Esame di Ingegneria della Conoscenza

A.A. 2020/2021

Ivano Cinquepalmi

Matricola 661075

Repository

Cos’è il k-means?

Il K-Means è un algoritmo di **apprendimento non supervisionato** che trova un numero fisso di cluster in un insieme di dati.

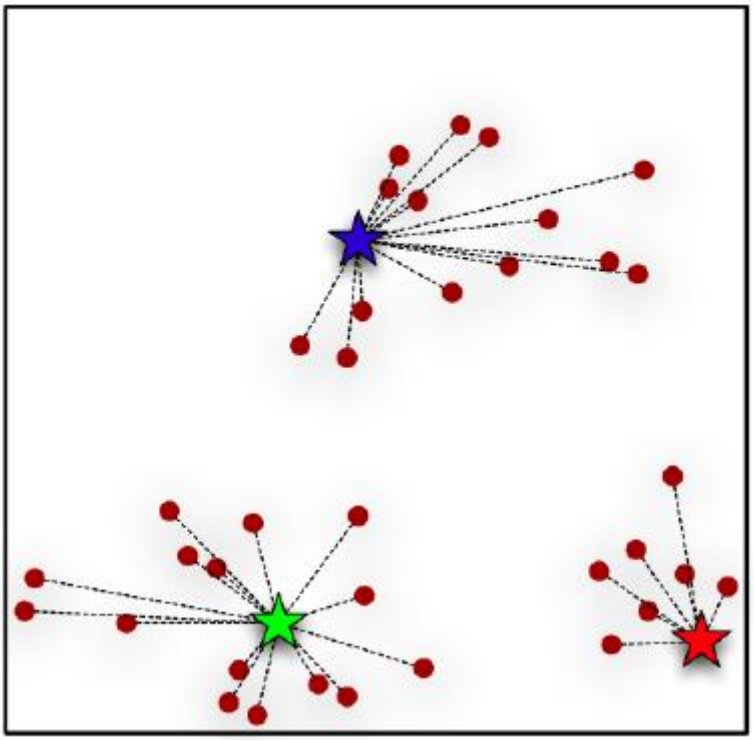
I **cluster** rappresentano i gruppi che dividono gli oggetti a seconda della presenza o meno di una certa somiglianza tra di loro, e vengono scelti a priori, prima dell’esecuzione dell’algoritmo.

Ognuno di questi cluster raggruppa un particolare insieme di oggetti, che vengono definiti **data points**.

L’insieme dei data points analizzati definisce il set di dati, che rappresenta l’insieme di tutte le istanze analizzate dall’algoritmo.

Quando si utilizza un algoritmo K-Means, per ogni cluster si definisce un **centroide**, ossia un punto (immaginario o reale) al centro di un cluster.

Nell’immagine sotto il centroide è rappresentato dalle tre stelle colorate di blu, rosso e verde, mentre i data points sono gli elementi che compongono i cluster, ossia i puntini rossi vicino alle stelle.



Principio di funzionamento del k-means

L'algoritmo k-means è un algoritmo **iterativo**, ossia che esegue ripetutamente alcune sue fasi e fondamentalmente si può affermare che è formato dai seguenti step:

Step 1: Inizializzazione

Per partire occorre inizializzare il k-means.

Lo si fa scegliendo l'ampiezza del set di dati e k centroidi iniziali disposti casualmente.

Scegliendo il numero di centroidi, si scelgono i cluster cui il data set sarà composto e quindi i raggruppamenti che si vogliono effettuare e visualizzare.

Step 2: Assegnazione del cluster

In questa fase, l'algoritmo analizza ciascuno dei data points e li assegna al centroide più vicino.

Quindi viene calcolata la distanza euclidea tra ogni data points e ogni centroide. Ogni data points sarà poi assegnato al centroide la cui distanza risulti minima.

Step 3: Aggiornamento della posizione del centroide

Dopo il passaggio 2 è probabile che si siano formati nuovi cluster, in quanto a quelli precedenti si saranno assegnati (o tolti a seconda che essi siano passati ad un altro cluster) nuovi data points. Di conseguenza, si ricalcola la posizione media dei centroidi. Il nuovo valore di un centroide sarà la media di tutti i data points che sono stati assegnati al nuovo cluster.

Si continuerà a ripetere i passaggi 2 e 3 finché i centroidi non si modificano, ossia si raggiunge un punto di convergenza tale per cui non si hanno più modifiche dei cluster.

Cos’è il K-Nearest Neighbors?

KNN è un algoritmo di **apprendimento supervisionato**, il cui scopo è quello di predire una nuova istanza conoscendo i data points che sono separati in diverse classi. Il suo funzionamento si basa sulla **somiglianza** delle caratteristiche: più un’istanza è vicina a un data point, più il knn li considererà simili. Solitamente la **somiglianza** viene calcolata tramite la **distanza euclidea**.

Minore sarà la distanza e maggiore sarà la somiglianza tra data point e l’istanza da prevedere. Oltre alla distanza, l’algoritmo prevede di fissare un parametro **K**, scelto arbitrariamente, che identifica il numero di data points più vicini. L’algoritmo valuta le k minime distanze così ottenute. La classe che ottiene il maggior numero di queste distanze è scelta come previsione.

Come funziona?

Il funzionamento dell’algoritmo K-Nearest Neighbors può essere definito tramite i seguenti step:

1. Scegli un valore k con cui prevedere il nuovo data point;
2. Nel caso di grandezze non comparabili utilizza tecniche per rendere le misure confrontabili, come la normalizzazione,altrimenti passa allo step 3;
3. Calcola la distanza (ad esempio quella euclidea) tra la nuova istanza e i vari data points;
4. Ordina le distanze calcolate dalla più piccola alla più grande;
5. Scegli le prime K: nel caso si stia svolgendo un problema di regressione, si può restituire la media delle etichette K. Se si sta svolgendo un problema di classificazione, si sceglierà la classe che include più valori k trovati precedentemente.

Esempi di applicazione

Entrambi gli algoritmi sono stati implementati in C++.

I dataset per il Kmeans sono formattati come richiesto dal programma.

Spiegazione della formattazione:

Prima riga: A B C D E

“A” è il numero dei data points.

“B” è il numero delle features.

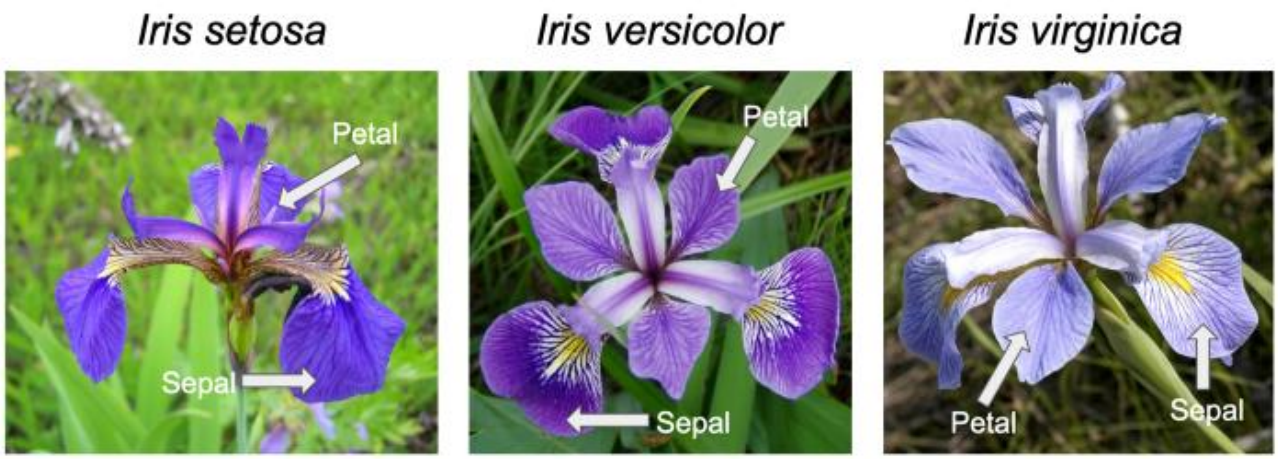
“C” è il numero dei clusters.

“D” è il numero massimo di iterazioni.

“E” è 0 (se non contiene) o 1 (se contiene) un nome per ogni data point.

Iris Dataset

Il **dataset Iris** è un dataset multivariato introdotto da Ronald Fisher nel 1936. Consiste in 150 istanze di Iris misurate da Edgar Anderson e classificate secondo tre specie: Iris setosa, Iris virginica e Iris versicolor. Le quattro feature considerate sono la lunghezza e la larghezza del sepalo e del petalo (in cm).



Numero di data points: 150 (50 per ogni specie).

Feature:

1. Lunghezza del sepalo in cm
2. Larghezza del sepalo in cm
3. Lunghezza del petalo in cm
4. Larghezza del petalo in cm

(5). Classe della specie: setosa, virginica o versicolor.

Esempio dell'iris dataset (Kmeans):

150 4 3 100 1

5.1 3.5 1.4 0.2 Iris-setosa

4.9 3.0 1.4 0.2 Iris-setosa

4.7 3.2 1.3 0.2 Iris-setosa

4.6 3.1 1.5 0.2 Iris-setosa

(...)

Nel Knn, dei 150 set di dati, 100 set (il cui numero di riga NON è divisibile per 3) vengono selezionati come set di dati di addestramento e i restanti 50 set vengono utilizzati come set di dati di prova. La metrica di valutazione utilizzata è la precisione.

Wine Dataset

Wine è un set di dati prodotto da un gruppo di ricerca di Genova. I dati (178) inclusi in questa raccolta sono i risultati di un'analisi chimica dei vini coltivati in una stessa regione d'Italia. Ma i campioni provengono da 3 coltivazioni diverse. L'analisi ha determinato le quantità di 13 costituenti trovati in ciascuno di loro.

Feature:

1. Alcol
2. Acido malico
3. Cenere
4. Alcalinità di cenere
5. Magnesio
6. Fenoli totali
7. Flavanoidi
8. Fenoli nonflavanoidi
9. Proantocianina
10. Intensità del colore
11. Tonalità
12. OD280 / OD315 di vini diluiti
13. Prolina

(14). Tipo di coltivazione

Links:

https://archive.ics.uci.edu/ml/datasets/Iris

https://archive.ics.uci.edu/ml/datasets/Wine