Machine Learning per la fisica applicata e la fisica delle alte energie

Lezione 21: Clustering

Il clustering è una forma comune di unsupervised learning in cui, dato un data set $\mathcal{D} = \{\vec{x}_n\}$ con $n = 1, \dots, N$ e $\vec{x}_n \in \mathcal{X}$ con $\mathcal{X} = \mathbb{R}^D$, si vogliono raggruppare opportunamente gli $\vec{x}_n$ in gruppi, detti cluster, con caratteristiche simili.

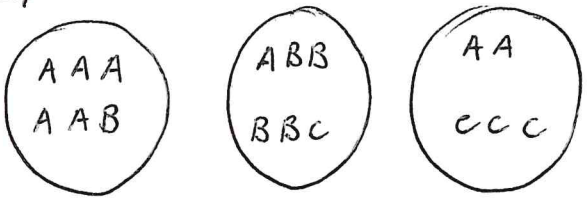
METRICHE

Siccome il clustering fa parte dell'unsupervised learning, è difficile definire delle figure di merito, dato che manca un mapping. Occorre misurare quanto dati simili siano nello stesso cluster e quanto dati sufficientemente diversi siano in cluster diversi.

a) PURITÀ: Sia N_{ij} il numero di punti nel cluster i che appartengono alla classe j. Definiamo $N_i = \sum_{j=1}^{} N_{ij}$ il numero di punti in un cluster i. Definiamo $p_{ij} = N_{ij}/N_i$. La purità di un cluster è $p_i = \max_j p_{ij}$ e la purità totale è

$$\text{PURITÀ} = \sum_i \frac{N_i}{N} p_i$$

Esempio:



$N_1 = 6 \quad N_2 = 6 \quad N_3 = 5 \quad N = 17$

$$\text{PURITÀ} = \frac{6}{17}\cdot\frac{5}{6} + \frac{6}{17}\cdot\frac{4}{6} + \frac{5}{17}\cdot\frac{3}{5} = 0{,}71$$

b) RAND INDEX: Siano $U = \{u_1, \dots, u_R\}$ e $V = \{v_1, \dots, v_C\}$ due partizioni di N dati. U è il clustering stimato e V è il clustering di riferimento. Ora definiamo:

TP (true positives): punti in U che sono anche in V (stesso cluster)

TN (true negatives): punti non in U e non in V (cluster diversi)

FN (false negatives): punti non in U ma in V (cluster diversi)

FP (false positives): punti in U ma non in V (cluster diversi)

$$\text{RAND INDEX} = \frac{TP + TN}{TP + FP + FN + TN} \qquad 0 \leq \text{RAND INDEX} \leq 1$$

Questa è la frazione delle "clustering decisions" che non sono corrette.

$$TP + FP = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40 \qquad TP = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

$FP = 20$ $\quad FN = 24$ $\quad TN = 72$ $\quad$ RAND INDEX $= 0{,}68$

HIERARCHICAL AGGLOMERATIVE CLUSTERING (HAC)

Lo hierarchical agglomerative clustering (HAC) è un algoritmo che riceve in input una matrice di dissimilarità $N \times N$ che denotiamo con $D_{ij} \geq 0$ e ritorna una struttura ad albero in cui i punti i e j con dissimilarità piccola sono raggruppati gerarchicamente. Si considerino 5 punti $\vec{x_n} \in \mathbb{R}^2$. Definiamo

$$d_{ij} = \sum_{k=1}^{2} |x_{ik} - x_{jk}|$$

la somma del modulo delle distanze in ciascuna dimensione. Costruiamo una struttura ad albero con 5 rami; raggruppiamo le coppie (1,3) e (4,5) in due cluster. Il punto rimanente è un cluster a parte. La struttura ad albero prende il nome di DENDOGRAM.

L'algoritmo, nella forma generale, parte con N gruppi, ciascuno dei quali contiene un oggetto e raggruppa i gruppi più simili finché non rimane un solo gruppo.

Poiché produrre i due gruppi più simili è un'operazione $O(N^2)$ e l'algoritmo ha $O(N)$ passi, l'algoritmo impiega un tempo $O(N^3)$. Esistono tre varianti dell'algoritmo.

1 SINGLE LINK. Nel single link clustering, anche detto nearest neighbor clustering, la distanza tra due gruppi G e H è definita come la distanza tra i due punti più vicini di un certo gruppo

$$d_{SL}(G,H) = \min_{i \in G;\, i' \in H} d_{ii'}$$

2 COMPLETE LINK. Nel complete link clustering, anche detto furthest neighbour clustering, la distanza tra due gruppi G e H è definita come la distanza tra le due coppie più distanti

$$d_{CL}(G,H) = \max_{i \in G;\, i' \in H} d_{ii'}$$

3 AVERAGE LINK. Nell'average link clustering si usa la distanza media tra tutte le coppie

$$d_{AVG}(G,H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

con n_G, n_H il numero di punti nei gruppi G e H.

Il metodo 1 produce clusters ben distanziati; il metodo 2 produce clusters compatti; il metodo 3 è una via di mezzo. Il metodo 3 NON è invariante rispetto alla scala dei punti.

Esempio: N=300 geni in funzione della temperatura T.

K-MEANS CLUSTERING

L'HAC ha alcuni problemi: è lento ($O(N^3)$) e spesso non funziona bene per data sets grandi; assume la nozione di similarità; è una procedura algoritmica priva di ottimizzazione

Il K-means algorithm non ha questi inconvenienti. L'idea è quella di calcolare la similarità in termini di distanza euclidea rispetto a centri $\mu_k \in \mathbb{R}^D$ che devono essere imparati.

Supponiamo che ci siano K centri del cluster $\vec{\mu_k} \in \mathbb{R}^D$ in modo che possiamo assegnare ad ogni punto $\vec{x_n} \in \mathbb{R}^D$ la sua distanza dal centro più vicino

$$z_n^* = \arg\min_k \| \vec{x_n} - \vec{\mu_k} \|_{\ell_2}^2$$

Ovviamente non conosciamo i centri del cluster, ma possiamo stimarli calcolando il valore medio dei punti loro assegnati

$$\vec{\mu_k} = \frac{1}{N_k} \sum_{n=1}^{k} \vec{x_n}$$

Questi passi possono essere iterati fino a convergenza.
In modo più formale, quello che vogliamo fare è trovare un minimo locale della seguente funzione, detta DISTORTION

$$J(M, Z) = \sum_{n=1}^{N} \| \vec{x_n} - \vec{\mu_{z_n}} \|^2 = \| X - Z M^T \|^2$$

dove $X \in \mathbb{R}^{N \times D}$, $Z \in [0,1]^{N \times K}$ e $M \in \mathbb{R}^{D \times K}$, contiene i centri dei cluster nelle sue colonne.

Esempio: N = 300 geni in funzione della temperatura T.

K-MEANS ++

L'algoritmo K-means ottimizza un oggetto non convesso, pertanto deve essere inizializzato con cura. Un approccio è di selezionare K punti in modo casuale, usare questi come centroidi $\vec{\mu_k}$, ripetere il procedimento per diverse inizializzazioni e prendere il risultato migliore. Questo procedimento, tuttavia, è inefficiente.

Un approccio migliore consiste nel prendere i punti K in sequenza cercando di "coprire" i dati. Cioè, prendiamo il punto iniziale con probabilità uniforme e ciascuno dei punti successivi tra i punti rimanenti con probabilità proporzionale alla sua distanza al quadrato rispetto al centro del cluster più vicino:

$$p\left(\vec{\mu_t} = \vec{x_n}\right) = \frac{D_{t-1}(\vec{x_n})}{\sum_{n'=1}^{N} D_{t-1}(\vec{x_n'})} \qquad D_t(\vec{x}) = \min_{k=1}^{t-1} \|\vec{x} - \vec{\mu_k}\|_{l_2}^2$$

In tal modo, punti distanti dal centroide hanno maggiore probabilità di essere selezionati.

SELEZIONE DEL NUMERO DI CLUSTERS K

Un metodo euristico per scegliere il numero di clusters fa uso dei SILHOUETTE COEFFICIENTS. Essi sono definiti come

$$SC(i) = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- a_i distanza media dagli altri punti dello stesso cluster.
- b_i distanza media dei punti nel cluster più vicino.

$k_i = \arg\min_k \|\vec{\mu_k} - \vec{x_i}\|$ a_i misura la compattezza di un cluster

$k_i' = \arg\min_{k_2 \neq k_i} \|\vec{\mu_k} - \vec{x_i}\|$ b_i misura la distanza tra clusters

$$-1 < SC(i) < +1$$

Un valore +1 denota che un punto è vicino a tutti i punti nel medesimo cluster e lontano dagli altri clusters. Un valore 0 denota che un punto è vicino al confine del cluster. Un valore di −1 denota che il punto è nel cluster sbagliato. Definiamo silhouette score la media di $SC(i)$ su punti del

cluster

$$SS = \sum_{i=1}^{c} SC(i) .$$

Il numero ottimale di cluster K massimizza SS.