS: Basic Sequential Algorithmic Scheme

⇒ algoritmo di clustering sequenziale facile e intuitivo

Assunzioni/Idee

⇒ i pattern vengono processati una volta sola, in ordine

⇒ ogni pattern processato viene assegnato ad un cluster esistente oppure va a creare un nuovo cluster

⇒ il numero di cluster non è conosciuto a priori ma viene stimato durante il processo

# BSAS: algoritmo

Notazione/parametri:

- ⇒  $\mathbf{x}_i$ : vettore di punti,  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  dataset da clusterizzare
- ⇒  $C_j$ : j-esimo cluster
- ⇒  $d(\mathbf{x}, C)$ : distanza tra un punto e un insieme (un cluster)  
(simile alla distanza tra insiemi)
  - ⇒ Max: distanza massima
  - ⇒ Min: distanza minima
  - ⇒ Average: distanza media
  - ⇒ center-based: distanza dal "rappresentante"
- ⇒  $\Theta$ : soglia di dissimilarità
- ⇒  $m$ : numero di cluster trovati ad un determinato istante

# BSAS: algoritmo

Algoritmo:

```
m=1
 $C_m = x_1$ 
for i = 2 to N
    trova  $C_k$  tale che  $d(x_i, C_k) = \min_{1 \leq j \leq m} d(x_i, C_j)$ 
    if  $d(x_i, C_k) > \theta$ 
        m = m+1
         $C_m = \{x_i\}$ 
    else
         $C_k = C_k \cup \{x_i\}$ 
        (se necessario aggiornare i rappresentanti)
    end if
end for
```

# BSAS: algoritmo

⇒ Se la distanza  $d(\mathbf{x}, C) = d(\mathbf{x}, \mathbf{m}_C)$  (distanza dalla media del cluster), allora l'aggiornamento dei rappresentanti può essere fatto on-line

⇒ Notazioni

⇒  $m_{C_k}$  è la media del cluster  $k$

⇒  $x$  è il punto aggiunto al cluster  $C_k$

⇒  $n_{C_k}$  è la cardinalità del cluster  $C_k$

$$m_{C_k}^{new} = \frac{(n_{C_k}^{new} - 1)m_{C_k}^{old} + x}{n_{C_k}^{new}}$$

# Clustering sequenziale

Commenti su BSAS:

- ⇒ si può osservare che l'ordine con cui vengono processati i pattern è cruciale
  - ⇒ ordini diversi possono produrre risultati diversi
- ⇒ la scelta della soglia  $\theta$  è cruciale
  - ⇒  $\theta$  troppo piccola, vengono determinati troppi cluster
  - ⇒  $\theta$  troppo grande, troppo pochi cluster
- ⇒ si può scambiare la dissimilarità con la similarità (cambiando min con max e  $>$  con  $<$ )
- ⇒ con i rappresentanti (con le medie) i cluster che escono sono compatti

# Clustering sequenziale

⇒ Metodo per calcolare il numero ottimale di clusters:

⇒ for  $\theta = a$  to  $b$  step  $c$

⇒ Eseguiere  $s$  volte l'algoritmo BSAS, ogni volta processando i pattern con un ordine differente

⇒ stimare  $m_\theta$  come il numero più frequente di cluster

⇒ end for

⇒ visualizzare il numero di cluster  $m_\theta$  vs il parametro  $\theta$

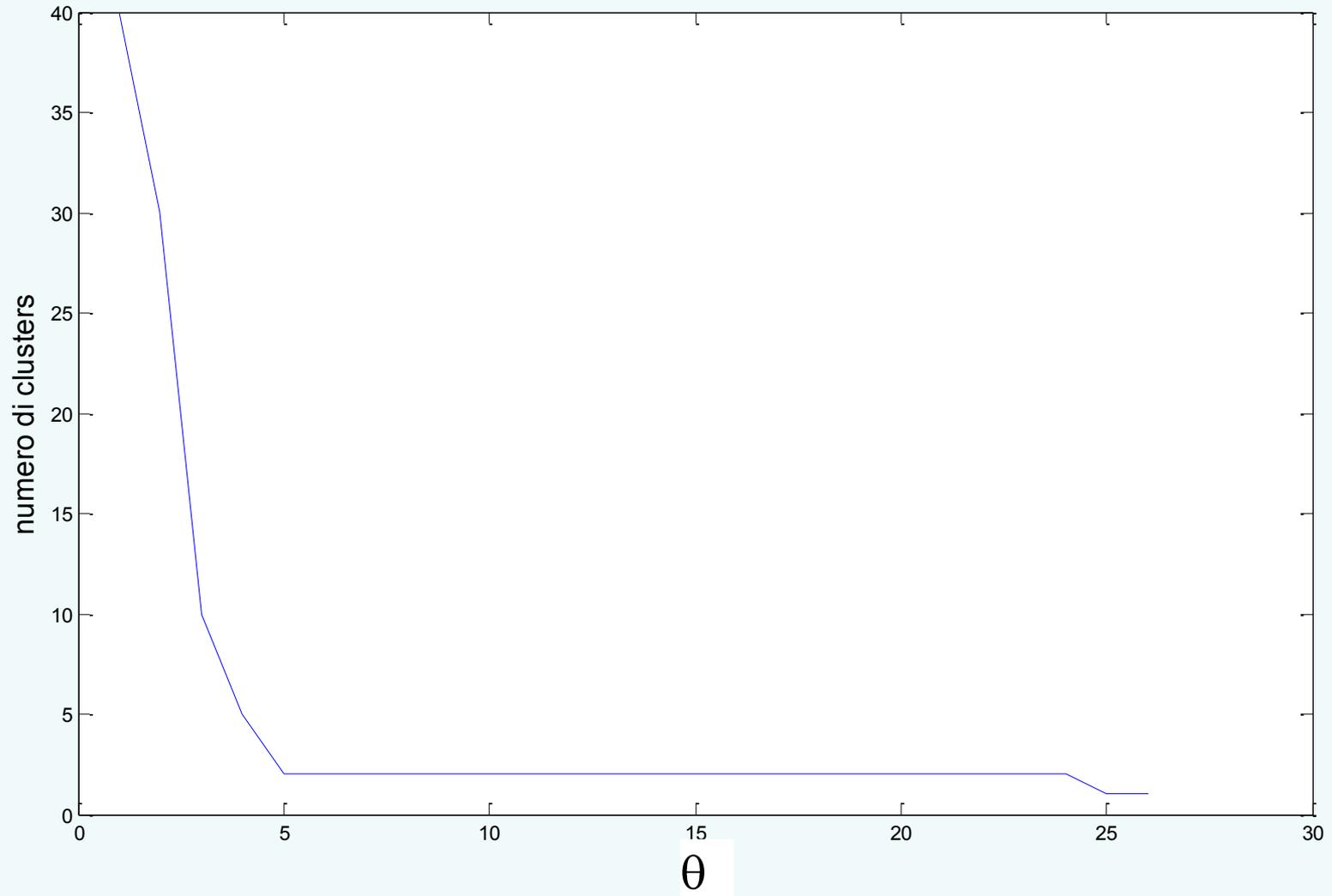
⇒ il numero di cluster ottimale è quello della regione "piatta" più lunga

⇒ dettagli

⇒  $a$  è la distanza minima tra i punti,  $b$  la distanza massima

⇒ assumiamo che "esista" un clustering

# Clustering sequenziale



# Center-based clustering

## K-means

⇒ Algoritmo più famoso di clustering partizionale

⇒ **IDEE:**

⇒ minimizza una funzione di errore

⇒ ogni cluster è rappresentato dalla sua media

⇒ si parte da una clusterizzazione iniziale, ed ad ogni iterazione si assegna ogni pattern alla media più vicina

⇒ si riaggiornano le medie

⇒ si continua fino a convergenza

# Center-based clustering

## ⇒ Commenti

- ⇒ il numero di cluster deve essere fissato a priori
- ⇒ l'ottimizzazione spesso porta ad un ottimo "locale"
  - ⇒ l'inizializzazione è cruciale: una cattiva inizializzazione porta ad un clustering pessimo
- ⇒ è molto efficiente nel clusterizzare dataset grandi, perché la sua complessità computazionale è linearmente dipendente dalla dimensione del data set
- ⇒ i cluster ottenuti hanno una forma convessa
- ⇒ lavora solo su dati vettoriali numerici (deve calcolare la media)
- ⇒ tipicamente viene utilizzata la distanza euclidea

# Center based clustering

## Varianti del K-means

- ⇒ cercare di migliorare l'inizializzazione ([Anderberg 1973])
- ⇒ ISODATA (*Iterative Self-Organizing Data Analysis Techniques*)
  - ⇒ permettere lo splitting e il merging dei cluster risultanti
  - ⇒ Ad ogni iterazione effettua dei controlli sui cluster risultanti:
    - ⇒ un cluster viene diviso se la sua varianza è sopra una soglia prefissata, oppure se ha troppi punti
    - ⇒ due cluster vengono uniti se la distanza tra i due relativi centroidi è minore di un'altra soglia prefissata, oppure se hanno troppo pochi punti
  - ⇒ la scelta delle soglie è cruciale, ma fornisce anche una soluzione alla scelta del numero di cluster

# Center based clustering

## Varianti del K-means

- ⇒ utilizzo della distanza di Mahalanobis come distanza per i punti ([Mao Jain 1996])
  - ⇒ vantaggio: posso anche trovare cluster ellissoidali
  - ⇒ svantaggio: devo calcolare ogni volta la matrice di covarianza
- ⇒ PAM (Partitioning around the medoids)
  - ⇒ l'idea è quella di utilizzare come "centri" del K-means i medoidi (o i punti più centrali) invece che le medie
    - ⇒ non introduco nuovi elementi nel dataset
    - ⇒ più robusto agli outliers
    - ⇒ posso lavorare anche con dati non vettoriali (data una funzione di distanza tra questi dati)

# Model-based clustering

## ⇒ IDEE:

- ⇒ utilizzare un insieme di modelli per i cluster
- ⇒ l'obiettivo diventa quello di massimizzare il fit tra i modelli e i dati
- ⇒ si assume che i dati siano generati da una mistura di funzioni di probabilità differenti, ognuna delle quali rappresenta un cluster
- ⇒ ovviamente il metodo di clustering funziona bene se i dati sono conformi al modello

## ⇒ Due approcci al model based clustering

- ⇒ classification likelihood approach
- ⇒ mixture likelihood approach

# Model-based clustering

## ⇒ Commenti:

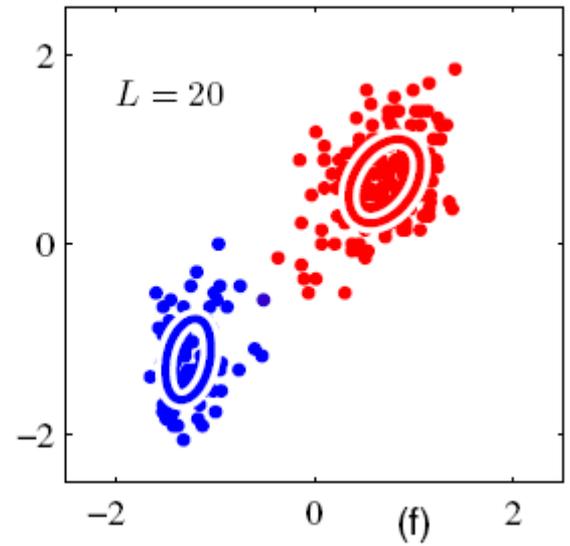
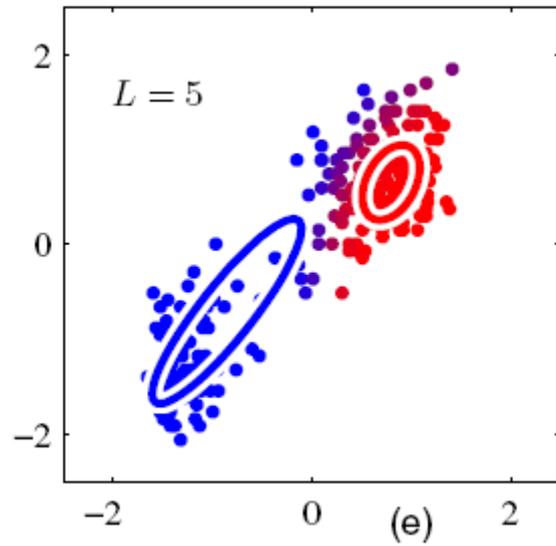
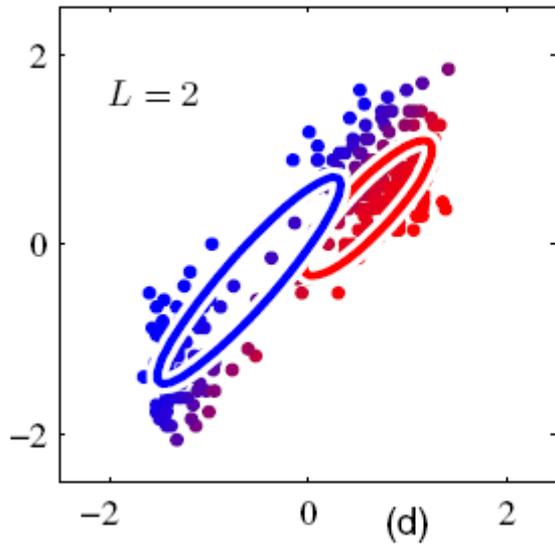
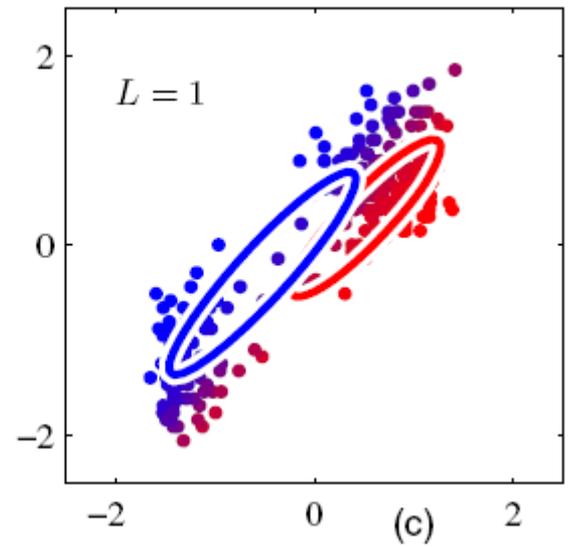
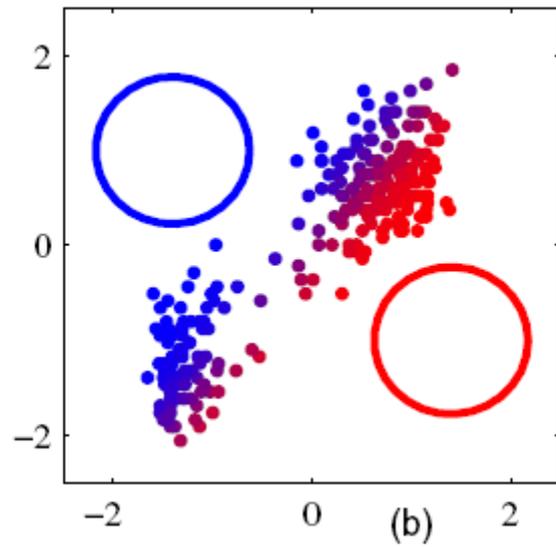
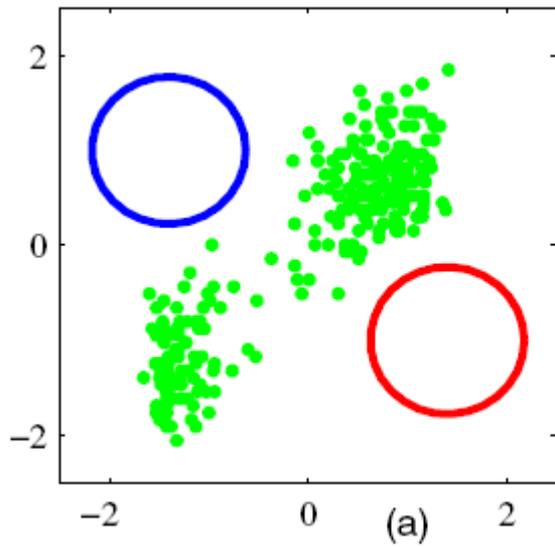
- ⇒ mixture likelihood approach: l'algoritmo più utilizzato è l'EM
- ⇒ esistono molti risultati sulla determinazione del numero ottimale di clusters
- ⇒ esistono alcuni approcci anche per inizializzare l'EM
  
- ⇒ classification likelihood approach: è equivalente al K-means assunto che:
  - ⇒ matrice di covarianza uguale per tutti i cluster
  - ⇒ matrice di covarianza proporzionale all'identità

# Model based clustering

## Gaussian Mixture Models (GMM) Clustering

- ⇒ tecnica di soft clustering molto utilizzata (mixture likelihood approach)
- ⇒ la mistura è composta da Gaussiane
- ⇒ il modello è stimato utilizzando Expectation-Maximization (EM)

# Esempio



# Model based clustering

Commenti:

⇒ molto utilizzato in svariati contesti

⇒ l'inizializzazione è un problema

⇒ Numero di cluster: il problema può essere visto come un problema di model selection:

⇒ qual'è la miglior dimensione del modello dati i dati?

⇒ vedi slides su HMM (equivalente alla determinazione del miglior numero di stati di un HMM)

# Clustering gerarchico

- ⇒ Algoritmi di clustering che generano una serie di partizioni innestate
- ⇒ Rappresentazione di un clustering gerarchico: il dendrogramma

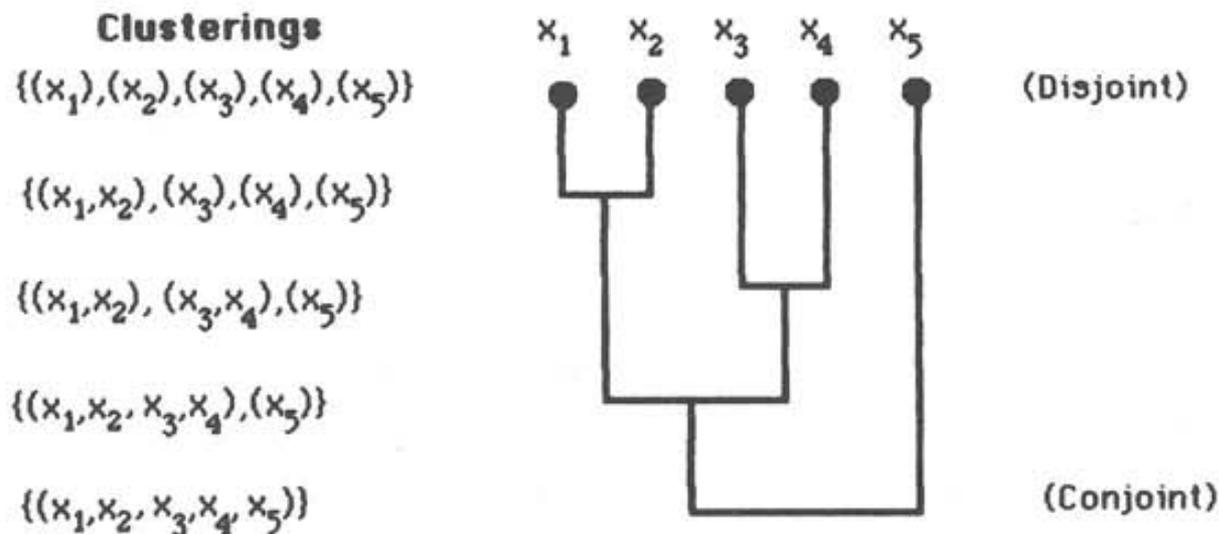


Figure 3.2 Example of dendrogram.

# Clustering gerarchico

⇒ Clustering gerarchico agglomerativo:

⇒ si parte da una partizione in cui ogni cluster contiene un solo elemento

⇒ si continua a fondere i cluster più "simili" fino ad avere un solo cluster

⇒ definizioni diverse del concetto di "cluster più simili" genera algoritmi diversi

⇒ Approcci più utilizzati:

⇒ single link

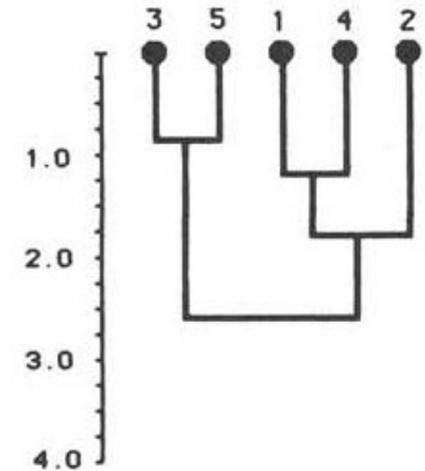
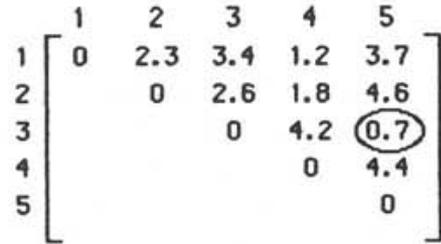
⇒ complete link

⇒ formulazione con le matrici (alla lavagna)

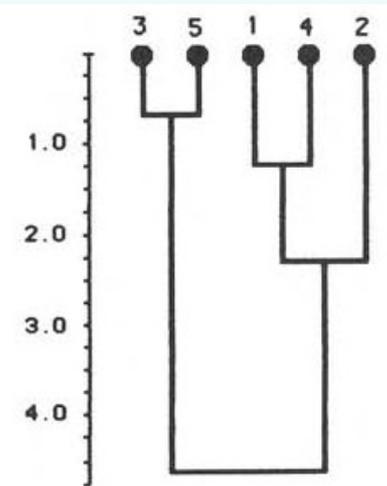
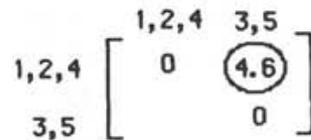
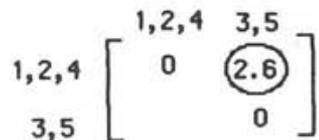
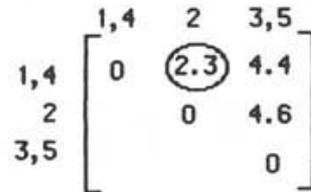
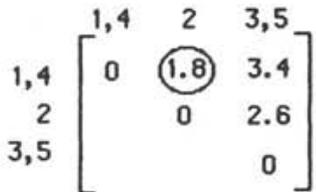
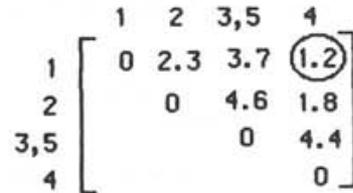
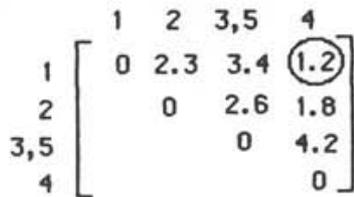
# Clustering gerarchico

SL  $d[(k), (r, s)] = \min \{d[(k), (r)], d[(k), (s)]\}$

CL  $d[(k), (r, s)] = \max \{d[(k), (r)], d[(k), (s)]\}$



Single Link



Complete Link

single link

complete link

# Clustering gerarchico

Single link / Complete link:

formulazione con i grafi (cenni)

⇒ Definizione "threshold graph"  $G(v)$

⇒ grafo in cui i nodi sono tutti gli elementi

⇒ esiste un arco tra  $i$  e  $j$  se  $d(i,j) \leq v$

ESEMPIO:  $G(5)$

$$\mathcal{D}_1 = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix} & \begin{bmatrix} 0 & 6 & 8 & 2 & 7 \\ 6 & 0 & 1 & 5 & 3 \\ 8 & 1 & 0 & 10 & 9 \\ 2 & 5 & 10 & 0 & 4 \\ 7 & 3 & 9 & 4 & 0 \end{bmatrix} \end{matrix}$$

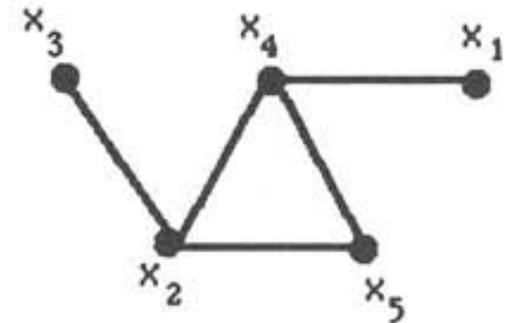
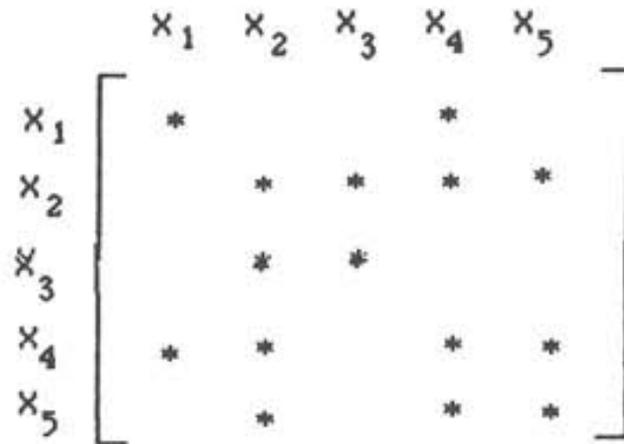


Figure 3.3 Binary relation and threshold graph for threshold 5.

# Clustering gerarchico

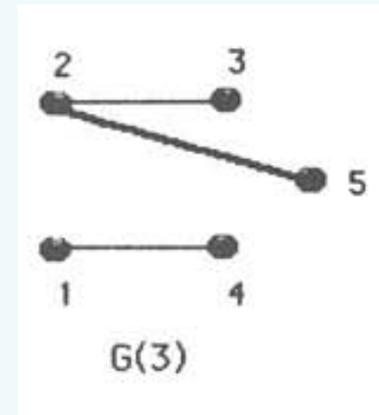
⇒ Si calcolano tutti i possibili grafi  $G(v)$

⇒ Assunzioni: no ties

⇒  $v$  varia tra gli interi (scala ordinale)

⇒ Single Link:

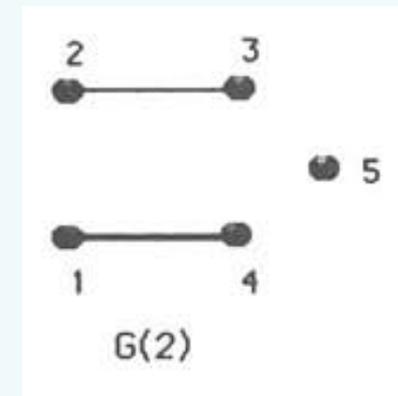
⇒ i cluster sono le componenti connesse  
(sottogradi massimamente connessi)



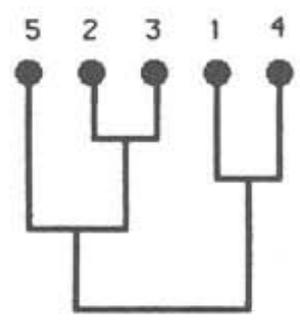
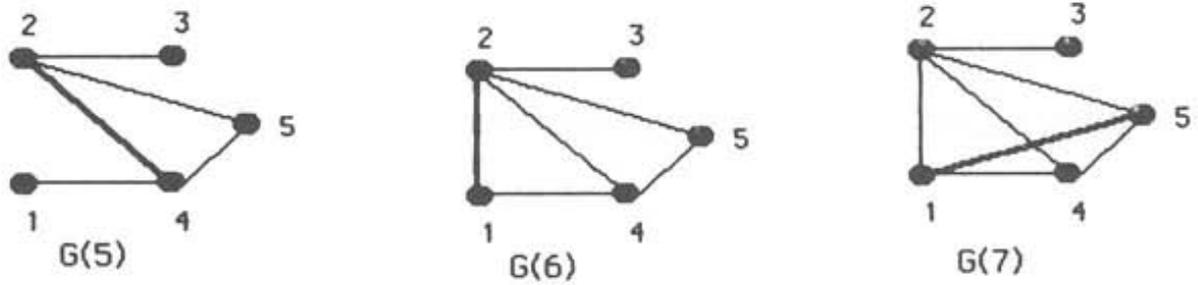
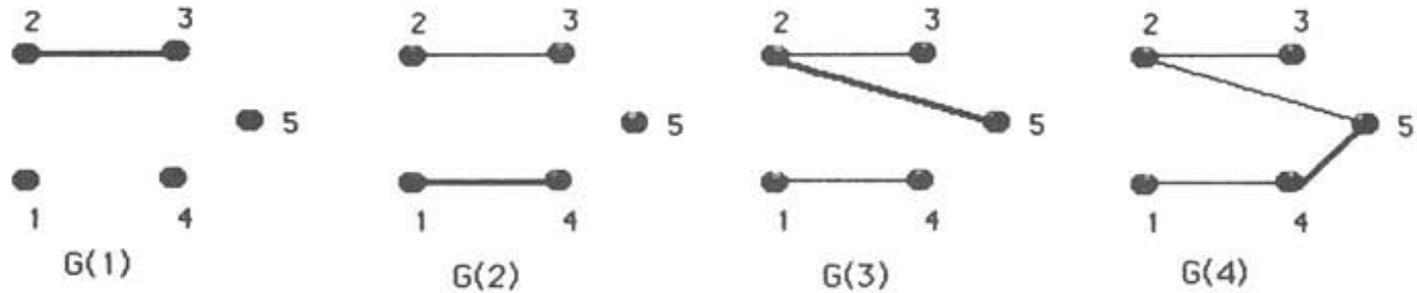
⇒ Complete Link

⇒ i cluster sono clique tra cluster  
precedentemente determinati

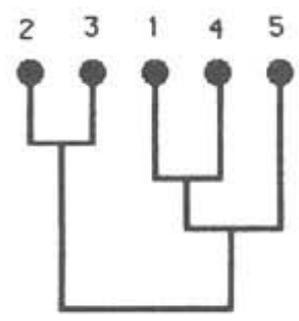
⇒ clique: sottografo massimamente completo –  
tutti con tutti



# Clustering gerarchico



Single Link



Complete Link

# Clustering gerarchico

## Commenti:

- ⇒ Single link unisce due cluster se esiste un solo edge
  - ⇒ tende a formare cluster allungati
- ⇒ Complete link unisce due cluster se tutti gli elementi sono connessi
  - ⇒ più conservativo, tende a formare cluster convessi
- ⇒ In generale è stato dimostrato che Complete Link funziona meglio

# Clustering gerarchico

## Altri criteri di unione dei cluster

⇒ UPGMA (Unweighted pair group method using arithmetic averages)

⇒ la distanza tra cluster è definita come la media delle distanze di tutte le possibili coppie formate da un punto del primo e un punto del secondo

⇒ utilizzato nel periodo iniziale della filogenesi

⇒ Metodo di Ward

⇒ fonde assieme i cluster che portano alla minima perdita di informazione

⇒ informazione intesa in termini di varianza

# Altre metodologie

## ⇒ Fuzzy clustering:

- ⇒ Tecniche di clustering che si basano sulla teoria dei Fuzzy Sets (Zadeh 1965)
- ⇒ Nella classica teoria degli insiemi, l'appartenza di un punto ad un insieme è una variabile binaria (o appartiene o non appartiene)
- ⇒ La teoria degli insiemi fuzzy permette che un elemento appartenga a più di un insieme contemporaneamente
- ⇒ questo è descritto tramite una funzione di membership
  - ⇒ a valori nell'intervallo  $[0, 1]$ .

## ⇒ Logica Fuzzy:

- ⇒ logica dove le variabili non hanno un valore binario ma un valore nell'intervallo  $[0,1]$ 
  - ⇒ 0.8 -> quasi vero

# Altre metodologie

Differenza tra probabilità e funzione di membership

⇒ Sottile e controversa

⇒ Più accettata:

⇒ probabilità: approccio frequentista

⇒ un oggetto appartiene ad una sola classe (e.g. testa o croce)

⇒ la probabilità misura quanto spesso l'oggetto appartiene ad una classe

⇒ approccio basato sulla misura (ripetizioni)

⇒ funzione di membership

⇒ un oggetto può appartenere a più classi

⇒ più appropriato per concetti sfumati e soggettivi (esempio il concetto di caldo, freddo, alto, basso)

# Altre metodologie

## ⇒ Clustering con logica fuzzy

⇒ re-implementazione di diverse tecniche di clustering con la logica fuzzy

⇒ Esempio: fuzzy K-means

⇒ Minimizza la funzione

$$J_q(U, V) = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^q d^2(\mathbf{x}_i, \mathbf{V}_j)$$

⇒ u funzione di membership

⇒ q controlla la "fuzziness" del clustering risultante

# Altri metodi

Funzionamento:

Partendo da una configurazione iniziale, ripetere fino a convergenza

- ⇒ **Processo competitivo:** dato un vettore in ingresso, viene calcolato il neurone più "simile" (neurone vincente)
  - ⇒ il neurone che si "attiva" di più
  - ⇒ il neurone il cui vettore di pesi è il più simile al vettore in ingresso
- ⇒ **Processo adattivo**
  - ⇒ viene modificato il vettore dei pesi del neurone vincente e di tutto il suo vicinato (tipicamente definito con una gaussiana)
  - ⇒ viene modificato in modo da assomigliare al vettore di ingresso