K-Means Algorithm

Dr. Sharu Theresa Jose

University of Birmingham

February 16, 2024

Learning Outcomes

- Understand the basic principles of the most popular partitional clustering algorithm: K-means
- Learn to apply K-means to clustering problems
- Understand the challenges

Overview of Lecture

- Recap: Clustering as an optimization problem
- Introduction to K-Means
- Challenges of K-Means and ways to overcome them
- Application of K-Means: Vector Quantization

Recap: Goal of Partitional Clustering Algorithms

Recall that the partitional clustering problem can be framed as the following optimization problem:

Clustering as an optimization problem

Find a clustering structure $\mathcal{C}$ of K clusters that minimizes the following objective

$$\begin{aligned} & \min_{\mathcal{C}} \mathrm{WCSS}(\mathcal{C}) \\ \text{s.t} \quad & C_1 \cup C_2 \cup \ldots \cup C_K = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \\ & C_k \cap C_{k'} = \emptyset, \quad \forall k \neq k', \end{aligned} \tag{1}$$

where

$$\mathrm{WCSS}(\mathcal{C}) = \sum_{C \in \mathcal{C}} \mathrm{intertia}(C) = \sum_{C \in \mathcal{C}} \sum_{e \in C} d_{\mathrm{Euc}}(e, \mathrm{Centroid}(C))^2.$$

K-Means

- K-means is an iterative, greedy descent algorithm that aims to find a sub-optimal solution to the optimization problem in (1).
- Specifically, K-means iteratively alternate between the following two steps:
  - **Assignment Step**: For a given set of K cluster centroids, K-means assigns each example to the cluster with closest centroid.

    $$\mathrm{WCSS}(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{e \in C} d_{\mathrm{Euc}}(e, \underbrace{\mathrm{Centroid}(C)}_{\text{fixed}})^2.$$

    This is equivalent to fixing the centroids and optimizing the cluster assignment to ensure low WCSS.
  - **Refitting step**: Re-evaluate and update the cluster centroids.

    This is equivalent to fixing the cluster assignment and optimizing the centroids to ensure low WCSS.
  - Consequently, over iterations, WCSS continue to decrease.

The Algorithm

K-Means Algorithm

Input: Number K of clusters and N examples $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$.

1. Select K of the N examples at random as centroids and label it as $c_1, \ldots, c_K$.
2. Repeat until cluster centroids do not change:
   - (Assignment Step) Form K clusters by assigning each observation to its closest cluster centroid, i.e.,

$$\text{Cluster}(\mathbf{x}^{(i)}) = \arg \min_{k=1,\ldots,K} d_{\text{Euc}}(\mathbf{x}^{(i)}, c_k)^2, \quad \text{for } i = 1, \ldots, N$$

   - (Refitting Step) Compute the centroid of the obtained K clusters as

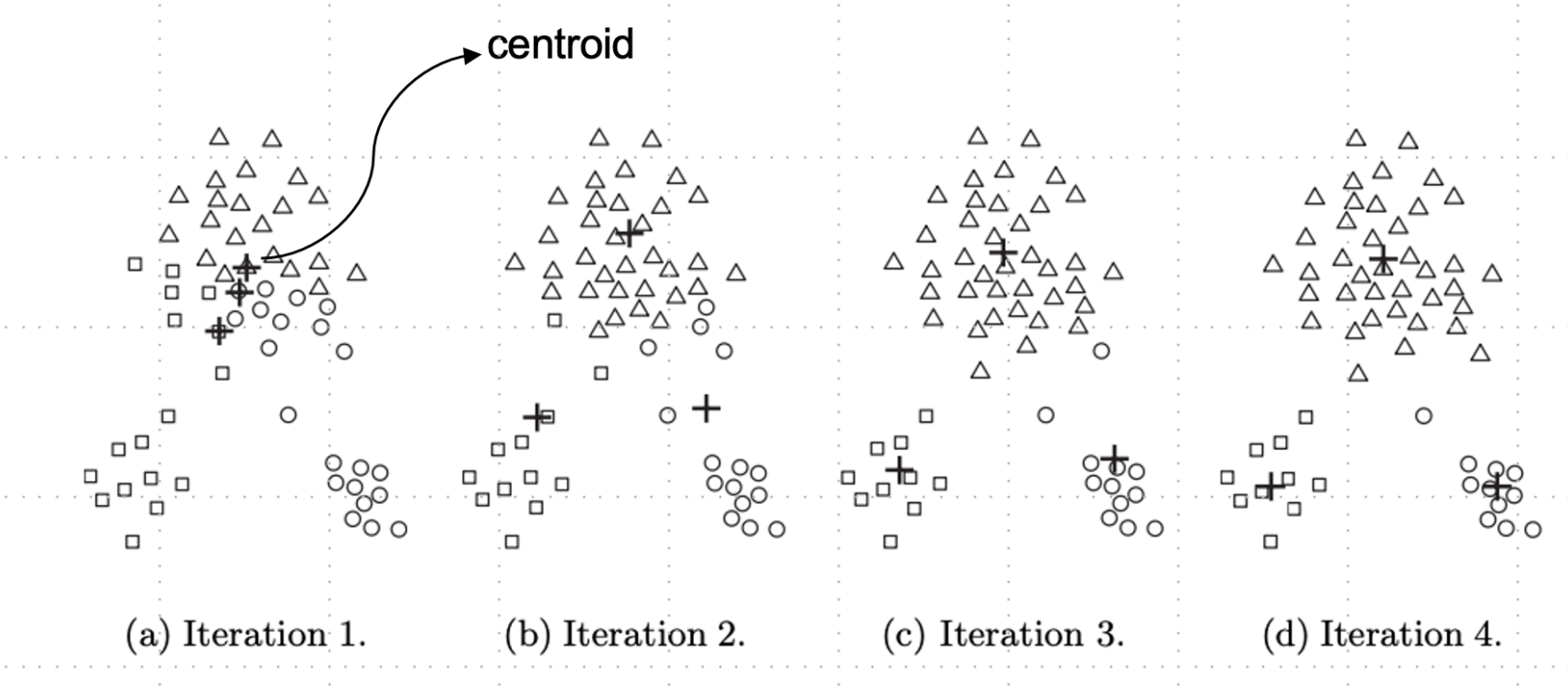
$$c_k = \frac{1}{N_k} \sum_{i:\text{Cluster}(\mathbf{x}^{(i)})=k} \mathbf{x}^{(i)}, \quad \text{for } k = 1, \ldots, K,$$

   where N_k is the total number of examples in the kth cluster.

Points to Note

- Initial choice of cluster centroids are **randomly** chosen from the data examples.
- However, centroids found by K-means over the iterations need not be data examples.
- Cluster assignment is based on squared Euclidean distance.
- Stopping criterion: stop when cluster centroids do not change.

Illustration

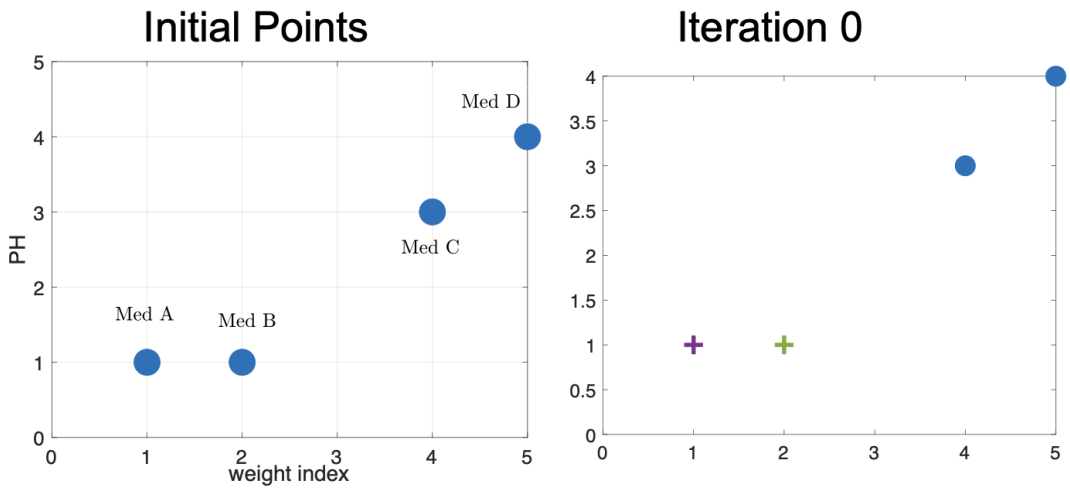


Note that cluster centroids need not be examples in later iterations.

Example 1: Clustering of Medicines

Problem: Use K-means algorithm to cluster the following medicines into $K = 2$ clusters.

	Weight Index	PH
Med A	1	1
Med B	2	1
Med C	4	3
Med D	5	4

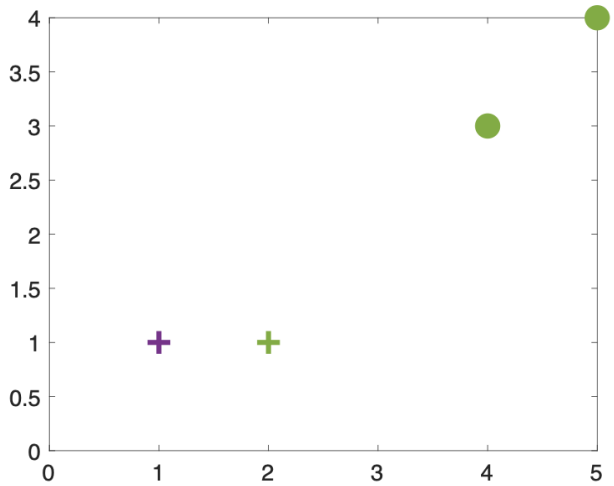


Iteration 0: Let initial centroids of cluster 1 and cluster 2 be chosen as Med A and Med B respectively, i.e., $c_1 = (1, 1)$ and $c_2 = (2, 1)$.

Iteration 1:

1. Calculate the squared Euclidean distance of each point to cluster centroids to form an object-centroid distance matrix:

	Med A	Med B	Med C	Med D
c_1	0		13	25
c_2		0	8	18



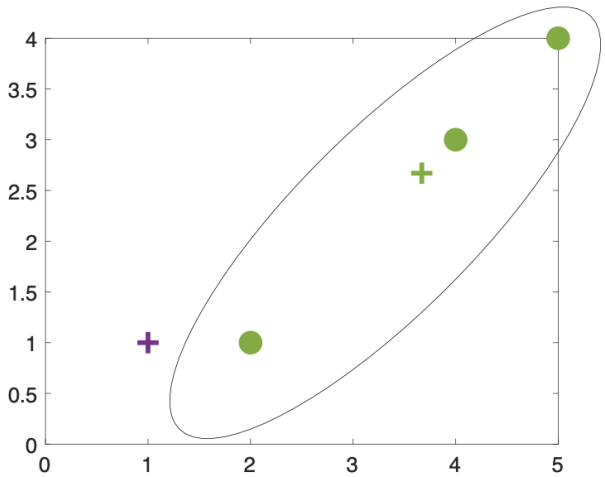
$$d_{\text{Euc}}(MedC, c_1)^2 = (4-1)^2 + (3-1)^2 = 13$$
$$d_{\text{Euc}}(MedC, c_2)^2 = (4-2)^2 + (3-1)^2 = 8$$
$$d_{\text{Euc}}(MedD, c_1)^2 = (5-1)^2 + (4-1)^2 = 25$$
$$d_{\text{Euc}}(MedD, c_2)^2 = (5-2)^2 + (4-1)^2 = 18$$

Based on the matrix above, assign each medicine to its closest centroid. Thus, medicines B, C and D are assigned to Cluster 2.

2. Update the centroids of the clusters obtained in the previous step. This results in

$$c_1 \leftarrow c_1$$

$$c_2 \leftarrow \frac{1}{3}(\mathrm{MedB} + \mathrm{MedC} + \mathrm{MedD}) = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = (3.67, 2.67).$$



Iteration 2:

1. Calculate distance of each point to new cluster centroids.

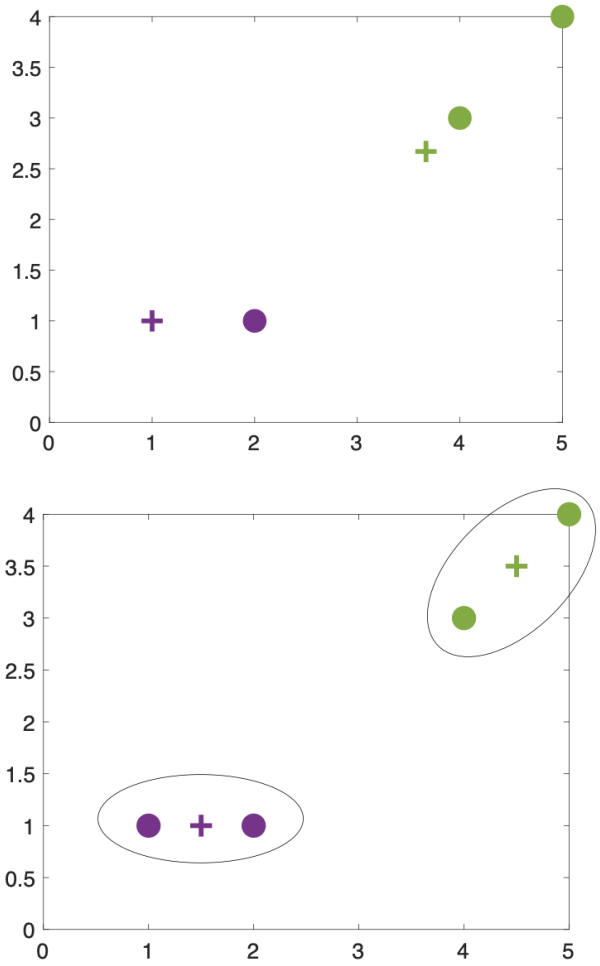
	Med A	Med B	Med C	Med D
c_1	0	1	13	25
c_2	9.92	5.58	0.22	3.53

Med B is thus moved to cluster 1.

2. Update the centroids of the cluster.

$$c_1 \leftarrow \frac{1}{2}(\text{MedA} + \text{MedB}) = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1.5, 1)$$

$$c_2 \leftarrow \frac{1}{2}(\text{MedC} + \text{MedD}) = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4.5, 3.5).$$



Iteration 3:

- Repeat the same steps as before.
- Note that cluster assignments do not change.
- Algorithm has converged.

Space and Time Complexity of K-Means

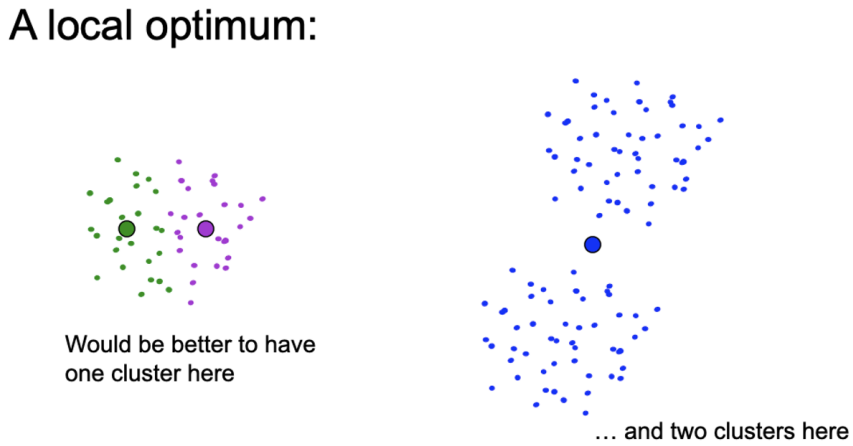
- Space requirement for K-means is modest because only data observations and centroids are stored.
- Storage complexity is of the order $O((N + K)m)$ where m is the number of feature attributes.
- Time complexity of K-means is of the order $O(IKNm)$ where I is the number of iterations required for convergence.
- Importantly, time complexity of K-means is linear in N.

Challenges and Issues in K-Means

- Does the K-means algorithm always converge?
- Can it always find optimal clustering?
- How should we initialize the algorithm?
- How should we choose the number of clusters?

Convergence of K-Means

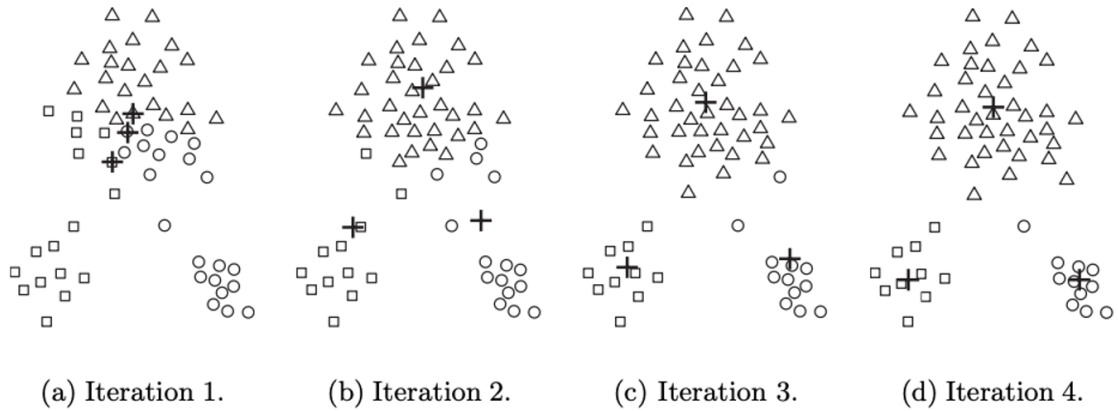
- At each iteration, the assignment and refitting steps ensure that the WCSS in (1) monotonically decreases.
- Moreover, K-means work with finite number (K^N) of partitions of the data.
- The above two conditions ensure that the K-means algorithm always converge.
- However, the optimization problem of (1) is non-convex. As such, K-means algorithm may not converge to the global minimum, but to a local minimum.



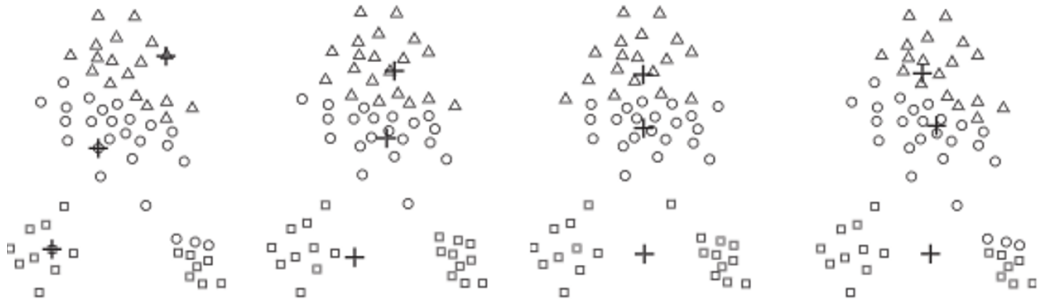
Escaping local minima: multiple random restarts and choose the clustering with lowest WCSS.

Choice of Initial Cluster Centroids

- Choosing initial cluster centroids is crucial for K-means algorithm.
- Different initializations may lead to convergence to different local optima.
- K-means is a **non-deterministic** algorithm.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

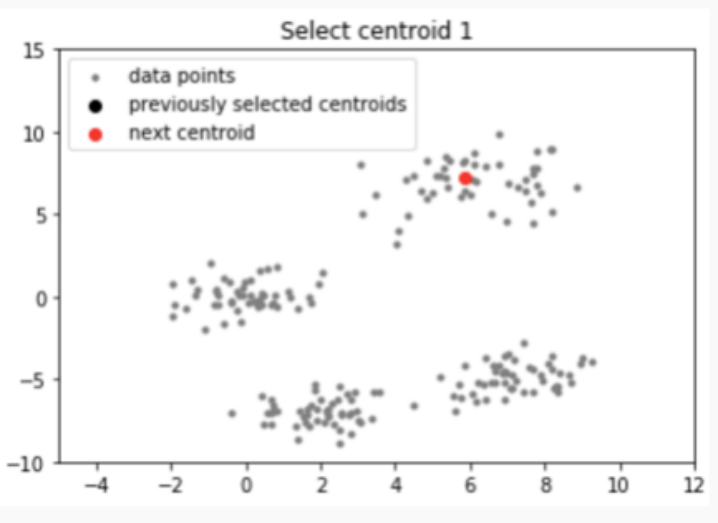
Solutions

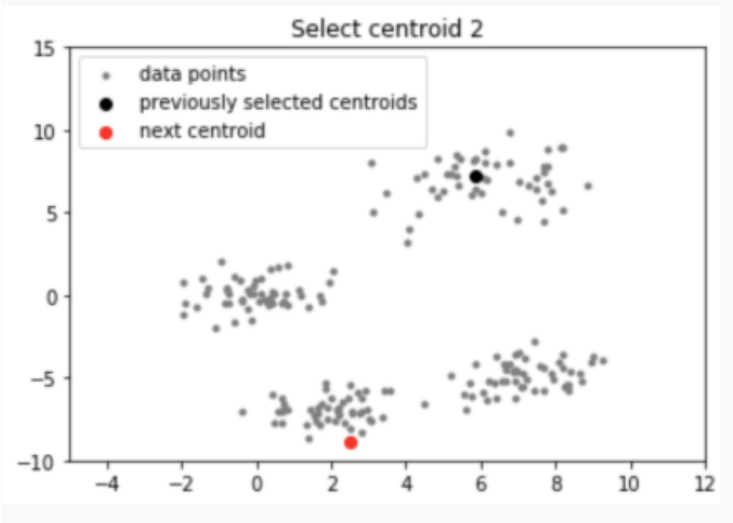
- Run multiple K-means algorithm starting from randomly chosen cluster centroids. Choose the cluster assignment that has the minimum WCSS.
- Specialized initialiation strategies: K-means ++

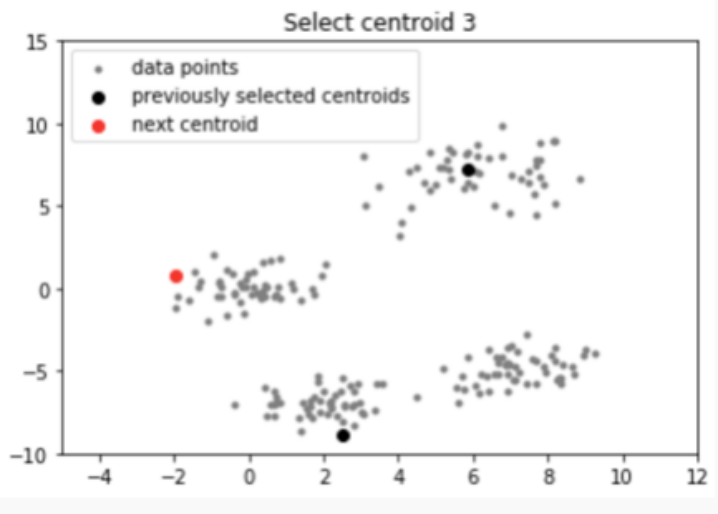
K-means ++ Algorithm

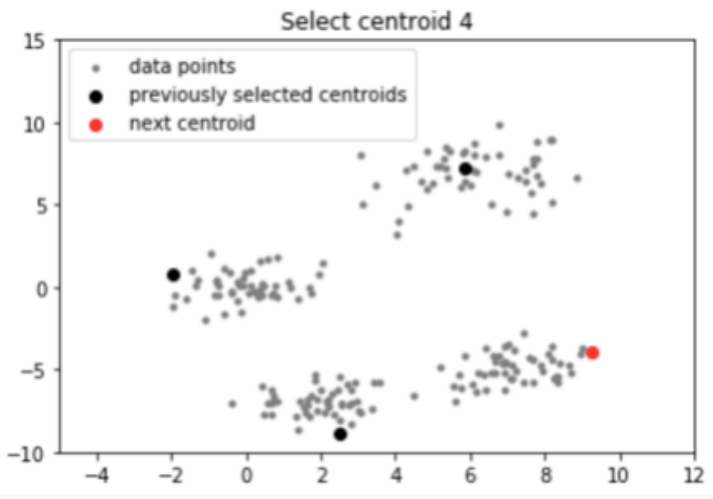
- Choose first centroid c_1 at random.
- Repeat until K centroids are found:
  - For each data point $\mathbf{x}$, compute the distance $d_{\mathrm{Euc}}(\mathbf{x}, c) = d(\mathbf{x})$ from its nearest centroid c.
  - Choose a new data point $\mathbf{x}$ randomly with probability proportional to $d(\mathbf{x})^2$ as the next centroid. (Thus, a data point $\mathbf{x}$ that is farthest from the nearest centroid is highly likely to be picked as the next centroid)
- Use the obtained K centroids as initial centroids for the K-means algorithm.
- Run the K-means.

Illustration of K-Means++









PC: geeksforgeeks.org

Choice of the Number K of Clusters

- Conventional approach: Use prior domain knowledge
  - Example: Data segmentation - a company determines the number of clusters into which its employees must be segmented.
- A data-based approach for estimating the number of natural clusters in the dataset: **Elbow Method**

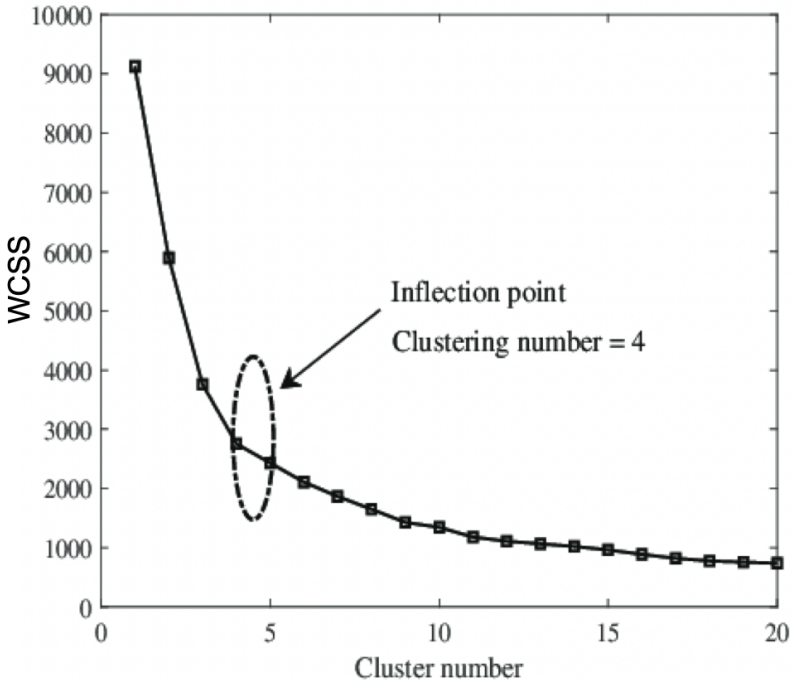
Elbow Method

- Run K-means algorithm repeatedly for increasing values of K.
- Evaluate the WCSS of the obtained clustering structure in each run of K-means.
- Plot WCSS($\mathcal{C}$) as a function of the number K of clusters.
- Optimal K lies at the elbow of the plot.

Some drawbacks of the elbow method:

- Heuristic concept
- Not easy to identify the elbow

Elbow Method

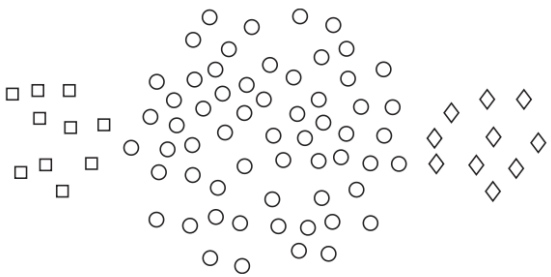


Intuition behind elbow method:

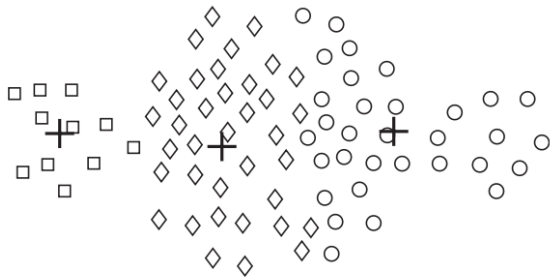
- Assume that there are actually K^* distinct clusters in the observed dataset.
- For $K < K^*$, clusters returned by K-means will each contain a subset of the true underlying clusters.
  - Consequently, WCSS tends to decrease **substantially** with each successive increase in K (until it reaches K^*) as natural groups are successively assigned to separate clusters.
- For $K > K^*$, one of the clusters estimated by K-means must partition at least one of the natural groups into two.
  - This will provide **smaller decrease** in WCSS as K is further increased.
- $K = K^*$ corresponds to the number of clusters where there is sharp decrease in successive differences in WCSS. This point thus corresponds to the elbow or kink in the plot.

Other limitations of K-Means

- K-means has problems when outliers are present. Outliers can influence the clusters that are found and can also increase the WCSS.
- K-means has problems when clusters are of differing
  - Sizes

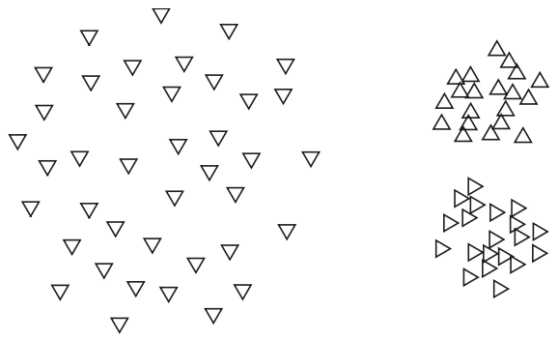


(a) Original points.

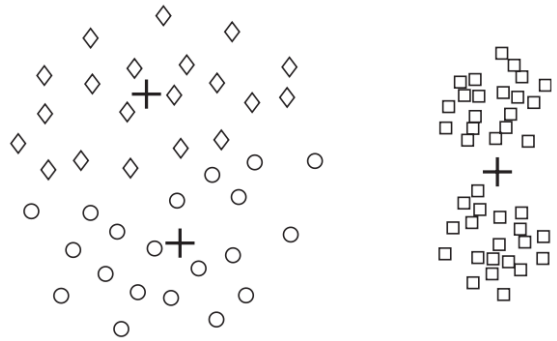


(b) Three K-means clusters.

- K-means has problems when clusters are of differing
  - Densities

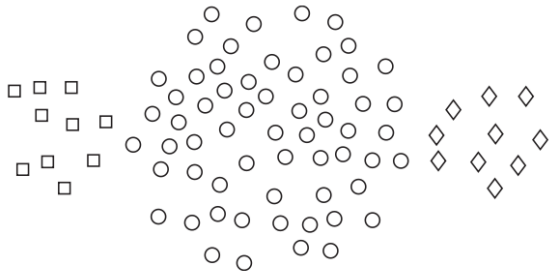


(a) Original points.

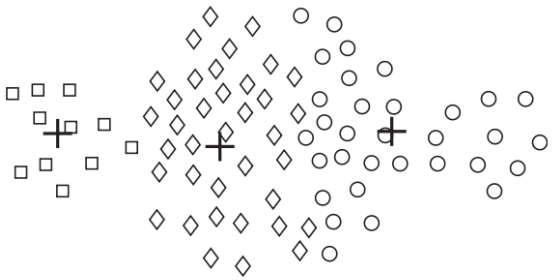


(b) Three K-means clusters.

  - Non-globular shapes



(a) Original points.



(b) Three K-means clusters.

One solution is to find a large number of clusters such that each of them represents a part of a natural cluster. But these small clusters need to be put together in a post-processing step.

Application

A practical application of K-Means

K-means algorithm is a key tool in the area of image and signal compression, particularly in **vector quantization**.

- (Left photo) 1024×1024 pixel image, where each pixel is a greyscale value ranging from 0 to 255. Requires 8 bits per pixel, and 1 megabyte of storage.
- (Center photo) Vector quantization (VQ)-based compressed version of left photo. Requires 0.239 of the storage of the left photo.
- (Right photo) Even more compressed version. Requires only 0.0625 of the storage of the left photo.

Vector Quantization via K-means

- Break left image into small blocks of 2×2. This results in 512×512 blocks of four numbers, each regarded as a (feature) vector in $\mathbb{R}^4$.
- Run K-means algorithm in the resulting space of feature vectors. The clustering process is called the **encoding step** in VQ, and the collection of centroids is called the **codebook**.
- Center image uses $K = 200$ while the right one uses $K = 4$.
- To represent the compressed image, approximate each of the 512×512 blocks by its closest **cluster centroid**, known as **codeword**. Then, construct the image from the centroids, a process called **decoding**.
- Storage requirements:
  - For each block, the identity of the cluster centroid need to be stored.
  - This requires $\log_2(K)$ bits per block, which equals $\log_2(K)/4$ bits per pixel.

Summary

Properties of K-means:

- Optimizes a global objective function
- Based on squared Euclidean distance
- Non-deterministic

Challenges of K-means:

- Requires as input the number K of clusters and an initial choice of centroids.
- Convergence to local minima implies multiple restarts.

References

- "The Elements of Statistical Learning (Data Mining, Inference, and Prediction)", Hastie *et.al.* - Chapter 14.3.6, 14.3.9, 14.3.11
- "Introduction to Data Mining", Tan, Steinbach and Kumar: Chapter 8