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, ...)

- LABORATORIO: implementazione di alcuni semplici algoritmi di clustering

2

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, Alghoritms for clustering data, 1988

  ⇨ Jain et al., Data Clustering: a review, ACM Computing Surveys, 1999

3

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

4

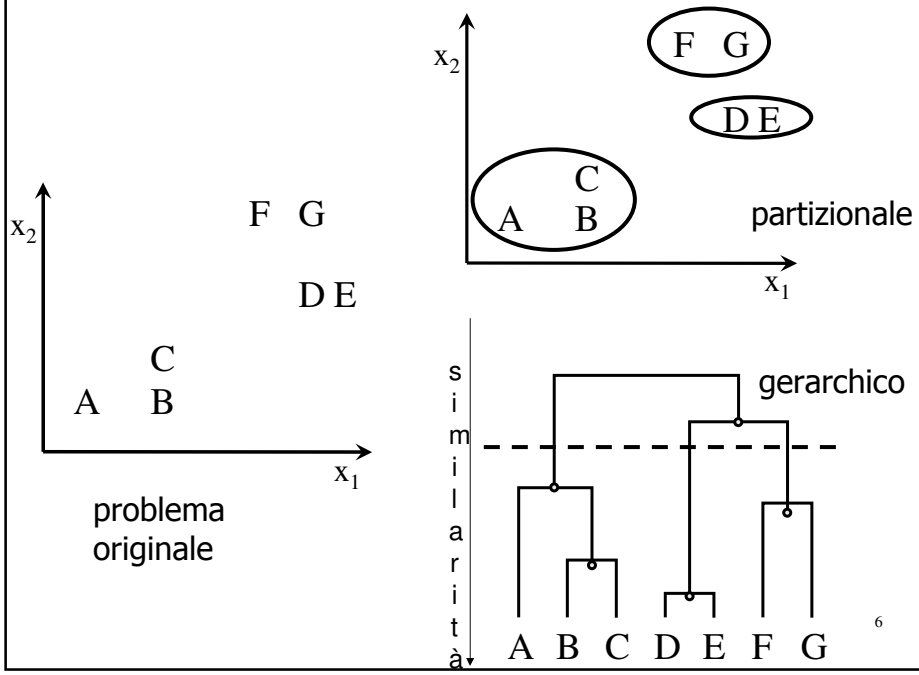
Gerarchico vs partizionale

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

5



6

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,...

7

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

8

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

9

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

10

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.

⇨ vengono utilizzate tutte le features sia per il calcolo della distanza che per il clustering

11

Monothetic vs polythetic

⇨ Esempio di algoritmo monothetic [Anderberg 73]

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

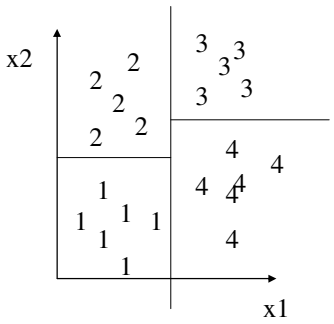
⇨ ognuno di questi data sets viene poi diviso in due utilizzando la seconda feature

⇨ si procede così fino alla fine

⇨ Svantaggi:

⇨ d features 2^d clusters

⇨ d grande, troppo frammentato



12

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)

13

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!

14

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)

15

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

16

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

17

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

18

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

19

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≤j≤m} d(x_i, C_j)
      if d(x_i, C_k) > θ
            m = m+1
            C_m = {x_i}
      else
            C_k = C_k ∪ {x_i}
            (se necessario aggiornare i rappresentanti)
      end if
end for
```

20

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

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

21

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

22

Clustering sequenziale

⇨ Metodo per calcolare il numero ottimale di clusters:

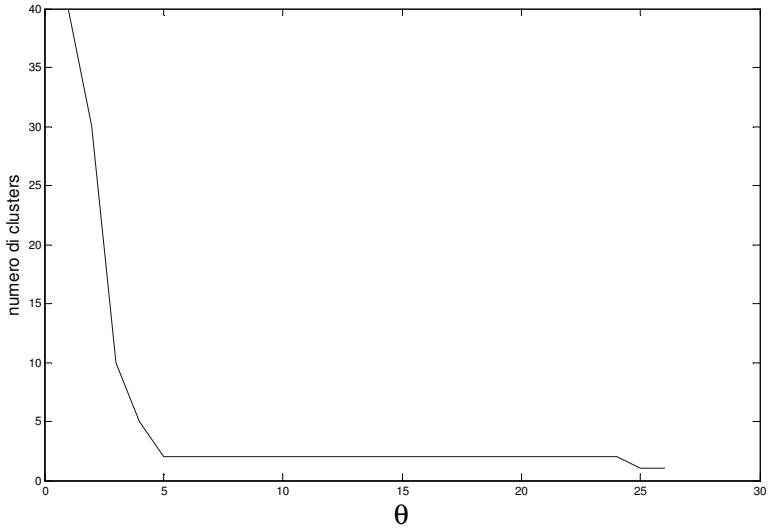
- ⇨ for θ = *a* to *b* step *c*
  - ⇨ Eseguire *s* volte l'algoritmo BSAS, ogni volta processando i pattern con un ordine differente
  - ⇨ stimare m_θ come il numero più frequente di cluster
- ⇨ end for
- ⇨ visualizzare il numero di cluster m_θ vs il parametro θ
- ⇨ il numero di cluster ottimale è quello della regione "piatta" più lunga

⇨ dettagli

- ⇨ *a* è la distanza minima tra i punti, *b* la distanza massima
- ⇨ assumiamo che "esista" un clustering

23

Clustering sequenziale



24

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)

25

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

26

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

27

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)

28