Riconoscimento e recupero dell’informazione per bioinformatica

Clustering: metodologie

Manuele Bicego

Corso di Laurea in Bioinformatica
Dipartimento di Informatica - Università di Verona

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, ...)

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:

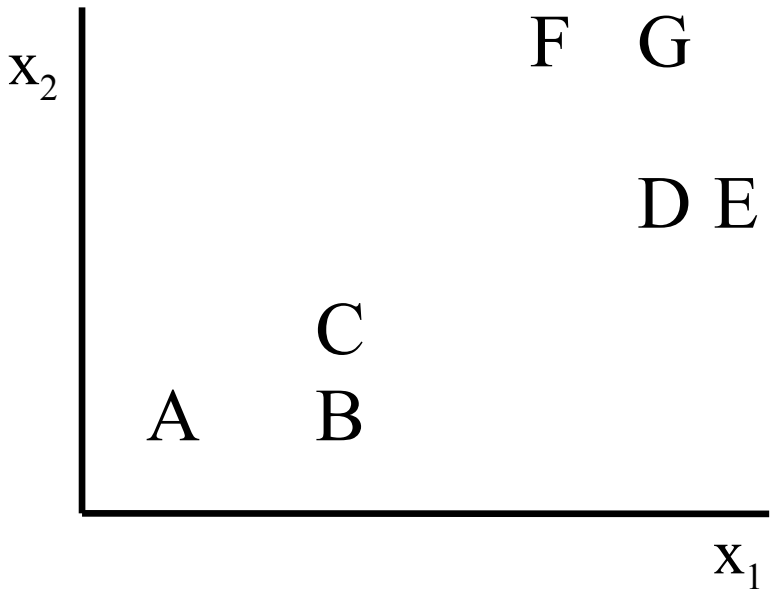
Principali punti di vista

- ⇨ Gerarchico vs partizionale
- ⇨ Hard clustering vs soft clustering
- ⇨ Agglomerativo vs divisivo
- ⇨ Sequenziale vs simultaneo
- ⇨ Incrementale vs non incrementale

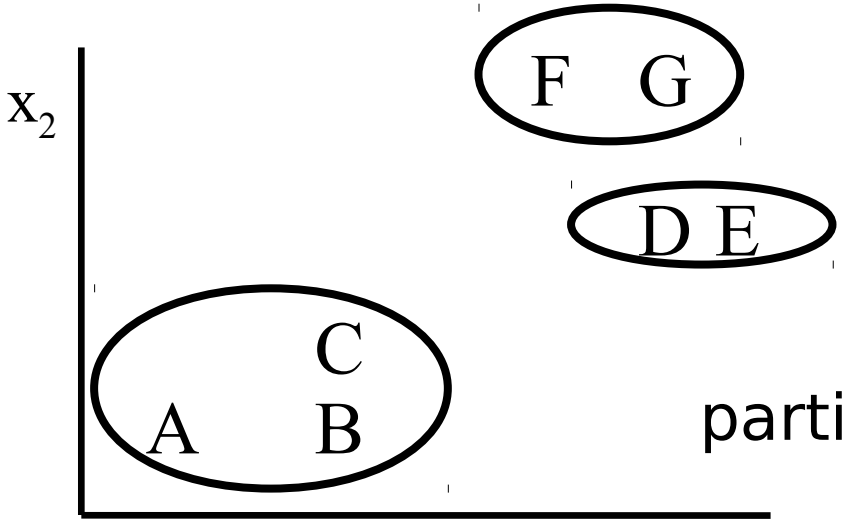
Gerarchico vs partizionale

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

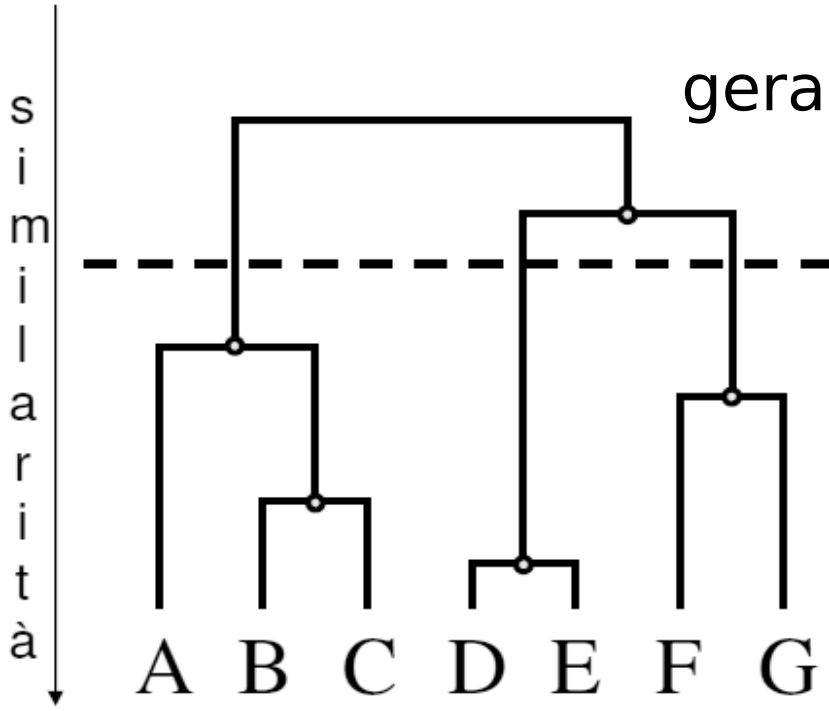


problema originale



partizionale

gerarchico



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 (su un sottoinsieme dei pattern) che globalmente (su 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

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

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!

Il clustering partizionale

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

- 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, $\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$ 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”
- Θ: soglia di dissimilarità
- m: numero di cluster trovati ad un determinato istante

BSAS: algoritmo

Algoritmo:

m=1
C_m = $\mathbf{x}_1$
for i = 2 to N
    trova C_k tale che $d(\mathbf{x}_i, C_k) = min_{1 \leq j \leq m} d(\mathbf{x}_i, C_j)$
    if $d(\mathbf{x}_i, C_k)$ > θ
        m = m+1
        C_m = $\{\mathbf{x}_i\}$
    else
        $C_k = C_k \cup \{\mathbf{x}_i\}$
        (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_{Ck} è la media del cluster k
  - x è il punto aggiunto al cluster C_k
  - n_{Ck} 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

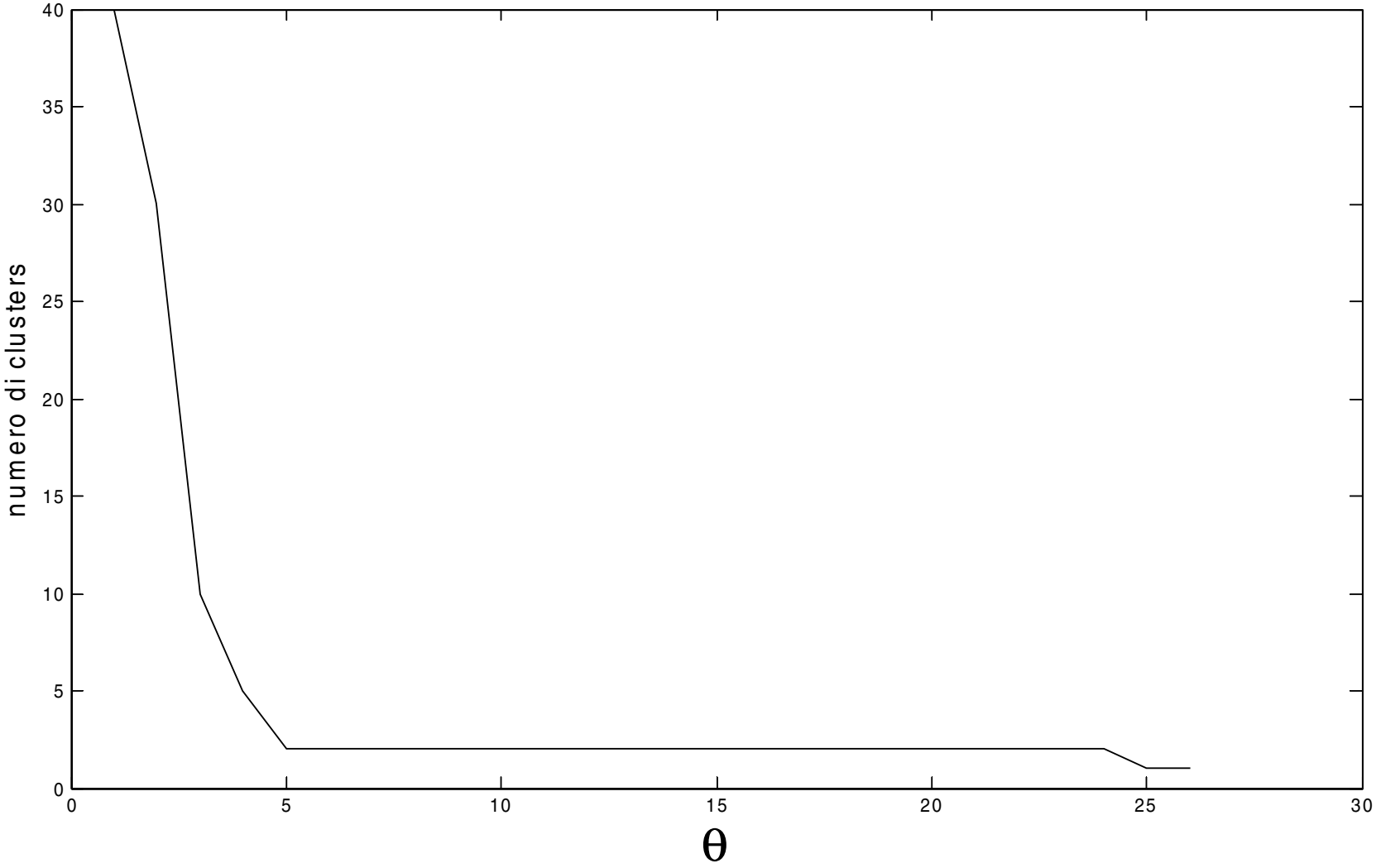
Commenti su BSAS:

- si può osservare che l'ordine con cui vengono processati i pattern è cruciale
  - ordini diversi possono produrre risultati diversi
- la scelta della soglia θ è cruciale
  - θ troppo piccola, vengono determinati troppi cluster
  - θ 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 θ = *a* to *b* step *c*
    - Eseguire *s* volte l'algoritmo BSAS, ogni volta processando i pattern con un ordine differente
    - stimare m_θ come il numero più frequente di cluster
  - end for
  - visualizzare il numero di cluster m_θ vs il parametro θ
  - 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
- algoritmo (alla lavagna)

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)
  - non funziona bene su dati altamente dimensionali (soffre del problema della curse of dimensionality)
  - 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 $f_j(x|\Theta_j)$, ognuna delle quali rappresenta un cluster
  - Una mistura è descritta dalla seguente formula

$$p(x) = \sum_{j=1}^{K} \pi_j f_j(x|\Theta_j)$$

•πj è la probabilità della j-esima componente

•NOTA: ovviamente il metodo di clustering funziona bene se i dati sono conformi al modello

Model-based clustering

- Per massimizzare il fit di dati e modelli tipicamente si utilizza un approccio “Maximum Likelihood”
- Dato un dataset D che contiene N punti D={x1..xN}, si massimizza la likelihood (produttoria di tutti i p(xi))

$$\mathcal{L}(\theta_1, ..., \theta_K, \pi_1, ..., \pi_K | D) = \prod_{i=1}^{N} \sum_{j=1}^{K} \pi_j f_j(x_i | \theta_j)$$

- Funzione molto difficile da ottimizzare, tipicamente non si può fare in modo analitico, di solito si utilizza l'EM (Expectation Maximization)

Gaussian Mixture Models

- Tecnica di model-based clustering più utilizzata (soft clustering)
- Assume che ogni componente della mistura (ogni cluster) sia gaussiano

$$f_j(x_i|\theta_j) = f_j(x_i|\mu_j, \Sigma_j) = \frac{1}{2\pi^{d/2}|\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T \Sigma_j^{-1}(x_i-\mu_j)}$$

Gaussian Mixture Models

Assunzioni sulla forma della matrice di covarianza portano a diverse forme delle misture

- Sferica
- Diagonale
- Full
- Diversa / uguale per ogni cluster

(vedi parte sulla classificazione)

Gaussian Mixture Models

- il modello è stimato utilizzando Expectation-Maximization (EM)

IDEE: (Non vediamo nel dettaglio)

- Algoritmo iterativo, parte da un modello iniziale e lo migliora iterativamente
- Concettualmente simile al kmeans, ma tiene conto del “grado di appartenenza” ad un clustering

Gaussian Mixture Models

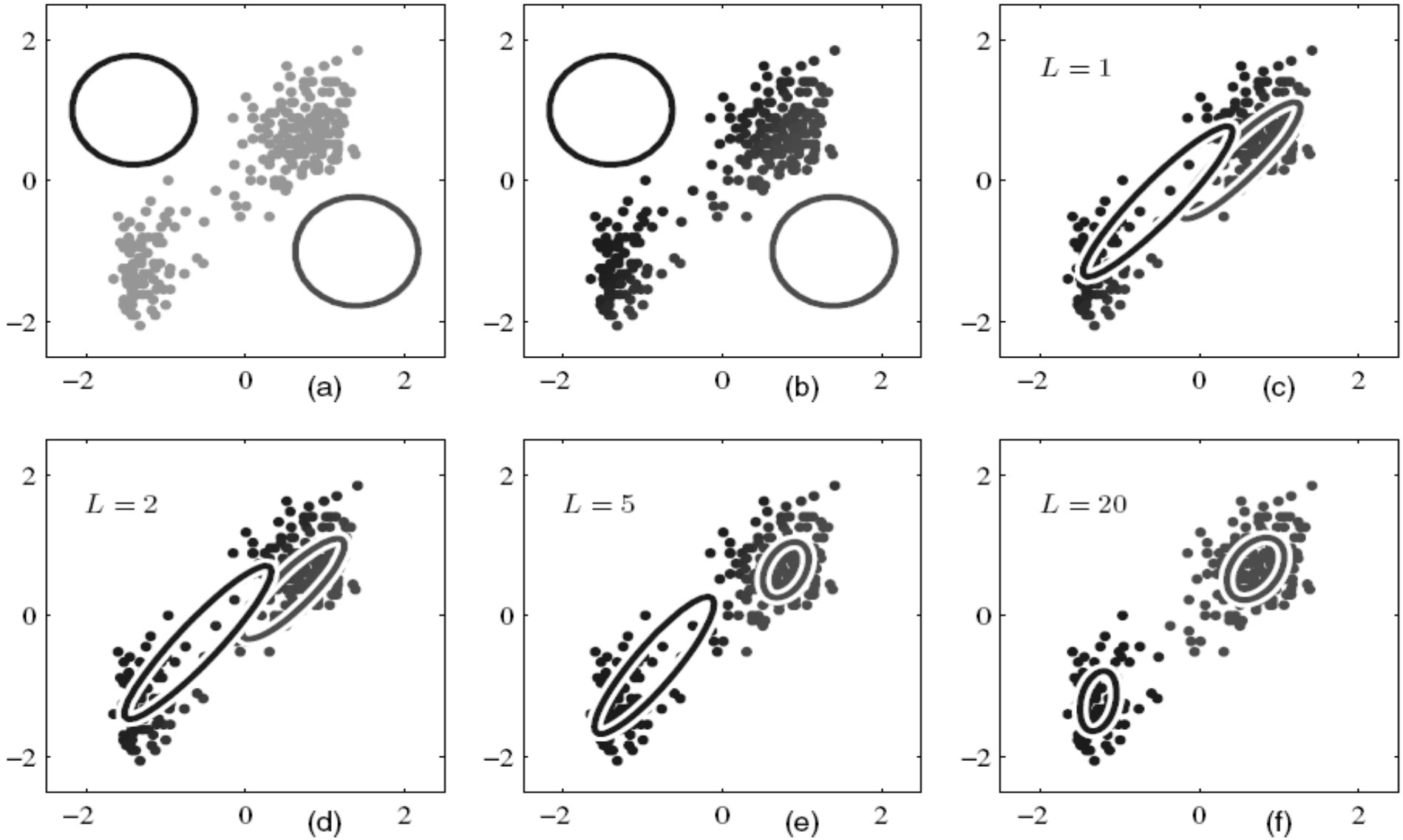
- Cicla continuamente tra questi due passi.

  E-step. Data la mistura, stima il grado di appartenenza di ogni punto alle diverse gaussiane

  M-step. Ristima i parametri delle gaussiane utilizzando queste informazioni

- Esempio: stima della media di una mistura di 2 gaussiane (alla lavagna)

Esempio



Model based clustering

VANTAGGI:

- molto utilizzato in svariati contesti per la sua flessibilità
- ritorna anche la probabilità con cui un punto appartiene ad un cluster

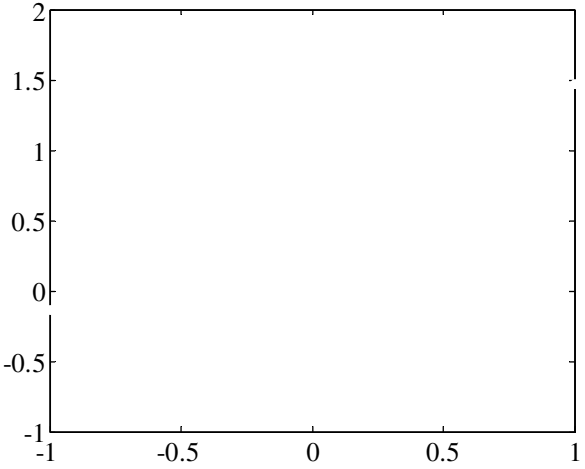
SVANTAGGI:

- l'inizializzazione è un problema
- Come si determina il numero di cluster?
  - il problema può essere visto come un problema di model selection

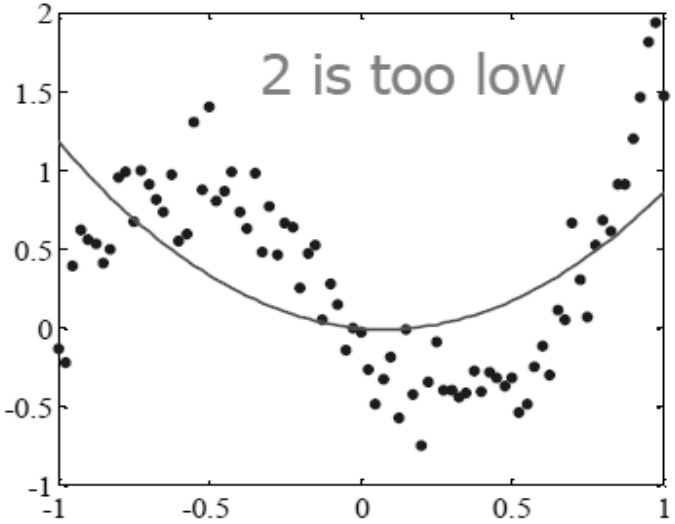
Model selection

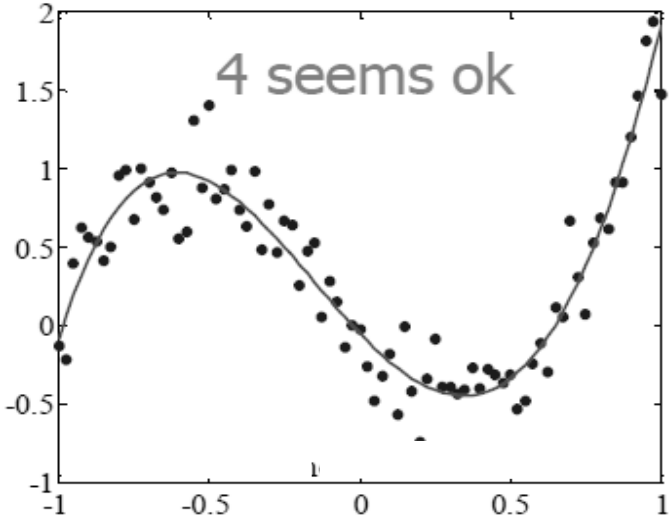
⇨ Problema molto noto in pattern recognition (tipicamente previene l'overtraining)

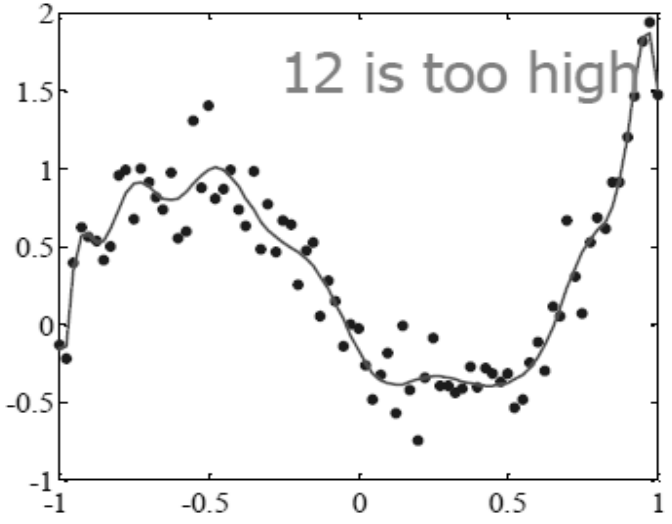
ESEMPIO: Si ha un insieme di punti su cui si vuole fittare un polinomio



Model selection: determinare il grado del polinomio

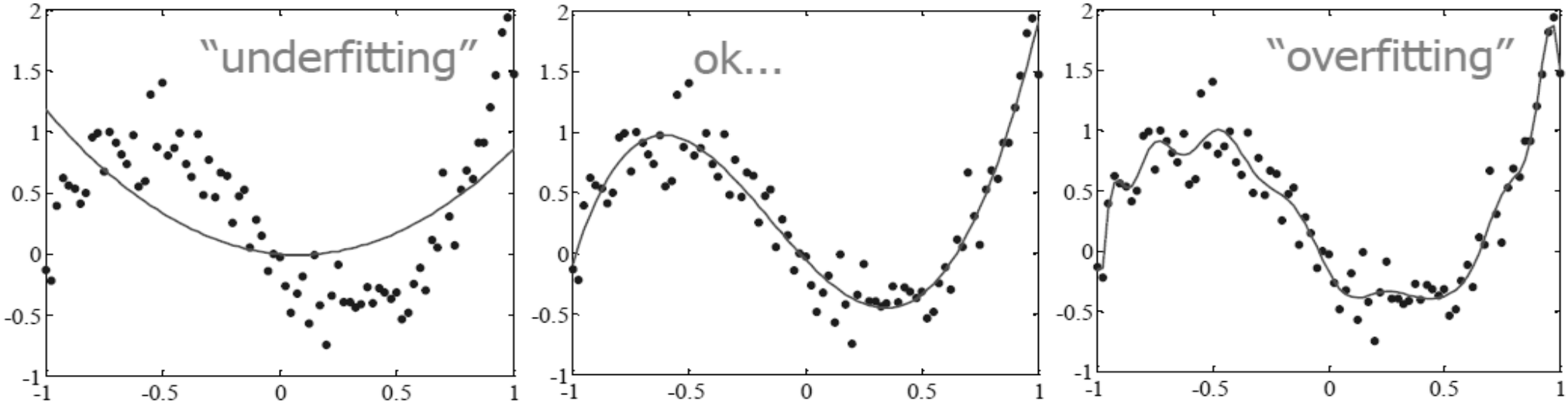






Model selection

- In generale: la model selection mira ad identificare il trend generale dei dati ignorando il rumore



Model Selection

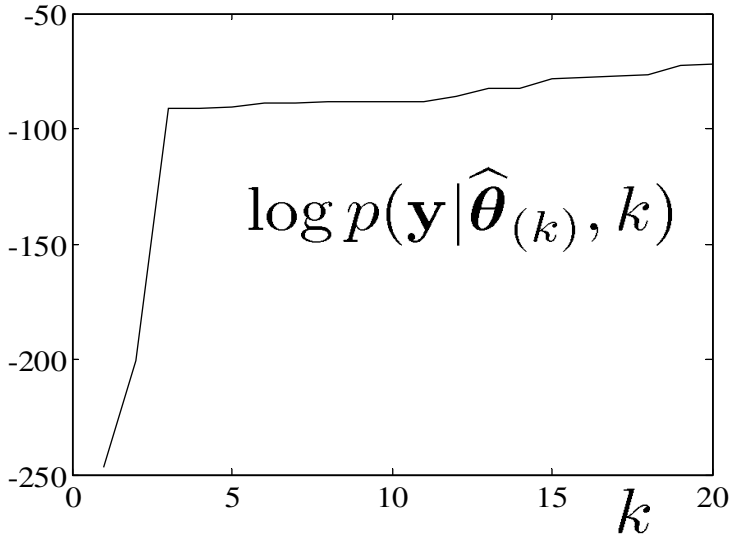
- Approccio tipico:
  - Addestrare molti modelli, ognuno con una dimensione (grado nell'esempio del polinomio) diversa
  - Scegliere il modello che massimizza un criterio di “ottimalità”

Domanda: qual'è il criterio di ottimalità migliore?

Prima soluzione ingenua: scegliere il modello che massimizza la loglikelihood!

Model Selection

- Problema: la loglikelihood è non decrescente al crescere dell'ordine
  - Più aumento il grado del polinomio, migliore è il fit del modello ai dati (caso estremo, il polinomio passa per ogni punto, fit ottimale)



Criterio non applicabile!

Model Selection

- Alternativa: approccio di “penalized likelihood”
  - Trovare un compromesso tra l'accuratezza del fitting e la semplicità del modello
- In pratica si aggiunge alla likelihood una penalità che cresce al crescere della dimensione del modello (scoraggia modelli troppo grandi)

$$K_{best} = \arg\max_k \left(LL(\{x1,..,xN\}|\theta_k) - C(k) \right)$$

complexity penalty

Esempi: BIC, MDL, MML, AIC, ...

Model Selection

- Esempio: BIC: Bayesian Information Criterion

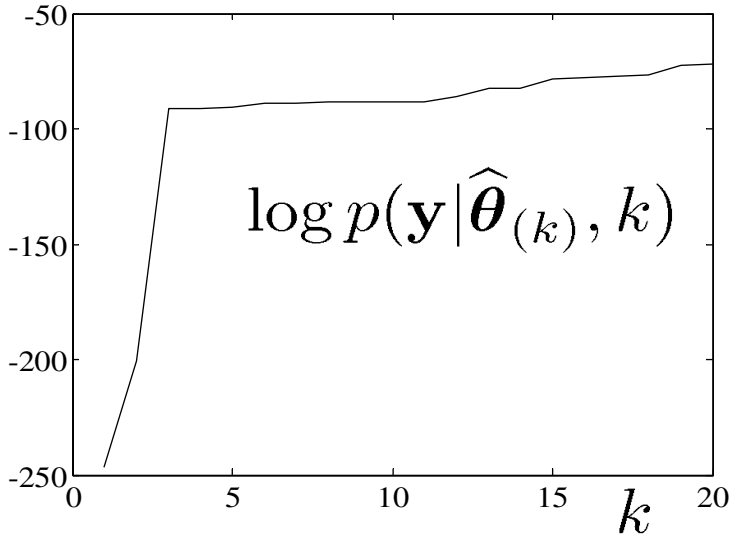
$$k_{\text{best}} = \arg\max_k \left\{ \text{LL}\left(\{ x_1, \ldots, x_N \} | \theta_k \right) - \frac{k}{2} \log\left(N \right) \right\}$$

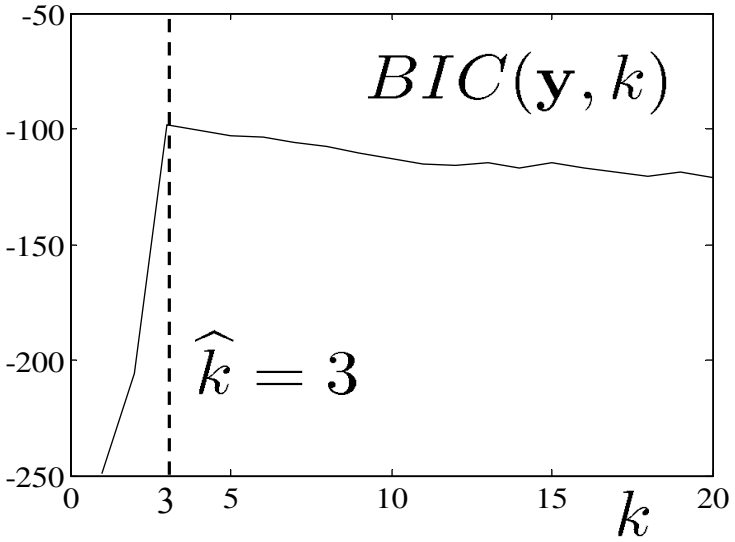
cresce con k

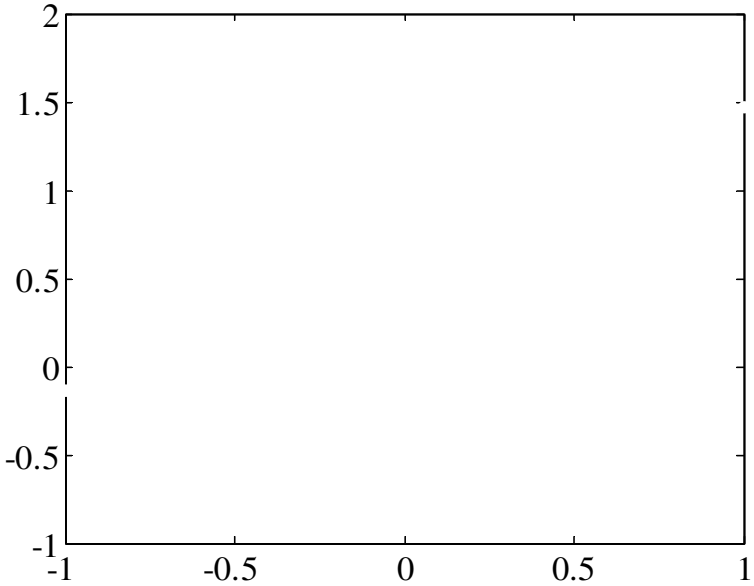
decresce con k
(penalizza k grandi)

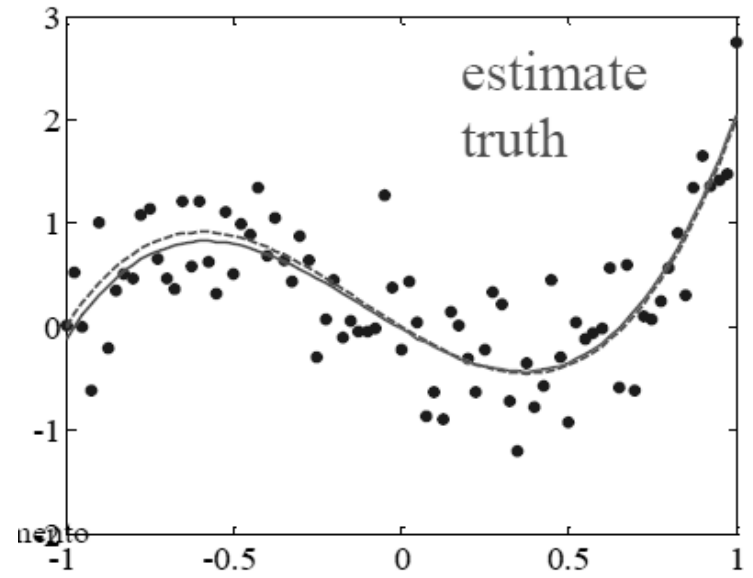
Model Selection

Esempio del
polinomio di prima









Clustering gerarchico

Clustering gerarchico

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

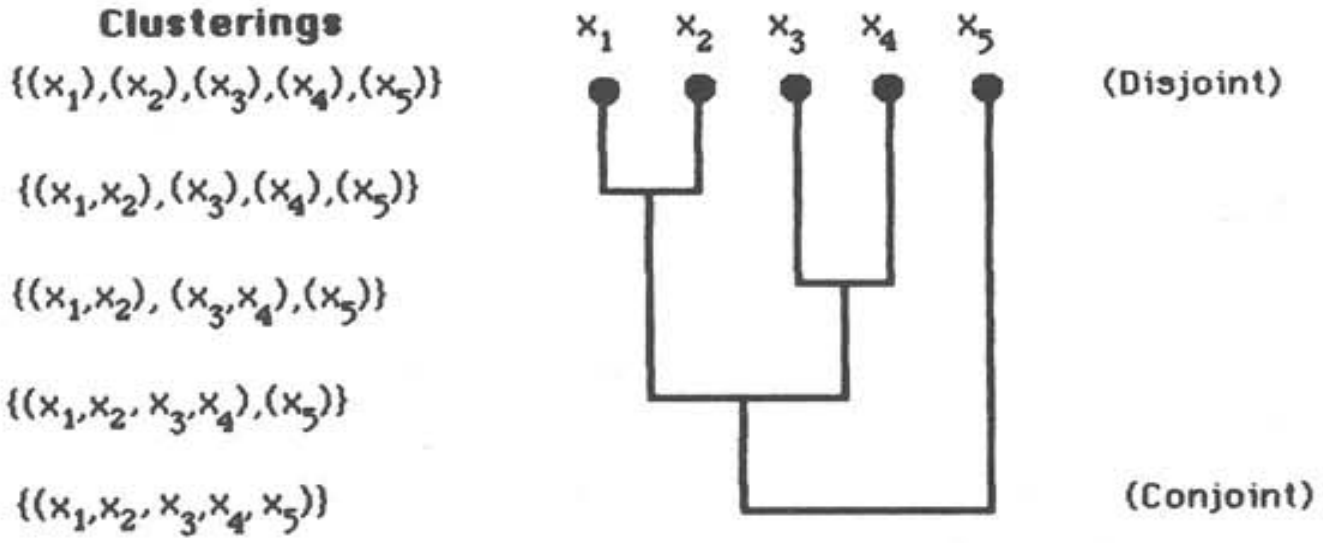


Figure 3.2 Example of dendrogram.

Clustering gerarchico

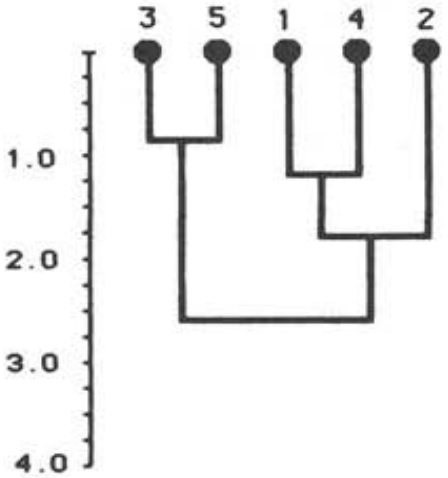
- Clustering gerarchico agglomerativo:
  - si parte da una partizione in cui ogni cluster contiene un solo elemento
  - si continua a fondere i cluster più “simili” fino ad avere un solo cluster
  - definizioni diverse del concetto di “cluster più simili” generano algoritmi diversi
- Approcci più utilizzati:
  - single link
  - complete link
  - formulazione con le matrici (alla lavagna)

Clustering gerarchico

Single Link: $d(C_{rs},C_j) = \min\{d(C_r,C_j), d(C_s,C_j)\}$

Complete Link: $d(C_{rs},C_j) = \max\{d(C_r,C_j), d(C_s,C_j)\}$

	1	2	3	4	5
1	0	2.3	3.4	1.2	3.7
2		0	2.6	1.8	4.6
3			0	4.2	**0.7**
4				0	4.4
5					0



	1	2	3,5	4
1	0	2.3	3.4	**1.2**
2		0	2.6	1.8
3,5			0	4.2
4				0

	1,4	2	3,5
1,4	0	**1.8**	3.4
2		0	2.6
3,5			0

	1,2,4	3,5
1,2,4	0	**2.6**
3,5		0

single link

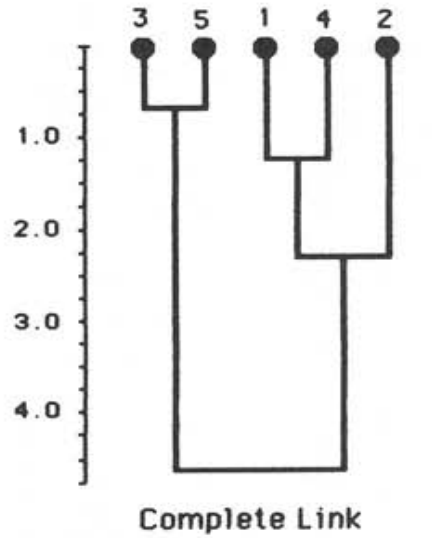
	1	2	3,5	4
1	0	2.3	3.7	**1.2**
2		0	4.6	1.8
3,5			0	4.4
4				0

	1,4	2	3,5
1,4	0	**2.3**	4.4
2		0	4.6
3,5			0

	1,2,4	3,5
1,2,4	0	**4.6**
3,5		0

complete link

Single Link



Clustering gerarchico

Altri criteri di unione dei cluster

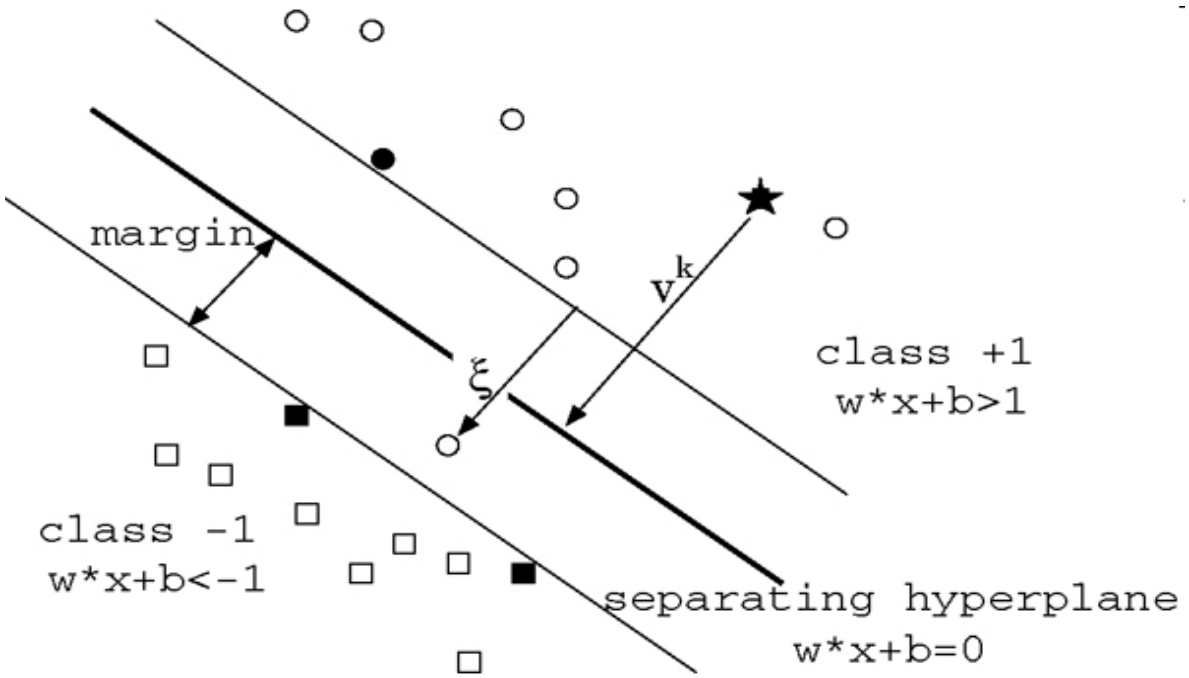
- UPGMA (Unweighted pair group method using arithmetic averages)
  - la distanza tra cluster è definita come la media delle distanze di tutte le possibili coppie formate da un punto del primo e un punto del secondo
  - utilizzato nel periodo iniziale della filogenesi

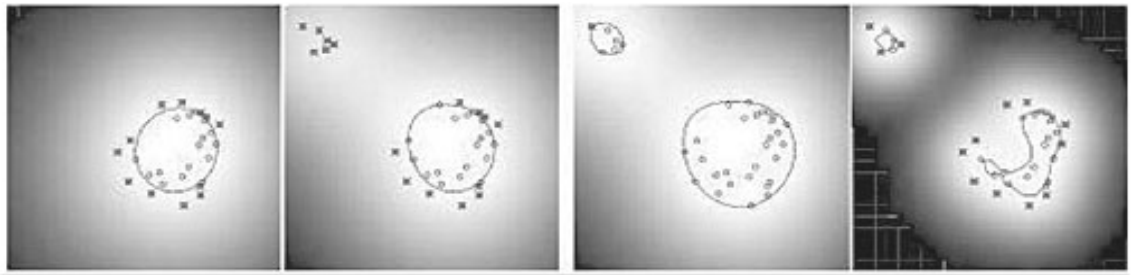
- Metodo di Ward
  - fonde assieme i cluster che portano alla minima perdita di informazione
  - informazione intesa in termini di varianza

Altre metodologie

Clustering con Kernel Machines

- ⇨ Idea, utilizzo delle One Class Support Vector Machines
- ⇨ One Class Support Vector Machines
  - ⇨ SVM che si addestrano con una sola classe
  - ⇨ SVM trovano il miglior iperpiano, OCSVM trova la miglior ipersfera che racchiude i dati





Altre metodologie

- Idea del clustering:
  - rappresentare ogni cluster con una OC-SVM
  - utilizzare un algoritmo iterativo come K-means
  - distanza = distanza dal centro della sfera (si può calcocare)
  - ricalcolo dei rappresentanti (addestramento di ogni OC-SVM con i punti assegnati al cluster corrispondente)
  - si ottengono cluster di qualsiasi forma
  - Problemi: determinare i parametri ottimali della OC-SVM

  - Altre versioni: versione soft clustering

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

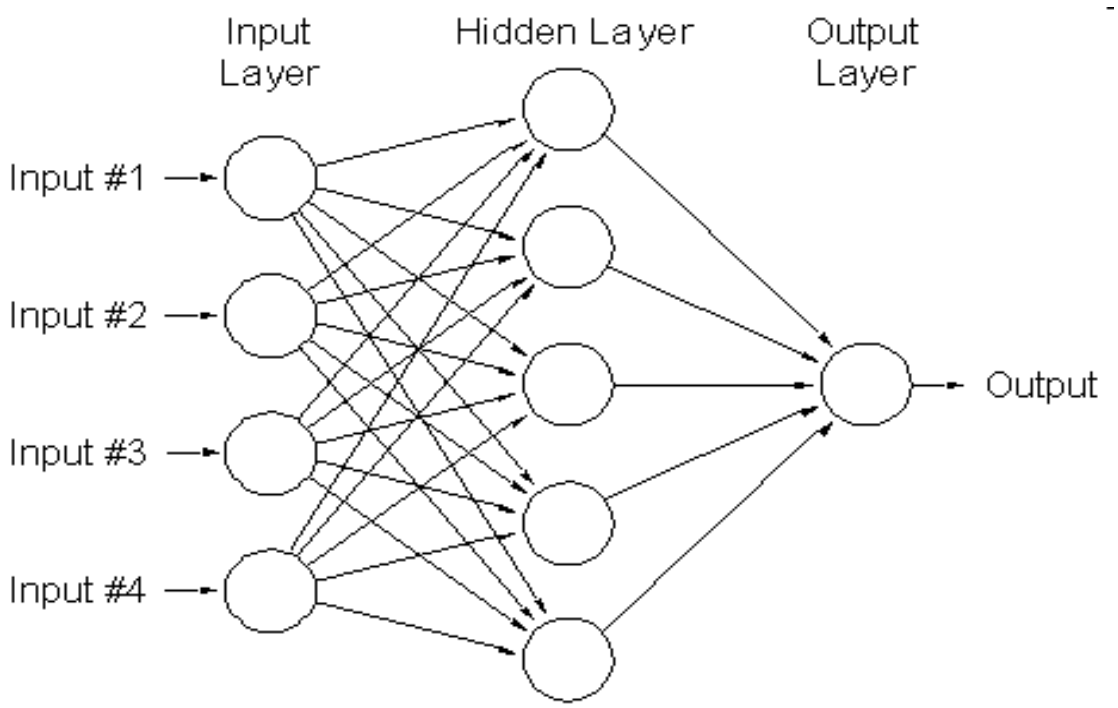
- Clustering con logica fuzzy
- Tipicamente si re-implementano diverse tecniche di clustering con la logica fuzzy
- Esempio: fuzzy K-means
  - Non troppo dissimile dal GMM, ma derivato in un altro dominio

Altre metodologie

Clustering con le reti neurali

Reti neurali: modello di calcolo che replica il meccanismo del cervello umano

- Tante piccole unità di calcolo elementari (i neuroni)
- I neuroni sono collegati assieme a formare una rete (livelli nascosti)
- L'uscita di un neurone dipende dalla somma pesata dei suoi ingressi (pesi = sinapsi)
- Esiste una funzione di attivazione che determina l'eventuale "accendersi" del neurone

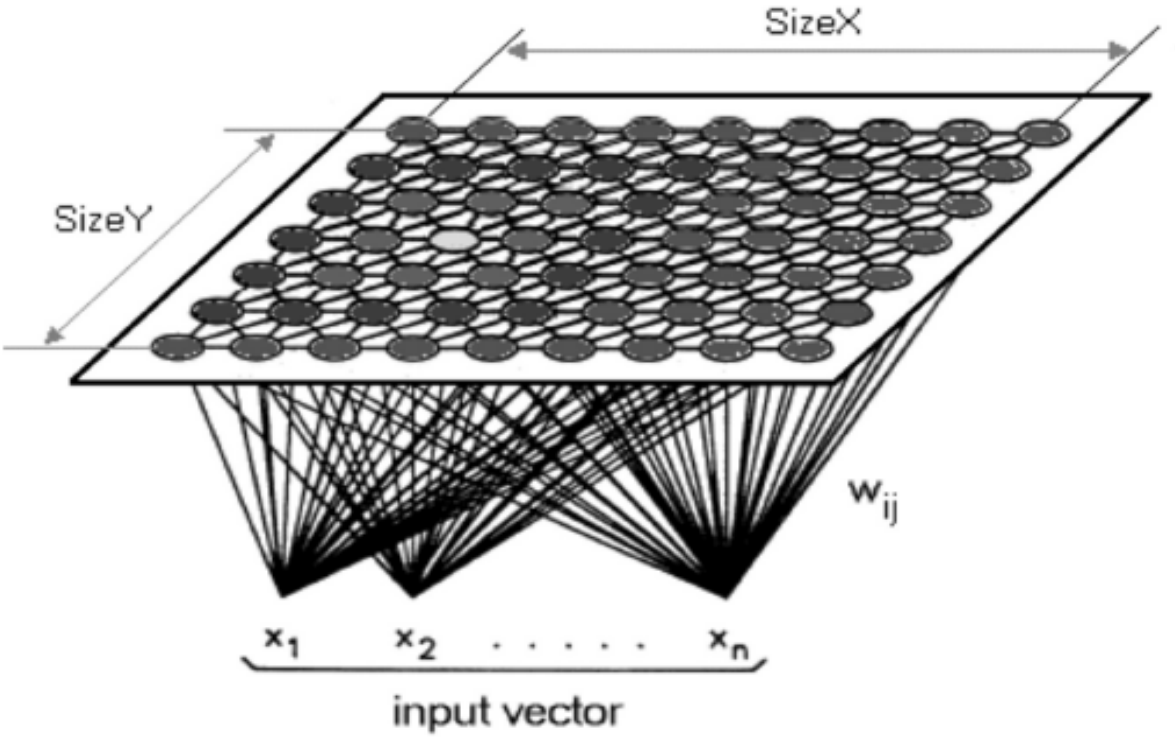


Altri metodi

- Clustering con le reti neurali: le SOM (Self-Organizing Maps)

  SOM:

- reti neurali ad un singolo strato, dove l'uscita è tipicamente bi-dimensionale
- ogni ingresso è connesso a tutti i vettori in uscita
- ogni neurone ha un vettore di pesi della stessa dimensionalità dell'ingresso



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

Altri metodi

Interpretazione:

- nella fase di addestramento i pesi di tutto il vicinato sono spostati nella stessa direzione
  - elementi simili tendono ad eccitare neuroni adiacenti.
- le SOM formano una mappa semantica dove campioni simili vengono mappati vicini e i dissimili distanti.
- Rappresentano un ottimo modo di visualizzare dati altamente dimensionali