31 ottobre 2023

Clustering - Unsupervised Learning

Learning non supervisionato, non ho etichette, bensì partiziono in gruppi di punti simili. Simili come?

Clustering

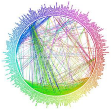
- **Goal**: Automatically partition unlabeled data into groups of similar datapoints.
- **Question**: When and why would we want to do this?
- **Useful for**:
  - Automatically organizing data.
  - Understanding hidden structure in data.
  - Preprocessing for further analysis.
  - Representing high-dimensional data in a low-dimensional space (e.g., for visualization purposes)

Punti vicini tra loro sono simili. A che ci serve? Per capire se i dati hanno struttura nascosta, per organizzarli.

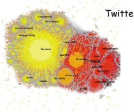
Applications (Clustering comes up everywhere…)

- Cluster users of social networks by interest (community detection).

interazioni tra utenti



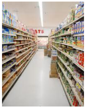
Facebook network



Twitter Network

- Cluster customers according to purchase history.

comprano stesse cose

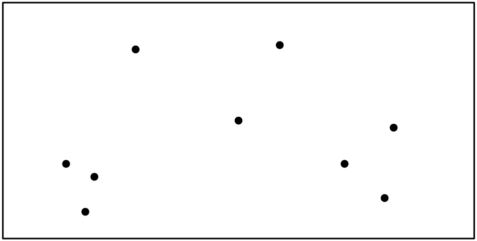


- Cluster galaxies or nearby stars (e.g. Sloan Digital Sky Survey)

- …

Clustering problem

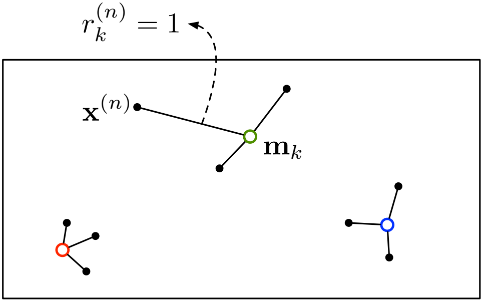
Esistono diversi modi per definire "K", a volte lo so già, a volte no! Slide dopo vediamo come trovarlo!



- Assume the data $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ lives in a Euclidean space, $\mathbf{x}^{(n)} \in \mathbb{R}^D$. sono istante x, ma non abbiamo 't' associati!
- Assume each data point belongs to one of K clusters K hyperparametro
- Assume the data points from same cluster are similar, i.e. close in Euclidean distance.
- How can we identify those clusters (data points that belong to each cluster)? Let's formulate as an optimization problem.

K-means Objective

Ad ogni Cluster corrisponde un punto/centro "m_k" del Cluster, e quali punti sono associati a tale cluster.



- K-means Objective: Find cluster center $\{\mathbf{m}_k\}_{k=1}^{K}$ and assignments $\{\mathbf{r}^{(n)}\}_{n=1}^{N}$ to minimize the sum of squared distances of data points $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$ to their assigned centers.
  - Data sample $n = 1, \ldots, N : \mathbf{x}^{(n)} \in \mathbb{R}^D$ (observed),
  - Cluster center $k = 1, \ldots, K : \mathbf{m}_k \in \mathbb{R}^D$ (not observed),
  - Cluster assignment for sample $n : \mathbf{r}^{(n)} \in \mathbb{R}^K$, 1-of-K encoding (not observed),

K-means Objective

anche i centri sono da trovare, oltre ai punti da associarvi

- K-means Objective: Find cluster center $\{\mathbf{m}_k\}_{k=1}^{K}$ and assignments $\{\mathbf{r}^{(n)}\}_{n=1}^{N}$ to minimize the sum of squared distances of data points $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$ to their assigned centers.
  - Data sample $n = 1, \ldots, N : \mathbf{x}^{(n)} \in \mathbb{R}^D$ (observed),
  - Cluster center $k = 1, \ldots, K : \mathbf{m}_k \in \mathbb{R}^D$ (not observed),
  - Cluster assignment for sample $n : \mathbf{r}^{(n)} \in \mathbb{R}^K$, 1-of-K encoding (not observed),
- Formulated as an optimization problem: ottimizzazione mista non lineare.

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \mathcal{J}(\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}) = \min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2$$

where $r_k^{(n)} = \mathbb{I}[\mathbf{x}^{(n)}$ is assigned to cluster $k]$, i.e., $\mathbf{r}^{(n)} = [0, \ldots, 1, \ldots, 0]^\top$ il primo val è cluster 0, il secondo cluster 1,...

= il punto n-esimo lo assegno al cluster k.

- Finding an optimal solution is an NP-hard problem!

K-means Objective

- Optimization problem:

somma dei quadrati delle distanze tra il centro del cluster e un punto appartenente

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N} \underbrace{\sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2}_{\substack{\text{distance between } x^{(n)} \\ \text{and its assigned cluster center}}}$$

- Since $r_k^{(n)} = \mathbb{I}[\mathbf{x}^{(n)}$ is assigned to cluster $k]$, i.e., $\mathbf{r}^{(n)} = [0, \ldots, 1, \ldots, 0]^\top$ the inner sum has only non zero term
  - e.g., say sample $\mathbf{x}^{(n)}$ is assigned to cluster $k = 3$, then

$$\mathbf{r}^{(n)} = [0, 0, 1, 0, \ldots]$$

and

$$\sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2 = \parallel \mathbf{m}_3 - \mathbf{x}^{(n)} \parallel^2$$

How to optimize?: Alternating Minimization

- Optimization problem:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2$$

- Problem is hard when minimizing jointly over the parameters $\{\mathbf{m}_k\}, \{\mathbf{r}^{(n)}\}$
- But note that if we fix one and minimize over the other, then it becomes easy.
- Doesn't guarantee the same solution!

Invece di ottimizzare i parametri INSIEME, ne fisso uno e minimizzo l'altro.

How to optimize?: Alternating Minimization

- Optimization problem:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2$$

- Note:

l'algoritmo sceglierà a caso questi punti

  - If we fix the centers $\{\mathbf{m}_k\}$ then we can easily find the optimal assignments $\{\mathbf{r}^{(n)}\}$ for each sample n

$$\min_{\{\mathbf{r}^{(n)}\}} \sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2$$

  - Assign each point to the cluster with the nearest center

vedi figura

$$r_k^{(n)} = \begin{cases} 1 & \text{if } k = \arg\min_j \parallel \mathbf{m}_j - \mathbf{x}^{(n)} \parallel^2 \\ 0 & \text{otherwise} \end{cases}$$

  - e.g., if $\mathbf{x}^{(n)}$ is assigned to cluster $\hat{k}$,

$$\mathbf{r}^{(n)} = \underbrace{[0, 0, \ldots, 1, \ldots, 0]^\top}_{\text{Only } \hat{k}-\text{th entry is 1}}$$

How to optimize?: Alternating Minimization

- Likewise, if we fix the assignments $\{\mathbf{r}^{(n)}\}$ then can easily find optimal centers $\{\mathbf{m}_k\}$
  - Set each cluster's center to the average of its assigned data points: For $l = 1, \ldots, K$

$$\frac{\partial}{\partial \mathbf{m}_l} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} \parallel \mathbf{m}_k - \mathbf{x}^{(n)} \parallel^2 = 0$$

lo faccio per i punti assegnati ad un certo "centroide".

$$\frac{\partial}{\partial \mathbf{m}_l} \sum_{n=1}^{N} r_l^{(n)} \parallel \mathbf{m}_l - \mathbf{x}^{(n)} \parallel^2 = 0$$

è il baricentro!

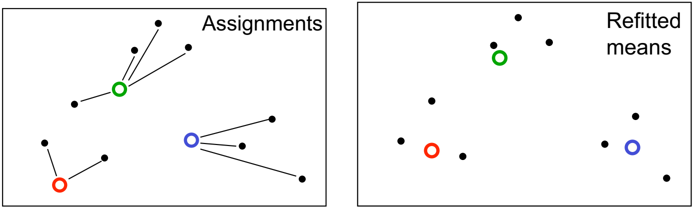
$$2 \sum_{n=1}^{N} r_l^{(n)} (\mathbf{m}_l - \mathbf{x}^{(n)}) = 0 \Longrightarrow \mathbf{m}_l = \frac{\sum_n r_l^{(n)} \mathbf{x}^{(n)}}{\sum_n r_l^{(n)}}$$

- Let's alternate between minimizing $\mathcal{J}(\{\mathbf{m}_k\}, \{\mathbf{r}^{(n)}\})$ with respect to $\{\mathbf{m}_k\}$ and $\{\mathbf{r}^{(n)}\}$
- This is called alternating minimization

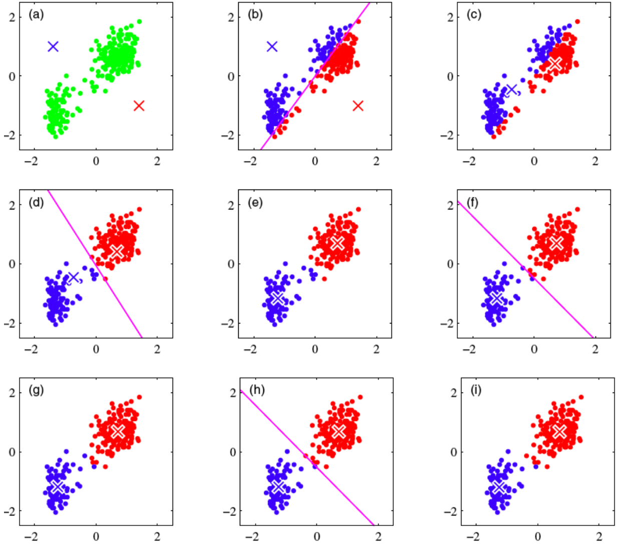
K-means algorithm

High level algorithm

- Inizialization: randomly initialize cluster centers
- The algorithm iteratively alternates between two steps:
    - Assignment step: Assign each data point to the closest cluster
    - Refitting step: Move each cluster center to the mean of the data assigned to it



K-means in action



K-means algorithm

- Inizialization: Set K cluster means $\mathbf{m}_1, \ldots, \mathbf{m}_K$ to random values
- Repeat until convergence (until assignments do not change):
  - Assignment step: Optimize $\mathcal{J}$ w.r.t. $\{\mathbf{r}^{(n)}\}$: Each data point $\mathbf{x}^{(n)}$ is assigned to the nearest center

$$\hat{k}^{(n)} = \arg\min_k \| \mathbf{m}_k - \mathbf{x}^{(n)} \|^2$$

$$r_k^{(n)} = \mathbb{I}[k = \hat{k}^{(n)}] \text{ for } k = 1, \ldots, K$$

  - Refitting step: Optimize $\mathcal{J}$ w.r.t. $\{\mathbf{m}\}$: Each center is set to mean of data assigned to it

$$\mathbf{m}_l = \frac{\sum_n r_l^{(n)} \mathbf{x}^{(n)}}{\sum_n r_l^{(n)}}$$

Questions about K-means

li assegno al baricentro, perchè vogliamo minimizzare il quadrato della distanza, quadrati "buoni" perchè derivate semplici. Se avessi un'altra funzione obiettivo non varrebbe!

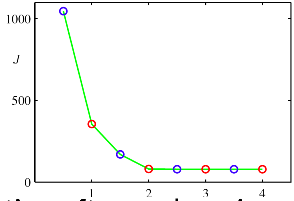
- Why does update set $\mathbf{m}_k$ to mean of assigned points?
- What if we used a different distance measure? cambia tutto!
- How can we choose the best distance? dipende dal concetto di "miglior distanza". Quadrato della distanza semplice, ma non è detto che sarà sempre la scelta giusta.
- How to choose K?
- Will it converge?

L'algoritmo converge sempre, per la scelta di K, vediamo cosa possiamo fare.

Why K-means Converges

La funzione obiettivo migliora e si riduce.

- K-means algorithm reduces the cost at each iteration.
  - Whenever an assignment is changed, the sum squared distances $\mathcal{J}$ of data points from their assigned cluster centers is reduced.
  - Whenever a cluster center is moved, $\mathcal{J}$ is reduced.
- Test for convergence: If the assignments do not change in the assignment step, we have converged (to at least a local minimum).
- This will always happen after a finite number of iterations, since the number of possible cluster assignments is finite



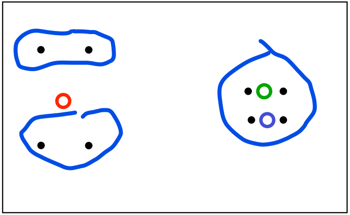
- K-means cost function after each assignment step (blue) and refitting step (red). The algorithm has converged after the third refitting step.

Local Minima

con 3 cluster la soluzione migliore prevede i tre raggruppamenti in blu. Tuttavia l'algoritmo lavora su minimo locale, e fornisce i tre punti rossi, verdi e blu, che però sono peggiori di quelli trovati da me!

- The objective $\mathcal{J}$ is non-convex
- There is nothing to prevent k-means getting stuck at local minima.
- We could try many random starting points

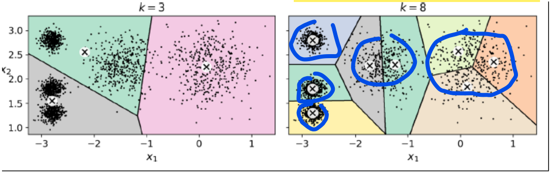
A bad local optimum



Determine the optimal value of K

Elbow method:

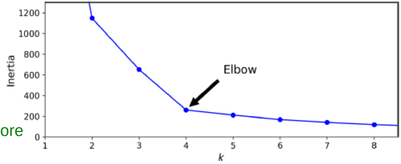
- run K-means plot cost function for different value of K



qui, vedendo i cluster, possiamo pensare a k = 5

- As K increases points will get closer to the centroids $\Longrightarrow$ the cost function decreases
- **Idea** Choose the point where the error decreases the most before slowing down

dopo "4" la decrescita rallenta di molto.



sulla y abbiamo l' errore

- Coarse method wrt, e.g., *silhoutte score*

Qui non abbiamo il concetto di etichette, non c'è nulla che ci dica se stiamo facendo bene o male.