Lecture 3: K - Means

1. Overview

- Hai bài toán lớn chủ đạo trong Unsupervised Learning là
  - +) Dimensional reduction.
  - +) Clustering
- Hiện nay, các large pretrained model cho kqua embedding khá tốt, biểu diễn dc tốt cấu trúc dữ liệu, tạo điều kiện cho những thuật toán như clustering perform tốt trên dữ liệu ko nhãn
- Khi cần so sánh một điểm mới thuộc cụm nào, ta chỉ cần so sánh điểm đó vs đại diện của các cụm để xem cái nào giống nhất.

2. K-Means Algorithm

- In clustering problem, we are given a training set $\{x_1, x_2 \dots x_n\}$ and want to group data into a few "cohesive" clusters.
  Note that $x_i \in \mathbb{R}^d$ and label y_i is not given.
- Mỗi cluster có một tham số đại diện, chính là centroid (điểm trung tâm) của từng cluster. Ký hiệu: $\mu_1, \mu_2, \dots \mu_K \rightarrow$ g/sử K clusters

Read more: Khái niệm hard vs soft clustering:
  - +) Hard clustering: mỗi điểm được assign rõ chỉ thuộc một cluster nào đó. $\Rightarrow$ K-means là hard clustering.
  - +) Soft clustering: một điểm có thể thuộc nhiều cluster dựa trên probability điểm đó thuộc cluster nào. VD: GMM...

- Ta ký hiệu $r_i \begin{cases} = 1 & \text{nếu } x_i \in C_K \text{ (thuộc cluster K)} \\ = 0 & \text{nếu } x_i \notin C_K \end{cases}$
- Khi xét một điểm thuộc một cluster nào đó, ta mong muốn k/cách từ điểm đó đến tâm của cluster là nhỏ nhất. Về mặt toán, đây là việc tìm μ_i và r_i tương ứng để minimize distortion measure:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{K} r_{ik} \| x_i - \mu_K \|^2$$

$\rightarrow$ vì trong các gtrị của K, chỉ có một r_{iK} có gtri bằng 1, còn lại bằng 0 nên cthức chứa k/c của từng điểm đến centroid của chúng rồi cộng tổng chúng lại $\Rightarrow$ K-means minimize tổng k/c của điểm đến centroid

- Hai bộ tham số cần tìm $\{\mu_1, \dots \mu_K\}$ và $\{r_1, \dots r_n\}$ ko thể được giải bằng cách đạo hàm ra nghiệm hay GD đồng thời 2 biến vì chúng phụ thuộc lẫn nhau.

Vậy nên quá trình optimize J sẽ được chia thành các bước như sau:

+) Khởi tạo một vector μ với các gtri $\{\mu_1 \dots \mu_K\}$ bất kỳ.

+) Step 1: Biết μ_K → minimize loss w.r.t r_{ik}

+) Step 2: Biết r_{ik} → ______________ μ_K

+) Repeat two above steps until convergence.

(*) <u>Chi tiết từng bước</u>

- Step 1: Fix μ tìm r $\quad J_i = r_{ik} \| x_i - \mu_K \|_2^2$ → k/c từ điểm x_i đến 1 centroid ✓

  ⇒ Khi đã biết tất cả các $\mu_1, \dots \mu_K$, ta tính k/c từ x_i đến tất cả các centroid rồi chọn $r_{ik} = 1$ với k/c nhỏ nhất.

- Step 2: Fix r tìm μ $\quad J_K = \sum_{i=1}^{n} r_{ik} \| x_i - \mu_K \|_2^2$

$$\Leftrightarrow \frac{\partial J_K}{\partial \mu_K} = -2 \sum_{i=1}^{n} r_{ik} (x_i - \mu_K) = 0 \Leftrightarrow \boxed{\mu_K = \frac{\sum_{i=1}^{n} r_{ik} \cdot x_i}{\sum_{i=1}^{n} r_{ik}}}$$

→ Trong ctức μ_K trên — tử số là Σ gtri các điểm thuộc cluster K
— mẫu số là Σ số điểm thuộc ______.

- Convergence là khi update 2 bước trên, thay đổi cũng ko đáng kể nữa.

③ Một số vde của k-means.
- phải chọn k.
- phải khởi tạo μ_K → hyper param.
- có thể ko converge về local minimum nếu chọn sai hyper parameter.

(*) Solution:

1) Chọn k → nhiều p.p chọn nên ko fatal lắm. VD: Link

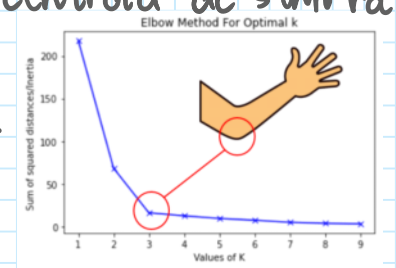
2) Khởi tạo μ_K: là bước khó nhất và chính là main drawback của k-means bởi nếu khởi tạo tệ → clustering tệ.

   → Một số kỹ thuật như k-means ++, k-means centroid đc sinh ra để cải tiến khâu khởi tạo μ.

3) Việc hội tụ hay ko hội tụ thì ko can thiệp đc.

(4) Choosing k methods

1) Elbow method



Line plot between K and inertia

- Performing kmeans on the dataset multiple time with different k and calculate WSS (within-cluster-sum of square) (k/c của các điểm đến centroid của nó). ⇒ plot the curve for k and wss.

- Choose k for which wss first starts to diminish. In the plot wss-vs k, it is visible as an elbow.

2) Silhouette Analysis

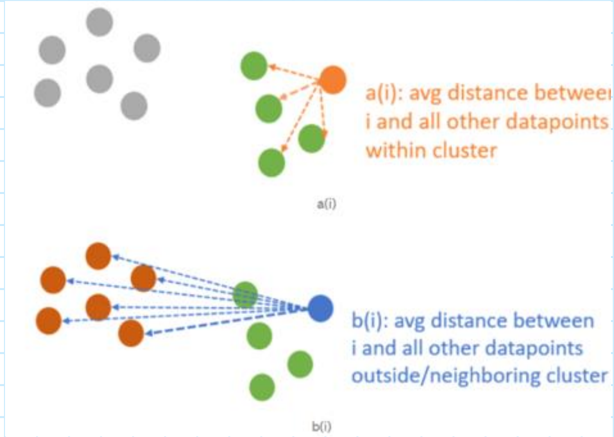
- The Silhouette coefficient or Silhouette Kmeans score is a measure of how similar a datapoint is within - cluster (cohesion) compared to other clusters (separation).

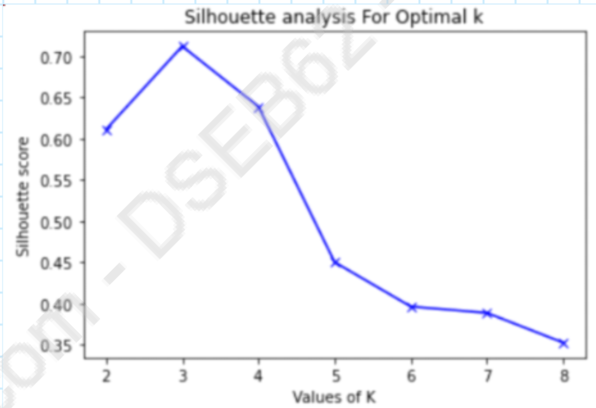
Formula: $S(i) = \dfrac{b(i) - a(i)}{\max\{a(i), b(i)\}}$

- $S(i)$ is Silhouette coefficient for point i.
- $a(i)$ is the average distance of point i to other points of the cluster where it belongs.
- $b(i)$ is the avg distance from i to other clusters.

⇒ we want $S(i)$ to be as max as possible.

(*) Minh họa:





(*) Plotting Silhouette-versus-k

⑤ Kmeans pros and cons

Advantages of k-means clustering

- **Many common implementations**. One of the main advantages of k-means clustering is that it has many common implementations across a variety of different machine learning libraries. No matter what language or library you are using to implement your clustering model, k-means is the most likely clustering model to be available. In some cases, k-means clustering may even be the only option that is available.
- **Popular and well studied**. The reason that k-means clustering has so many implementations across a variety of languages and libraries is that it is probably the most popular and well-studied clustering algorithm out there. This popularity confers some benefits of its own, as it will make it easier for other contributors to jump in to assist or even take over an ongoing project. If the model is going to be used to score data repeatedly, using a well studied algorithm will also reduce the burden of maintenance.
- **Comparatively fast**. While clustering algorithms are known to be relatively slow, the k-means algorithm is comparatively fast. K-means is an iterative algorithm that involves calculating the distance between each point in your data and the center of each cluster. Unlike many other clustering algorithms, it does not require you to calculate the pairwise distance between points in your dataset. That means the performance scales linearly with the number of data points in your dataset.

Disadvantages of k-means clustering

- **Assumes spherical density**. One of the main disadvantages of k-means clustering is that it constrains all clusters to have a spherical shape. This means that k-means clustering does not perform as well in situations where clusters naturally have irregular shapes. This is a relatively strict assumption that is not made by all clustering algorithms.
- **Sensitive to scale**. Since k-means clustering works by calculating the distance between your data points and the size of centers of your clusters, it can be thrown off by situations where your variables have different scales. If one of your variables is on a much larger scale than the others, for example, that variable will have an outsized effect on the distance calculated. This means that you generally need to re-scale your data before using k-means clustering.
- **Difficult to incorporate categorical variables**. As is common with many clustering algorithms, k-means is intended for situations where all of your features are numeric. As such, it does not perform as well in cases where you need to incorporate categorical features in your dataset.
- **Sensitive to outliers**. Unlike some other clustering algorithms that are able to identify and exclude outliers, k-means clustering includes every data point in a cluster. That means that the algorithm is somewhat sensitive to large outliers.

- **Sensitive to choice of seed**. K-means clustering is relatively sensitive to the starting conditions that are used to initialize the algorithm such as the choice of seed or the order of the data points. This means that you may not get the same results if changes are made to the initialization conditions.
- **Have to choose the number of clusters**. Like many other clustering algorithms, k-means clustering requires you to specify the number of clusters that will be created ahead of time. This may be difficult in cases where the true number of clusters is unknown.
- **Struggles with high dimensional data**. Like many other clustering algorithms, k-means clustering starts to struggle when many features are included in the model. If you have many potential features, you should consider applying feature selection or dimensionality reduction algorithms to your data before creating your clusters.