Clustering

Corso di AA, anno 2017/18, Padova

Fabio Aiolli

08 Gennaio 2018

Clustering

- Il Clustering è il processo di raggruppamento di un insieme di oggetti in gruppi di oggetti simili
- La forma più comune di apprendimento non-supervisionato (unsupervised learning)
- Al contrario dell'apprendimento supervisionato, nell'apprendimento non supervisionato non disponiamo di una classificazione degli esempi
- Task comune e molto importante che trova innumerevoli applicazioni
- Non solo clustering di esempi. Per esempio, clustering di features.

Problema del Clustering

Dati:

- Un insieme di esempi $D = \{d_1, \ldots, d_n\}$, $d_i \in \mathcal{D}$
- Una misura di similarità
- Un numero desiderato di clusters K

Calcolare:

- Una funzione di assegnamento $\gamma : \mathcal{D} \to \{1, \ldots, K\}$ tale che nessun cluster sia vuoto

Scelte da fare nel Clustering

- Rappresentazione per il clustering
  - Vector space? Normalizzazione? Kernels?
  - Necessita di una notione di similarità/distanza
- Quanti cluster?
  - Fissato a priori?
  - Completamente guidato dai dati?
  - Evitare clusters "banali" - troppo grandi o troppo piccoli

Funzione Obiettivo

- Spesso, il goal di un algoritmo di clustering è quello di ottimizzare una funzione obiettivo
- In questi casi, il clustering è un problema di ricerca (ottimizzazione)
- $K^n/K!$ clustering diversi possibili
- La maggior parte degli algoritmi di partizionamento partono da un assignamento (partizione) per poi raffinarlo
- Avere molti minimi locali nella funzione obiettivo implica che punti di partenza diversi possono portare a partizioni finali molto diverse (e non ottimali)

Cos'è un buon clustering? Valutazione

- Criteri interni che dipendono dalla nozione di similarità e/o dalla rappresentazione scelta. Andiamo a valutare la similarità intra-class (dovrebbe essere alta) e inter-class (dovrebbe essere bassa).
- Criteri esterni che, dato un "ground truth" dall'esterno misurano la sua "vicinanza" al clustering prodotto

Metodi interni

Un clustering è un buon clustering quando produce cluster nei quali:

- la similarità intra-class (tra esempi nello stesso cluster) è alta
- la similarità inter-class (tra esempi in cluster diversi) è bassa
- notare che tale misura di qualità dipende fortemente dalla rappresentazione scelta e dalla misura di similarità (distanza) impiegata per calcolarla

Esempio:

$$V(D, \gamma) = \sum_{k=1}^{K} \sum_{i:\gamma(d_i)=k} ||\mathbf{x}_i - c_k||^2$$

dove c_k è il centroide del k-esimo cluster (ovvero la media degli esempi assegnati al cluster k-esimo)

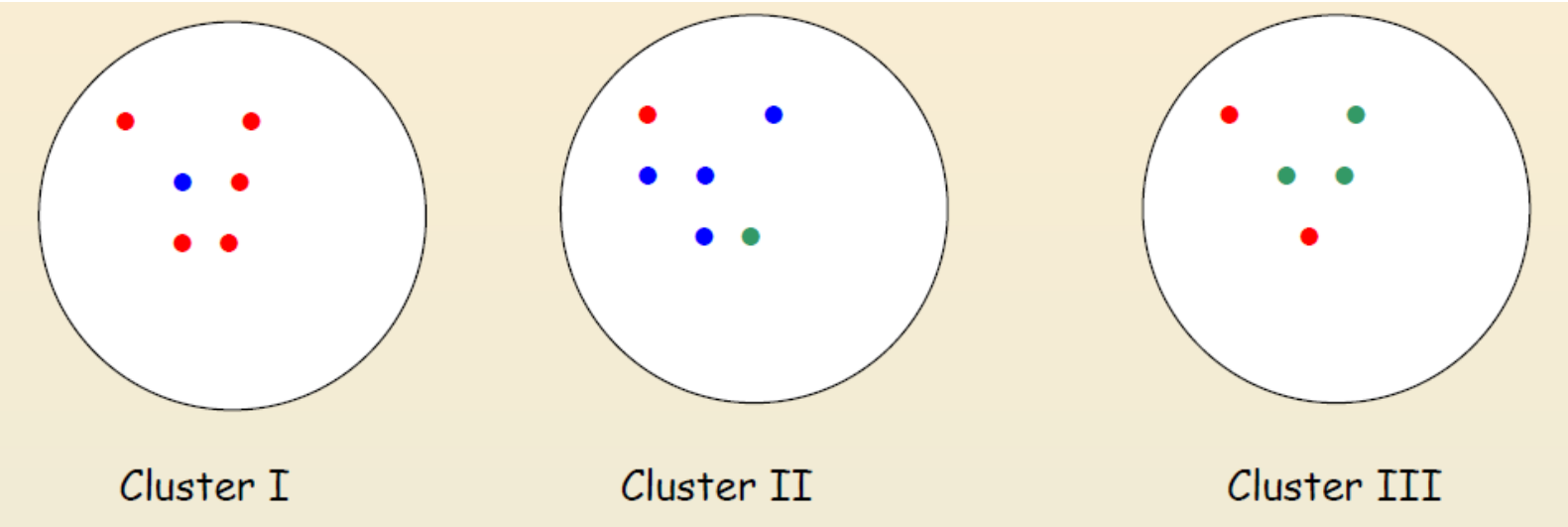
Metodi esterni

- La qualità è misurata come l'abilità dimostrata nel riconoscere alcuni o tutti i pattern nascosti e/o le classi latenti nei dati
- Disponiamo di una classificazione (ground truth) dei dati che abbiamo clusterizzato (tale ground truth NON è stato utilizzato per fare apprendimento!) e vogliamo misurare quanto il clustering prodotto assomigli a tale ground truth
- Assumiamo esempi di C classi diverse clusterizzati su K clusters $\omega_1, \ldots, \omega_K$

Metodi esterni

- Purity, ovvero il ratio tra il numero di elementi della classe dominante in un cluster e la cardinalità del cluster
- RandIndex, simile alla nozione di accuratezza (accuracy) usata nella classificazione, considerando per ogni coppia di esempi se essi sono stati correttamente distribuiti nei cluster (ovvero siano nello stesso cluster se e solo se sono della stessa classe nel ground truth)
- Altri metodi come l'entropia (o la mutua informazione) tra classi del ground truth e i clusters prodotti

Purity: un esempio



- Cluster I : $Purity(1) = \frac{max(5,1,0)}{6} = 5/6$
- Cluster II : $Purity(2) = \frac{max(1,4,1)}{6} = 4/6$
- Cluster III: $Purity(3) = \frac{max(0,2,3)}{5} = 3/5$

La purity totale sarà la media delle purity nei diversi clusters.

RandIndex

Col metodo di valutazione RandIndex, per ogni coppia di esempi, calcoliamo la seguente statistica:

- A: numero di coppie della stessa classe assegnate allo stesso cluster (true positive)
- B: numero di coppie di classe diversa assegnate allo stesso cluster (false positive)
- C: numero di coppie della stessa classe assegnate a cluster diversi (false negative)
- D: numero di coppie di classe diversa assegnate a cluster diversi (true negative)

Number of points	Same Cluster in clustering	Different Clusters in clustering
Same class in ground truth	A (tp)	C (fn)
Different classes in ground truth	B (fp)	D (tn)

RandIndex: definizione

$$RI = \frac{A + D}{A + B + C + D}$$

Number of points	Same Cluster in clustering	Different Clusters in clustering
Same class in ground truth	20	24
Different classes in ground truth	20	72

Possiamo anche considerare misure corrispondenti a Precision e Recall, ovvero:

$$P = \frac{A}{A + B} \qquad R = \frac{A}{A + C}$$

Algoritmi di clustering

- Algoritmi di partizionamento
  - Di solito partono con una partizione random (parziale)
  - Raffinandola iterativamente (p.e. K-means clustering e model-based clustering)
- Algoritmi gerarchici
  - Bottom-up o agglomerativi
  - Top-down o divisivi

Algoritmi di partizionamento

Costruiscono una partizione di n esempi in K clusters

- Dato un insieme di esempi e un numero K
- Trovano una partizione di K clusters che ottimizza un certo criterio
  - Ottimo globale: enumera esaustivamente tutte le possibili partizioni (inefficiente)
  - Metodi basati su euristiche: k-means e k-medoids sono esempi

Algoritmo K-Means: idea

- Si assume che gli esempi siano vettori a valori reali
- Per ogni cluster minimizziamo la media della distanza tra gli esempi e il "centro" del cluster (centroide):

$$\mu(c) = \frac{1}{|c|} \sum_{x \in c} x$$

- Le istanze vengono assegnate ai clusters in base alla similarità/distanza degli esempi dai centroidi dei cluster correnti

Algoritmo K-Means

1. Genera K punti nello spazio (seeds). Questi punti rappresentano i centroidi iniziali
2. Assegna ogni esempio al cluster il quale centroide è il più vicino
3. Dopo aver assegnato tutti gli esempi, ricalcola la posizione dei K centroidi
4. Ripeti i passi 2 e 3 fino a quando i centroidi si stabilizzano

Algoritmi Gerarchici

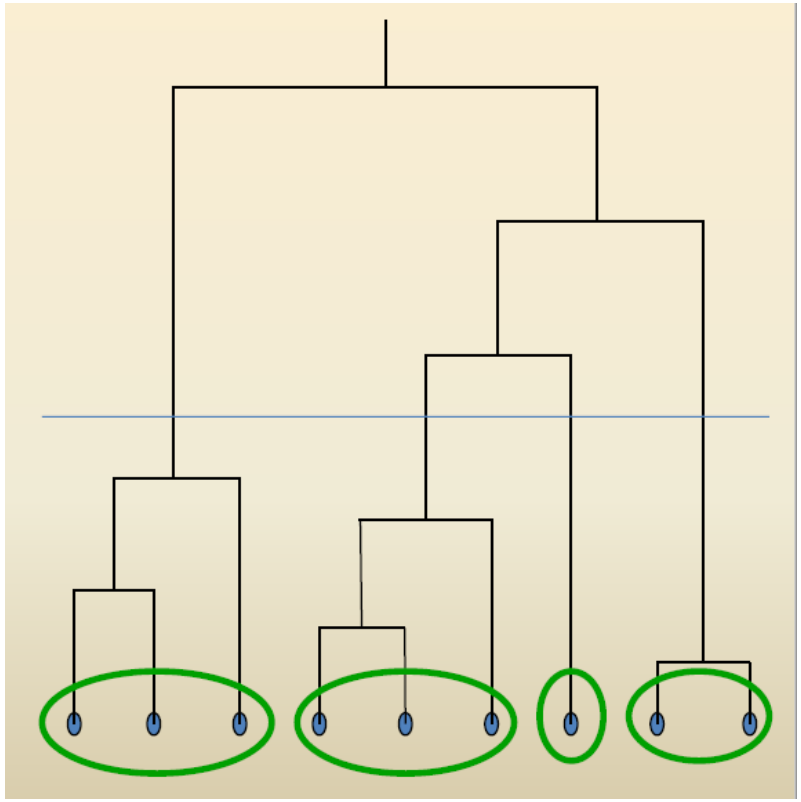
Il numero di cluster K ottimale non è dato di solito. Gli algoritmi gerarchici sono una buona alternativa in questi casi

Costruiamo una tassonomia gerachica (tree-based) da un insieme di esempi che rappresenta la struttura dei cluster (dendogramma o dendrogram in inglese)

Un primo approccio è considerare l'applicazione ricorsiva di un algoritmo di clustering partizionale (algoritmo gerarchico divisivo)

Dendogramma per algoritmi agglomerativi

- L'asse y del dendogramma rappresenta la similarità della combinazione, ovvero la similarità tra i cluster che vengono fusi
- Si assume che l'operazione di merge sia monotona, ovvero se $s_1, \ldots, s_k$ sono successive similarità ottenute nella fusione, allora $s_1 > s_2 > \cdots > s_k$
- Il clustering è ottenuto "tagliando" il dendogramma al livello desiderato: ogni componente connessa è un cluster



Clustering Gerarchico Agglomerativo (HAC)

- Partiamo con un cluster separato per ogni esempio. Poi, fondiamo via via le coppie di cluster più vicine, fino ad ottenere un solo cluster
- La storia delle fusioni forma un albero binario o gerarchia (dendogramma)

Cosa significa "la coppia di cluster più vicina"? Come possiamo misurare la distanza tra due clusters?

- Single-link: Similarità tra gli esempi più simili nei clusters
- Complete-link: Simmilarità tra gli esempi più distanti nei clusters
- Centroid: Similarità tra i centroidi dei clusters
- Average-link: Similarità media tra coppie di esempi dei clusters

Riepilogo su clustering gerarchico agglomerativo

Single-link	Max sim of any two points	$O(N^2)$	Chaining effect
Complete-link	Min sim of any two points	$O(N^2 \log N)$	Sensitive to outliers
Centroid	Similarity of centroids	$O(N^2 \log N)$	Non monotonic
Group-average	Avg sim of any two points	$O(N^2 \log N)$	OK