Clustering - 10/12

Cluster Analysis: la logica

Cos'è il Clustering?

L'obiettivo della cluster analysis è segmentare una collezione di oggetti. Per le applicazioni di data analysis gli oggetti corrispondono ai dati osservati.

Ciò è richiesto in quanto i dati che stiamo considerando sono unlabeled, cluster analysis è infatti un metodo non supervisionato.

Gli algoritmi di clustering generano sottoinsiemi delle osservazioni chiamati cluster. Tutte le osservazioni in un cluster sono simili.

Definire appropriatamente il concetto di similarità è critico.

Il ruolo della similarità

Al fine di clusterizzare le osservazioni è necessario definire quanto esse siano simili/dissimili.

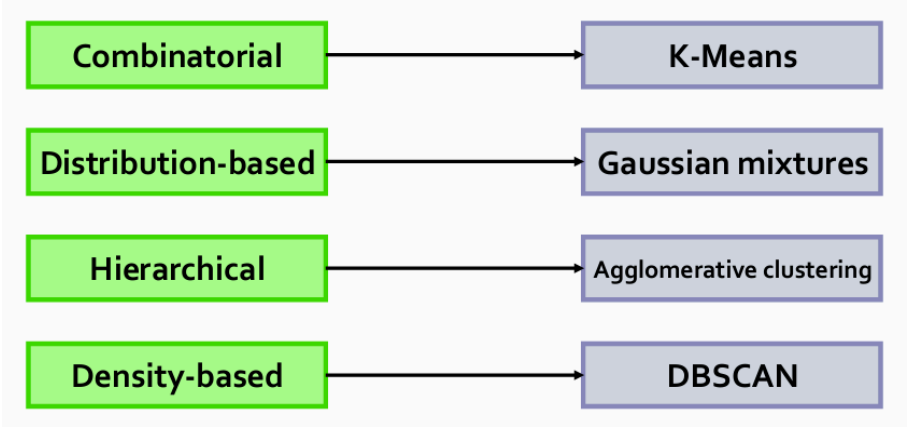
Non c'è una definizione universale di similarità, dipende dalla specifica applicazione o da criteri soggettivi.

Risulta possibile pensare alla misura di similarità come la controparte della fuzione di loss usata in precedenti applicazioni supervisionate.

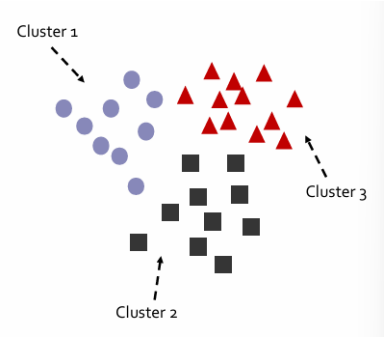
Se la loss function non è adeguata (suited) al modello che genera i dati il processo di supervised data analysis avrà scarse performance.

In maniera similare, se la misura di similarità non è adeguata (suited) alle osservazioni le operazioni di cluster analysis avranno scarse performance.

Metodi di Cluster Analysis popolari



L'assegnazione di ogni osservazione ad un cluster può essere ottenuta dalla minimizzazione, su un set di possibili assegnazioni, di una funzione adeguata (suited) basata sulla misura di similarità scelta.

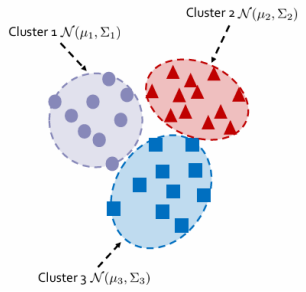


Combinatorial

Il problema di trovare l'assegnazione ottima è di solito combinatoriale visto che dobbiamo testare un numero proibitivamente grande di possibili combinazioni.

Distribution-based

I metodi Distribution-based assumono che le osservazioni siano state estratte da un misto di modelli generativi (ad esempio i dati possono essere estrazioni di diverse distribuzioni Gaussiane).



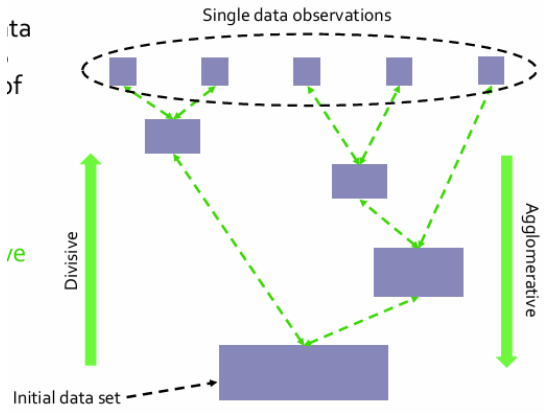
I parametri dei modelli sono stimati dalle osservazioni usando l'algoritmo expectation-maximization.

Le osservazioni sono poi assegnate ad un cluster (ad esempio ad un modello generativo) secondo la probabilità di essere state generate da uno dei modelli.

Hierarchical

I metodi gerarchici assegnano ogni osservazione ad un cluster secondo la similarità tra coppie di gruppi di osservazioni.

Il processo di clustering costruisce una struttura ad albero per rappresentare i dati.

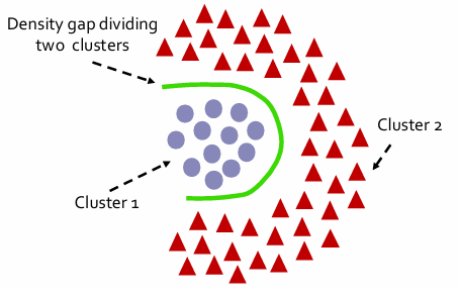


Tale struttura può essere ottenuta con un paradigma agglomerativo (bottom-up) o divisivo (top-down).

Una fetta di albero ad una data altezza fornisce i cluster.

Density-based

I metodi Density-based considerano la struttura dei dati più finemente, i cluster seguono più da vicino, accuratamente, la distribuzione dei dati nello spazio.



I clusters sono aree nello spazio dei dati con osservazioni densamente concentrate.

La densità è di solito intesa come il numero di osservazioni che cadono entro un certo volume.

Tecniche popolari di clustering

K-means clustering

Assumiamo di voler segmentare i nostri dati in K cluster.

Desideriamo trovare una assegnazione ottima delle osservazioni ad ogni cluster $C_1, \ldots, C_K$ al fine di minimizzare la somma dei quadrati

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$$

μ_k è il centroide del cluster C_k, definito come

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

I centroidi restituiscono (yield) il minimo di J per delle assegnazioni (assignments) fissate r_{nk} (il valore medio è il minimizzatore della dispersione di una variabile aleatoria).

Logica del K-means

Come possiamo trovare gli assignments ottimali? Possiamo enumerarle tutte e scegliere la migliore.

Tramite la enumerazione degli assignments possiamo calcolare i corrispondenti centroidi e valutare J. Questo però è un problema combinatoriale ed è quindi estremamente inefficiente.

Dati i centroidi μ_k, le assegnazioni r_{nk} dovrebbero seguire la regola nearest neighbor al fine di ridurre il costo J. $r_{nk} = 1$ se $k = argmin_j ||x_n - \mu_j||^2$, $r_{nk} = 0$ altrimenti.

Però, il valore dei centrodi dipende dagli assignments, è un problema dell'uovo e la gallina?

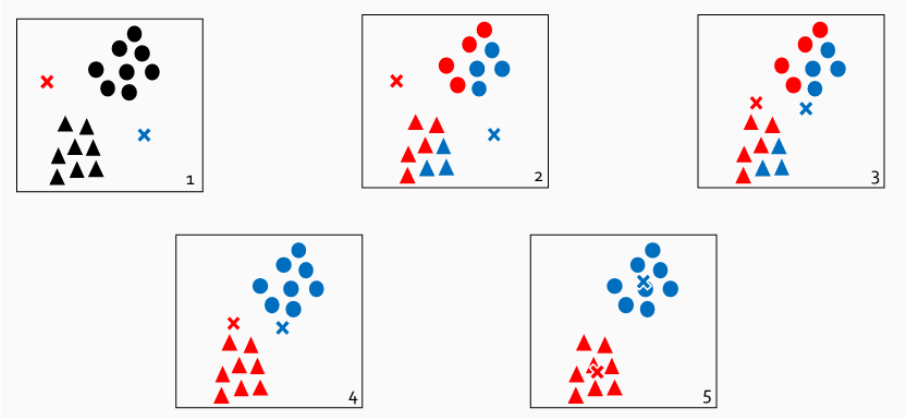
Algoritmo di Lloyd per la K-means

Minimizzare J è un problema difficile visto che non sappiamo come trovare l'assegnazione r_{nk} che porta al minimo globale di J.

> 💡 K-means è quindi implementato con algoritmi iterativi per approssimare l'assegnazione ottima e convergere ad una configurazione ammissibile.

Esistono varie implementazioni d K-means che offrono un certo compromesso tra accuratezza ed efficienza, il più popolare è l'algoritmo di Lloyd.

```
Algoritmo: K-means (algoritmo di Lloyd)
   Input: data set X, numero di cluster K, centroidi iniziali \mu_k
   Repeat:
      -forma K cluster assegnando ogni osservazione al centroide più vicino ad essa
      -ricalcola i centroidi
   Until: si raggiunge il criterio di terminazione
```



Sketch dell'analisi di convergenza di K-means

K-means, implementato con l'algoritmo di Lloyd, converge.

La regola Nearest Neighbor assicura che quando una cluster configuration cambia il costo di J decresce.

Se i cluster non cambiano tra due iterazioni allora non cambieranno più.

Visto che c'è un numero finito di possibili assegnazioni, K-means deve convergere ad una soluzione in un numero finito di passi.

Però, il numero di passi può essere molto grande, specialmente quando i dati sono in uno spazio ad alta dimensionalità. Questa è la motivazione per la quale è usato un criterio di terminazione (termination criterion).

K-means non garantisce la convergenza all'assegnazione ottimale. Ci potrebbero essere multeplici configurazioni ammissibili che soddisfano la regola nearest neighbor.

K-means initialization

Le performance di K-means (gli specifici cluster trovati e il tempo di convergenza) possono essere profondamente influenzate dall'inizializzazione dei centroidi.

Scegliere μ_k in modo casuale è un metodo molto semplice e popolare ma non ottimale, ad esempio, scegliere centroidi iniziali che sono lontani dalle osservazioni può rallentare l'algoritmo.

Un metodo di inizializzazione popolare usato in molte librerie software è il cosiddetto K-means++. Questo metodo usa una euristica probabilistica per scegliere i centroidi iniziali. I centroidi sono scelti casualmente dalle osservazioni ma la probabilità che siano scelte dipende dalla loro distanza dai centroidi già selezionati, quanto più un punto è lontano da un centroide, tanto più alta la probabilità che questo punto sia scelto come altro centroide.

GMM (Gaussian Mixture Model)

Assumiamo che le osservazioni del data set siano generate da un misto di distribuzioni sottostanti (mixture of underlying distributions), ognuna rappresentativa di un cluster.

Con K cluster, il misto che da la distribuzione di entry x_n del data set è

$$p(x_n) = \sum_{k=1}^{K} \pi_k p_k(x_n|\theta_k)$$

dove $p_k(x_n|\theta_k)$ sono le likelihoods.

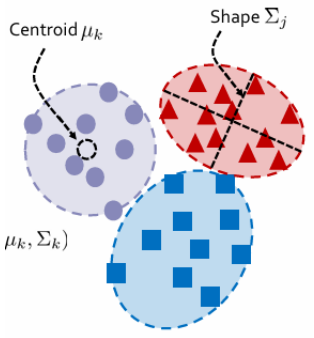
I coefficienti $\pi_1, \pi_2, \ldots, \pi_K$, con $0 \leq \pi_k \leq 1$ e $\sum_{k=1}^{K} \pi_k = 1$ sono chiamati mixing probabilities.

Le likelihood individuali $p_k(. \mid \theta_k)$ parametrizzate da θ_k sono normali multivariate h-dimensionali, con media μ_k e covarianza $\sum_k$

$$p_k(x \mid \mu_k, \sum_k) = \frac{1}{(2\pi)^{h/2}\sqrt{det \sum_k}} \exp(-\frac{1}{2}(x-\mu_k)^T \sum_k^{-1}(x-\mu_k))$$

Stimare le distribuzioni Gaussiane

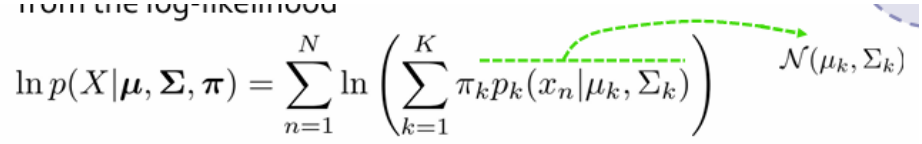
Nel modello GMM bisogna fare il fit delle K Gaussiane sconosciute rispetto ai dati $X = \{x_1, \ldots, x_N\}$.



Abbiamo i parametri ignoti $\pi = \{\pi_1, \ldots, \pi_K\}$, $\mu = \{\mu_1, \ldots, \mu_K\}$,

$\sum = \{\sum_1, \ldots, \sum_K\}$.

Potremmo fare la stima Maximum Likelihood partendo dalla log-likelihood.



La massimizzazione di questa log-likelihood è però mathematically intractable (no closed-form solution); cioè matematicamente irrisolvibile a causa dell'assenza di una soluzione in forma chiusa.

Dalla Maximum Likelihood a...

Ponendo uguale a zero il gradiente della funzione di log-verosimiglianza, otteniamo:

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}} \qquad \Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^\top}{\sum_{n=1}^{N} \gamma_{nk}} \qquad \pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}$$

Tutte queste quantità dipendono dalle responsabilities γ_{nk}, che rappresentano la probabilità a posteriori che x_n sia generata dalla k-esima Gaussiana

$$\gamma_{nk} = \frac{\pi_k p_k(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j p_j(x|\mu_j, \Sigma_j)}$$

...che a loro volta dipendono dai parametri μ_k, Σ_k, π_k che abbiamo bisogno di stimare!

...l'algoritmo expectation-maximization

La formulazione precedente suggerisce un algoritmo iterativo, chiamato algoritmo di aspettativa-massimizzazione (EM), per stimare in modo alternato i coefficienti sconosciuti e le responsabilità.

Algorithm: EM for GMM

Input: data set X,number of clusters K, initial parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}$

Repeat

Evaluate the value of the log-likelihood with the current parameters

E-step:

$$\gamma_{nk} = \frac{\pi_k p_k(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j p_j(x_n|\mu_j, \Sigma_j)}$$

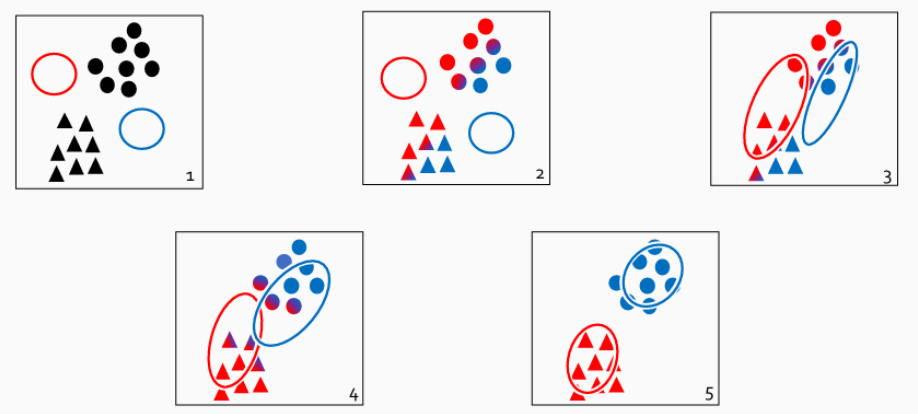
M-step:

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}} \qquad \Sigma_k = \frac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^\top}{\sum_{n=1}^{N} \gamma_{nk}} \qquad \pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}$$

Recalculate the value of the log-likelihood with the new parameters

Until termination criterion is met

Expectation-maximization per GMM



L'algoritmo EM

EM è una tecnica generale per la ricerca di soluzioni Maximum Likelihoods.

Risulta utile in presenza di variabili latenti, come variabili nel modello statistico che non possono essere direttamente osservate.

In GMM una variabile latente può essere utilizzata per rappresenta la forma Gaussiana dalla quale un punto è estratto.

EM massimizza un surrogato della objective function che permette di massimizzare la likelihood function di interesse per mezzo dell'iterazione di due passi:

- **Expectation step** (E-step): calcola l'expectation della log-likelihood (che dipende dai parametri ignoti) rispetto alla distribuzione condizionale delle variabili latenti date le osservazioni;
- **Maximization step** (M-step): trova i parametri che massimizzano l'expectation dalla E-step, che saranno usati nel successivo E-step.

EM non garantisce la convergenza al minimo globale della likelihood ma garantisce l'incremento del valore della funzione likelihood ad ogni passo.