Unsupervised Learning
K-MEANS

Dr. Amani RAAD

amani.raad@ul.edu.lb

2022-2023

Supervised learning vs. unsupervised learning

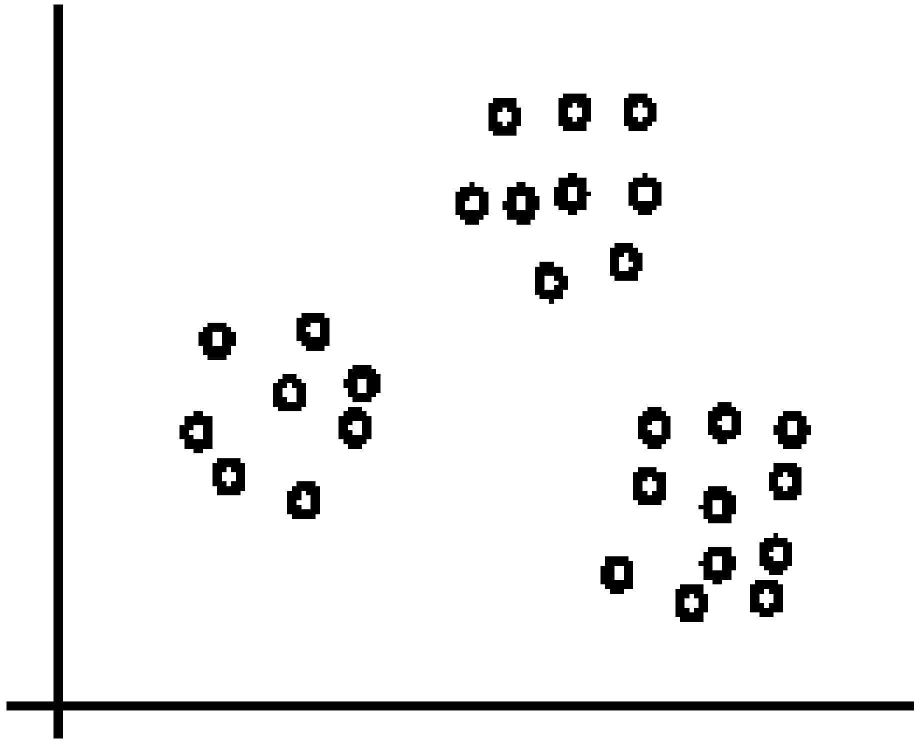
- Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.

Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
    - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
    - In fact, association rule mining is also unsupervised
- This chapter focuses on clustering.

An illustration

- The data set has three natural groups of data points, i.e., 3 natural clusters.



What is clustering for?

- Let us see some real-life examples
- Example 1: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
    - Tailor-made for each person: too expensive
    - One-size-fits-all: does not fit all.
- Example 2: In marketing, segment customers according to their similarities
    - To do targeted marketing.

What is clustering for? (cont...)

- Example 3: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy
- In fact, clustering is one of the most utilized data mining techniques.
  - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - In recent years, due to the rapid increase of online documents, text clustering becomes important.

Aspects of clustering

- A clustering algorithm
  - Partitional clustering
  - Hierarchical clustering
  - …
- A distance (similarity, or dissimilarity) function
- Clustering quality
  - Inter-clusters distance $\Rightarrow$ maximized
  - Intra-clusters distance $\Rightarrow$ minimized
- The quality of a clustering result depends on the algorithm, the distance function, and the application.

K-means clustering

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be

  $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The k-means algorithm partitions the given data into k clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user

K-means algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1) Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
  2) Assign each data point to the closest centroid
  3) Re-compute the centroids using the current cluster memberships.
  4) If a convergence criterion is not met, go to 2).

K-means algorithm – (cont …)

```
    Algorithm k-means(k, D)
1   Choose k data points as the initial centroids (cluster centers)
2   repeat
3       for each data point $\mathbf{x} \in D$ do
4           compute the distance from $\mathbf{x}$ to each centroid;
5           assign $\mathbf{x}$ to the closest centroid        // a centroid represents a cluster
6       endfor
7       re-compute the centroids using the current cluster memberships
8   until the stopping criterion is met
```

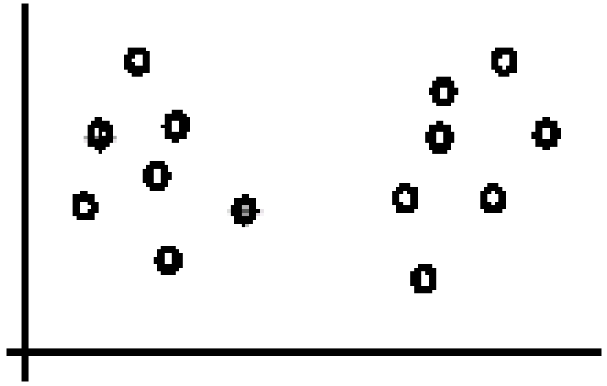
Stopping/convergence criterion

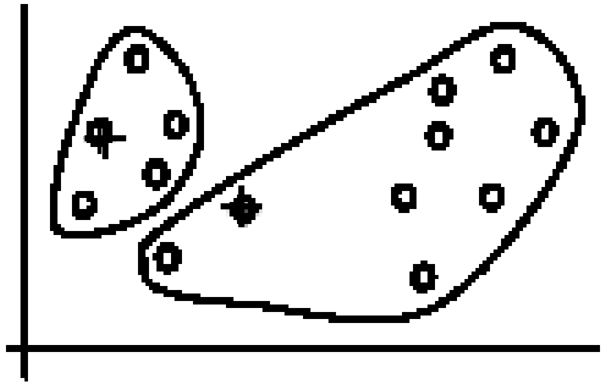
1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE) or WCSS,

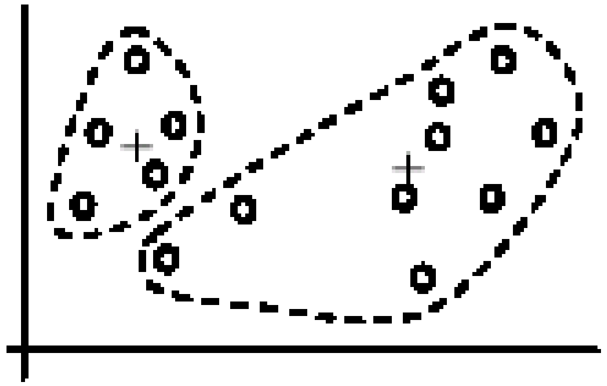
$$WCSS = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2 \qquad (1)$$

- C_i is the *j*th cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point **x** and centroid $\mathbf{m}_j$.

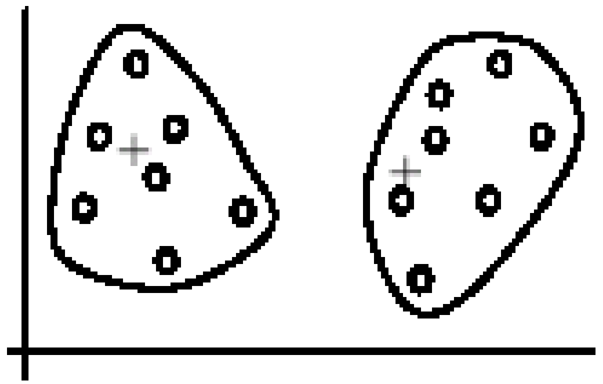
An example

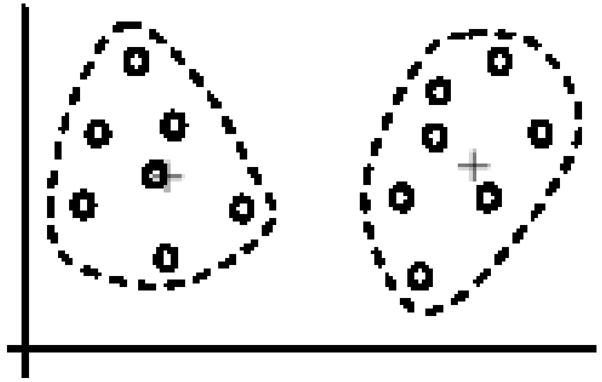

(A). Random selection of *k* centers

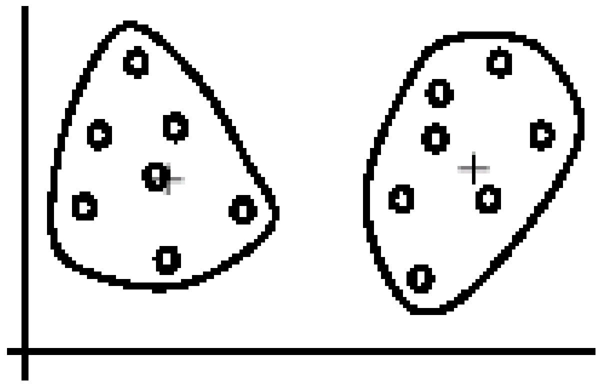

Iteration 1: (B). Cluster assignment

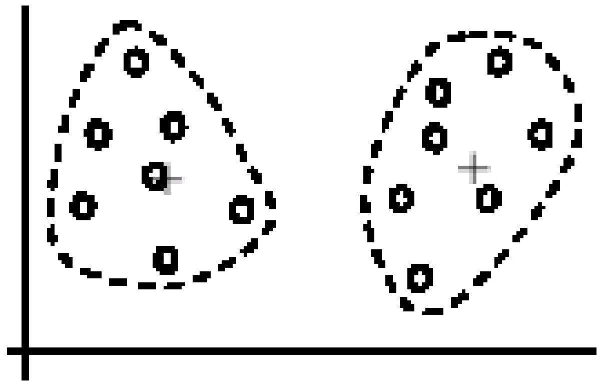

(C). Re-compute centroids

An example (cont …)

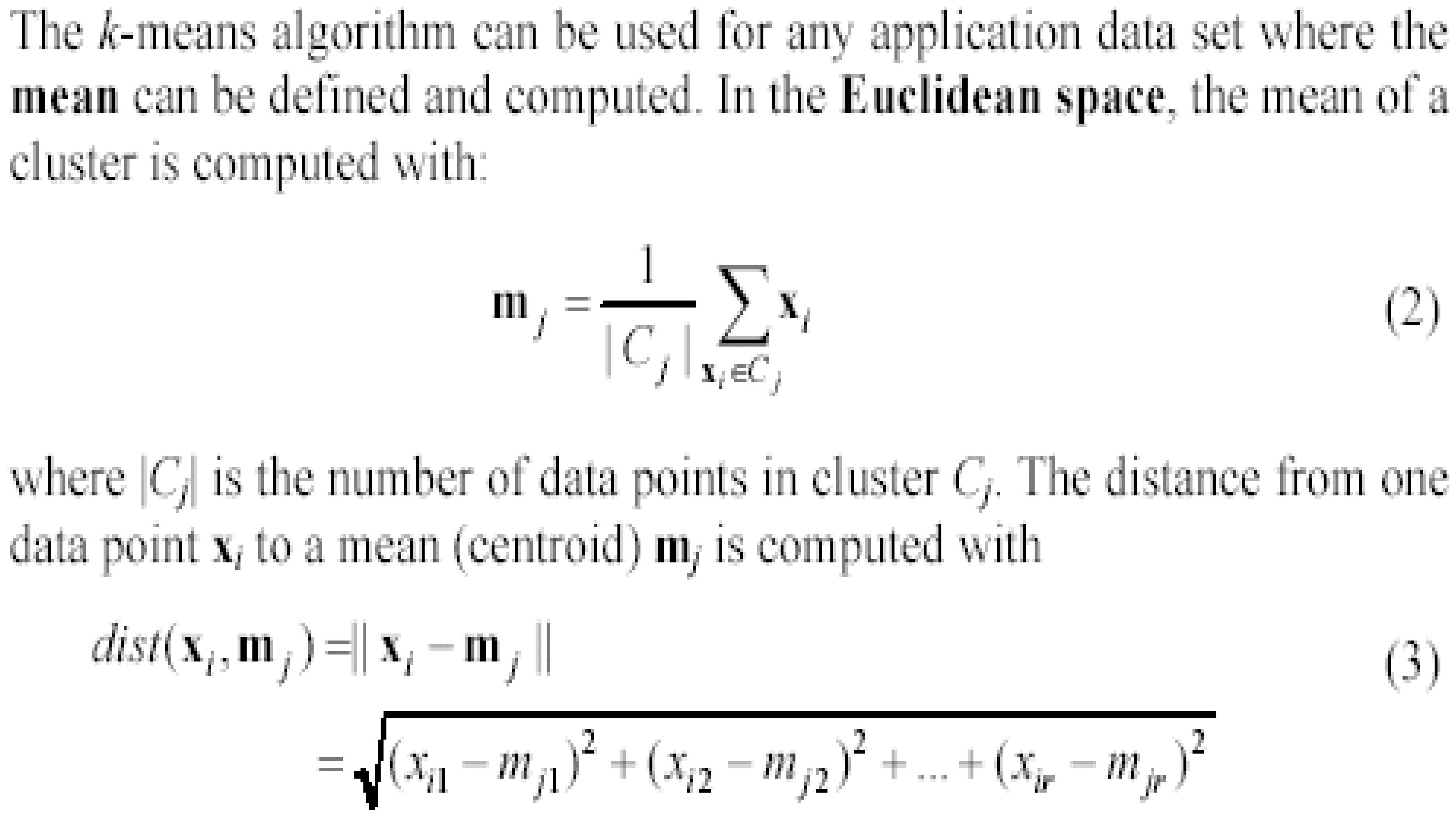

Iteration 2: (D). Cluster assignment


(E). Re-compute centroids


Iteration 3: (F). Cluster assignment


(G). Re-compute centroids

An example distance function



The *k*-means algorithm can be used for any application data set where the **mean** can be defined and computed. In the **Euclidean space**, the mean of a cluster is computed with:

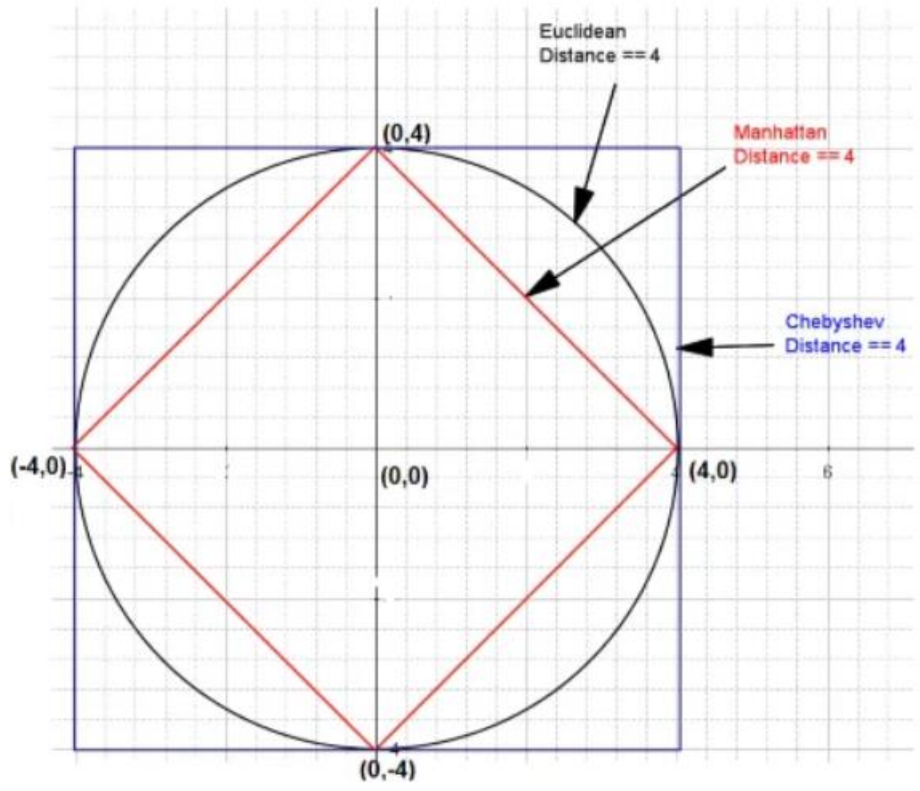
$$\mathbf{m}_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i \qquad (2)$$

where $|C_j|$ is the number of data points in cluster C_j. The distance from one data point $\mathbf{x}_i$ to a mean (centroid) $\mathbf{m}_j$ is computed with

$$dist(\mathbf{x}_i, \mathbf{m}_j) = \| \mathbf{x}_i - \mathbf{m}_j \| \qquad (3)$$

$$= \sqrt{(x_{i1} - m_{j1})^2 + (x_{i2} - m_{j2})^2 + ... + (x_{ir} - m_{jr})^2}$$

Other geometrical distances



$$d_E(\vec{p},\vec{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

Euclidean distance

$$d_M(\vec{p},\vec{q}) = \sum_{i=1}^{n}|p_i - q_i|$$

Manhattan distance

$$d_{Cheb}(\vec{p},\vec{q}) = \max_i |(p_i - q_i)|$$

Maximum(Chebychev) distance

Other distances: Mahalanobis

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

σ_1^2 - Horizontal variance

σ_2^2 - Vertical variance

$$d(x, c) = \sqrt{\frac{(x_1 - c_1)^2}{\sigma_1^2} + \frac{(x_2 - c_2)^2}{\sigma_2^2}}$$

Mahalanobis distance for a two dimensional vector with no covariance

Remarks:

- When you use Euclidean distance, you assume that the clusters have identity covariances. In 2D, this means that your clusters have circular shapes. Obviously, if the covariances of the natural groupings in your data are not identity matrices, e.g. in 2D, clusters have elliptical shaped covariances, then using Mahalanobis over Euclidean will be much better modeling.
- Manhattan distance should give more robust results, whereas Euclidean distance is likely to be influenced by outliers.

Remarks:

- **The value of distance measures is intimately related to the scale on which measurements are made.** Therefore, variables are often scaled before measuring the inter-observation dissimilarities. This is particularly recommended when variables are measured in different scales (e.g: kilograms, kilometers, centimeters, …); otherwise, the dissimilarity measures obtained will be severely affected.
- **Standardization makes the four distance measure methods — Euclidean, Manhattan, Correlation and Eisen — more similar than they would be with non-transformed data.** Note that, **when the data are standardized, there is a functional relationship between the Pearson correlation coefficient & Euclidean distance.**

Strengths of k-means

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where n is the number of data points,

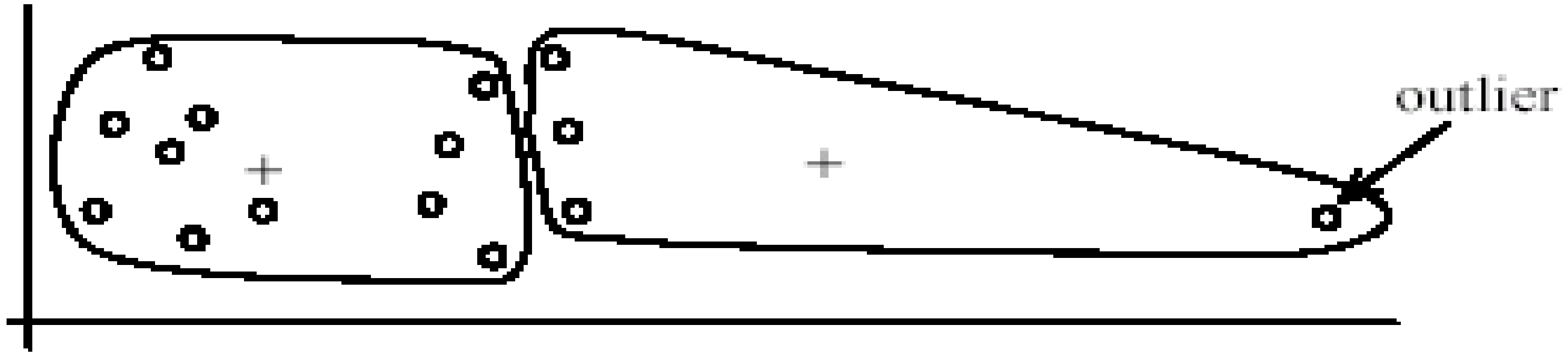
    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small. *k*-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE (or WCSS) is used. The global optimum is hard to find due to complexity.

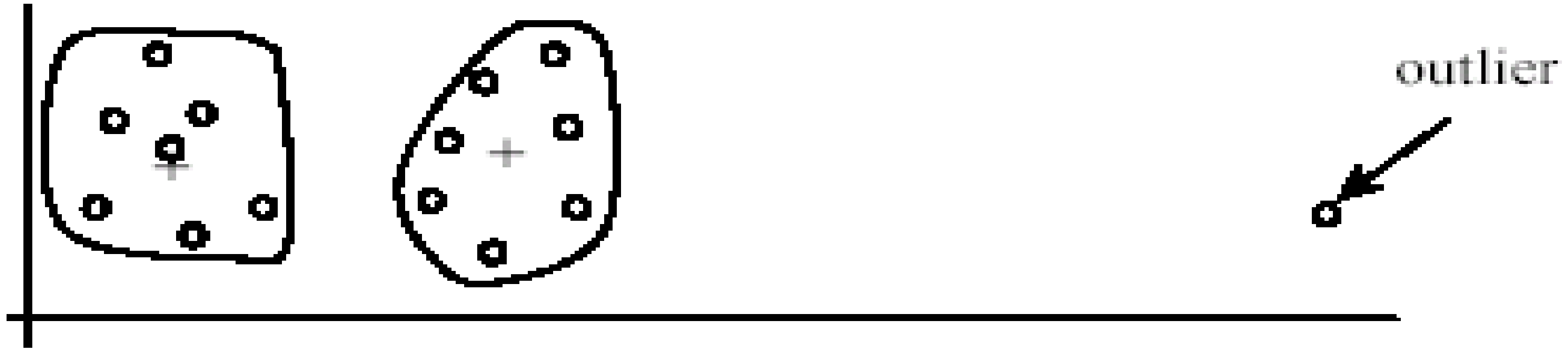
Weaknesses of k-means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Weaknesses of k-means: Problems with outliers



(A): Undesirable clusters



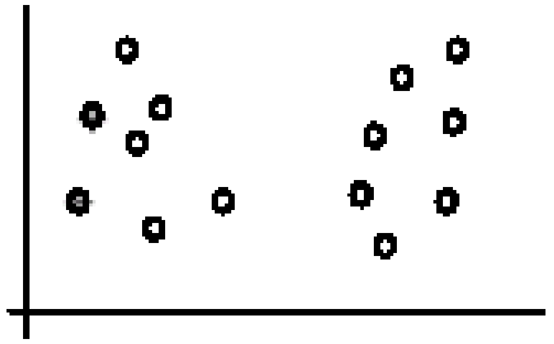
(B): Ideal clusters

Weaknesses of k-means: To deal with outliers

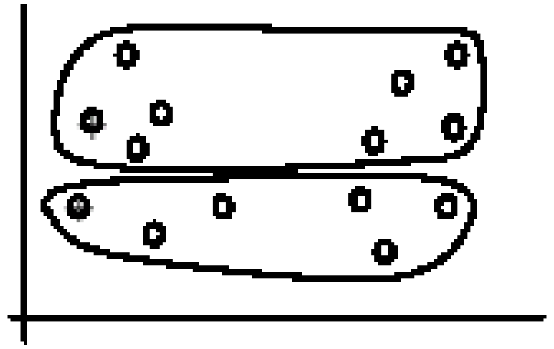
- One method is to remove some data points in the clustering process that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

Weaknesses of k-means (cont …)

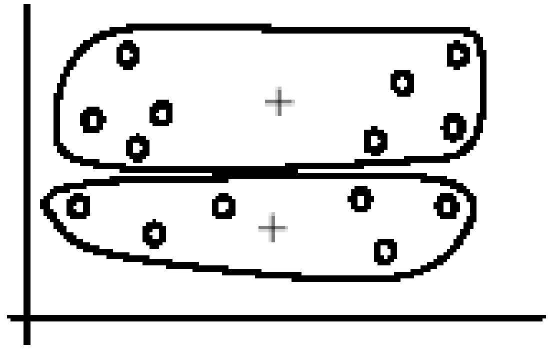
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



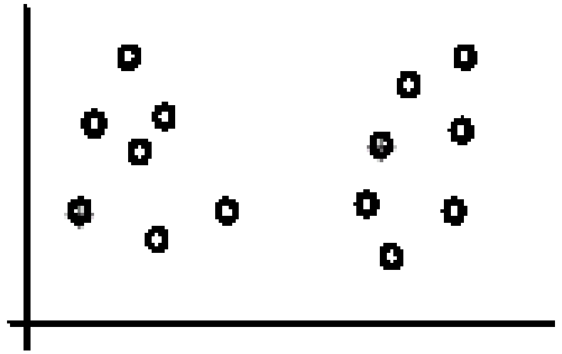
(B). Iteration 1



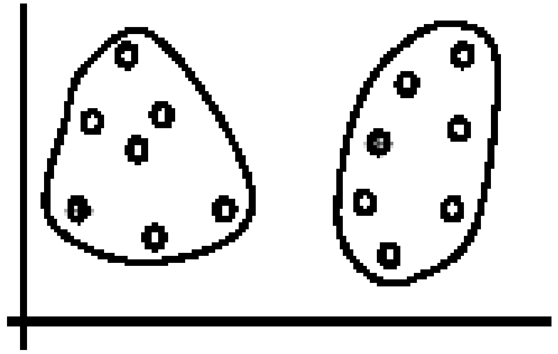
(C). Iteration 2

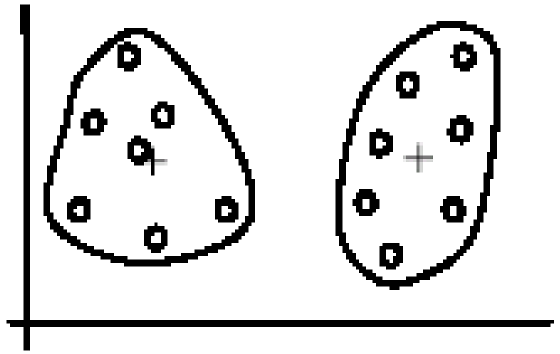
Weaknesses of k-means (cont …)

- If we use different seeds: good results


(A). Random selection of *k* seeds (centroids)

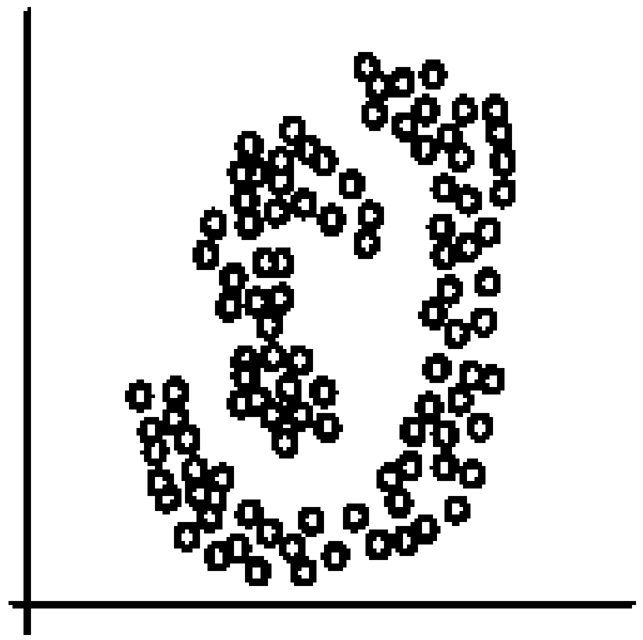
There are some methods to help choose good seeds

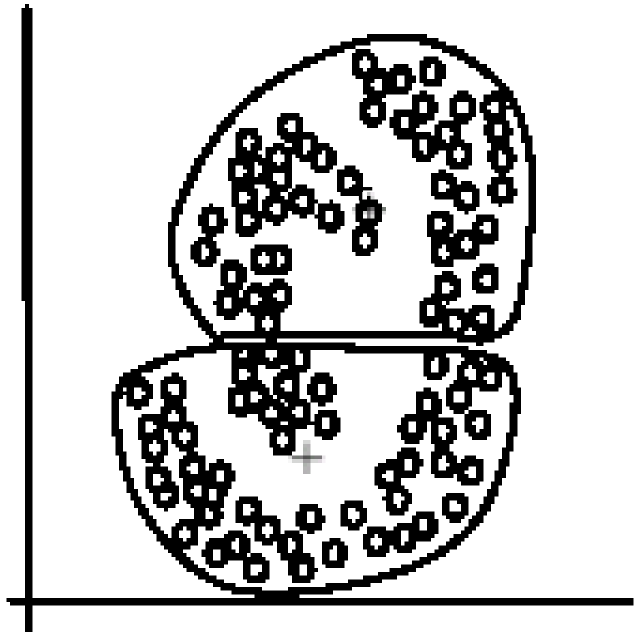

(B). Iteration 1


(C). Iteration 2

Weaknesses of k-means (cont …)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).


(A): Two natural clusters


(B): *k*-means clusters

Hyperparameter Tuning in K-Means Clustering

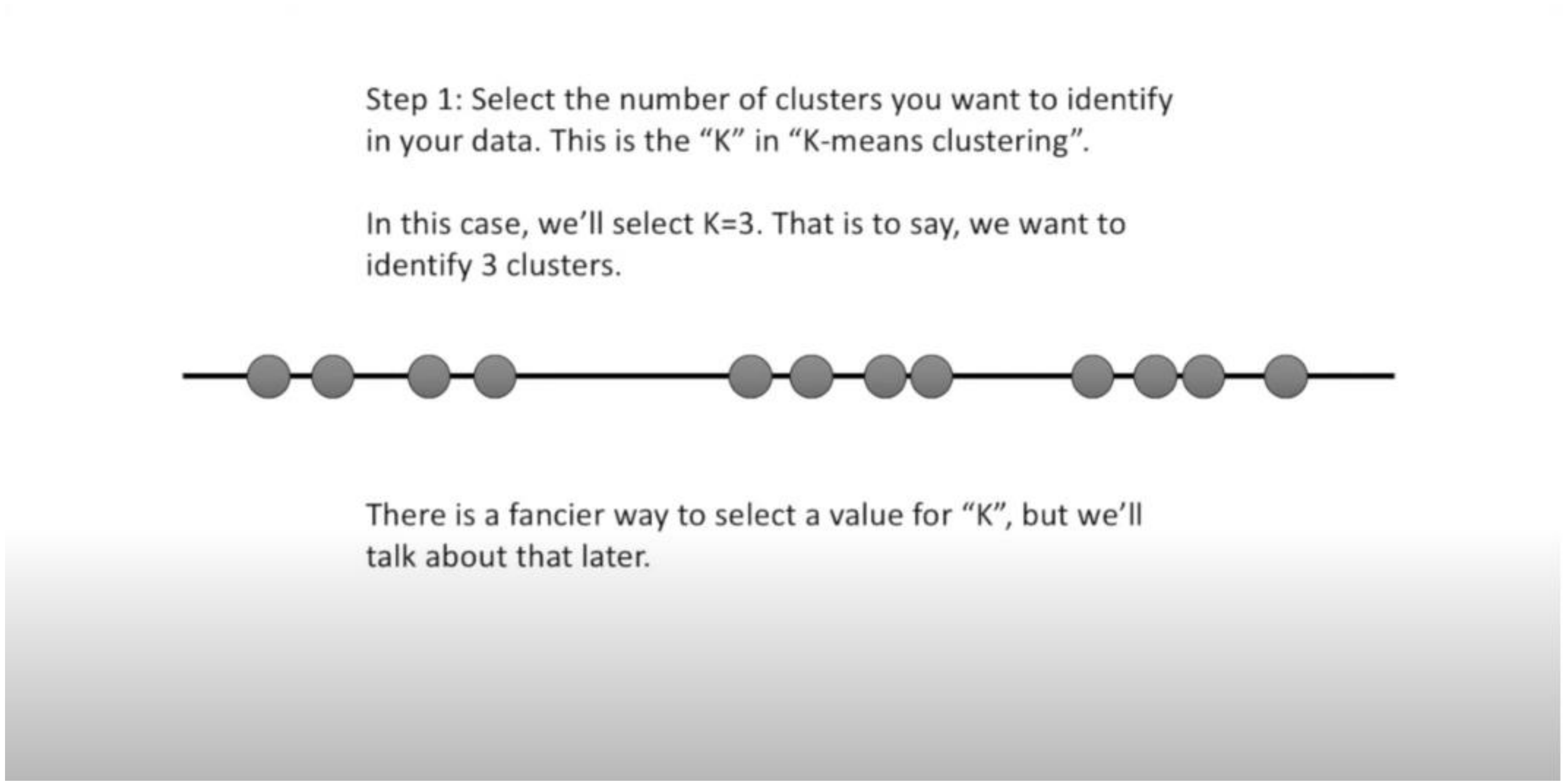
Hyperparameters in K-Means Clustering

Number of clusters

Seeds - initial values

Distance measures

How to select initial seeds?

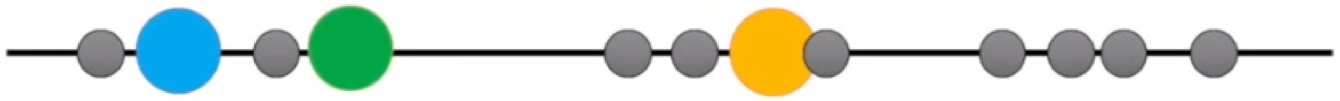


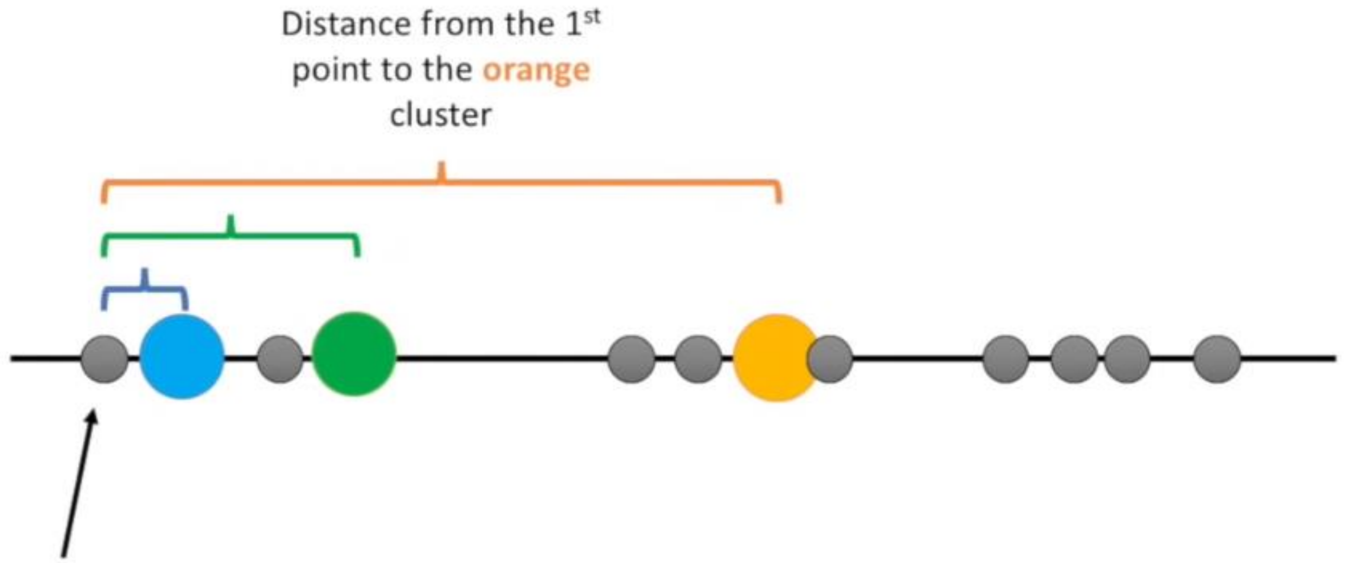
Step 1: Select the number of clusters you want to identify in your data. This is the "K" in "K-means clustering".

In this case, we'll select K=3. That is to say, we want to identify 3 clusters.

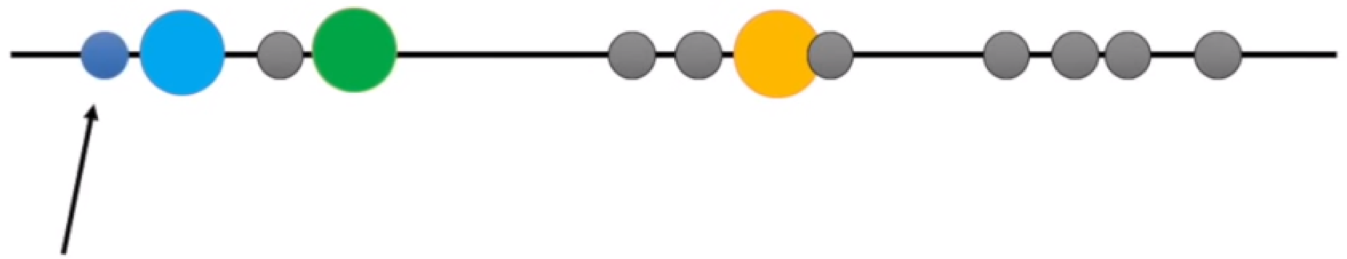
There is a fancier way to select a value for "K", but we'll talk about that later.

Step 2: Randomly select 3 distinct data points.



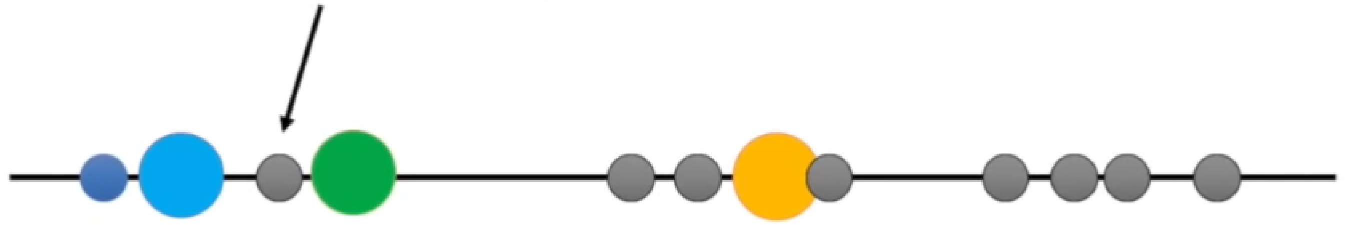


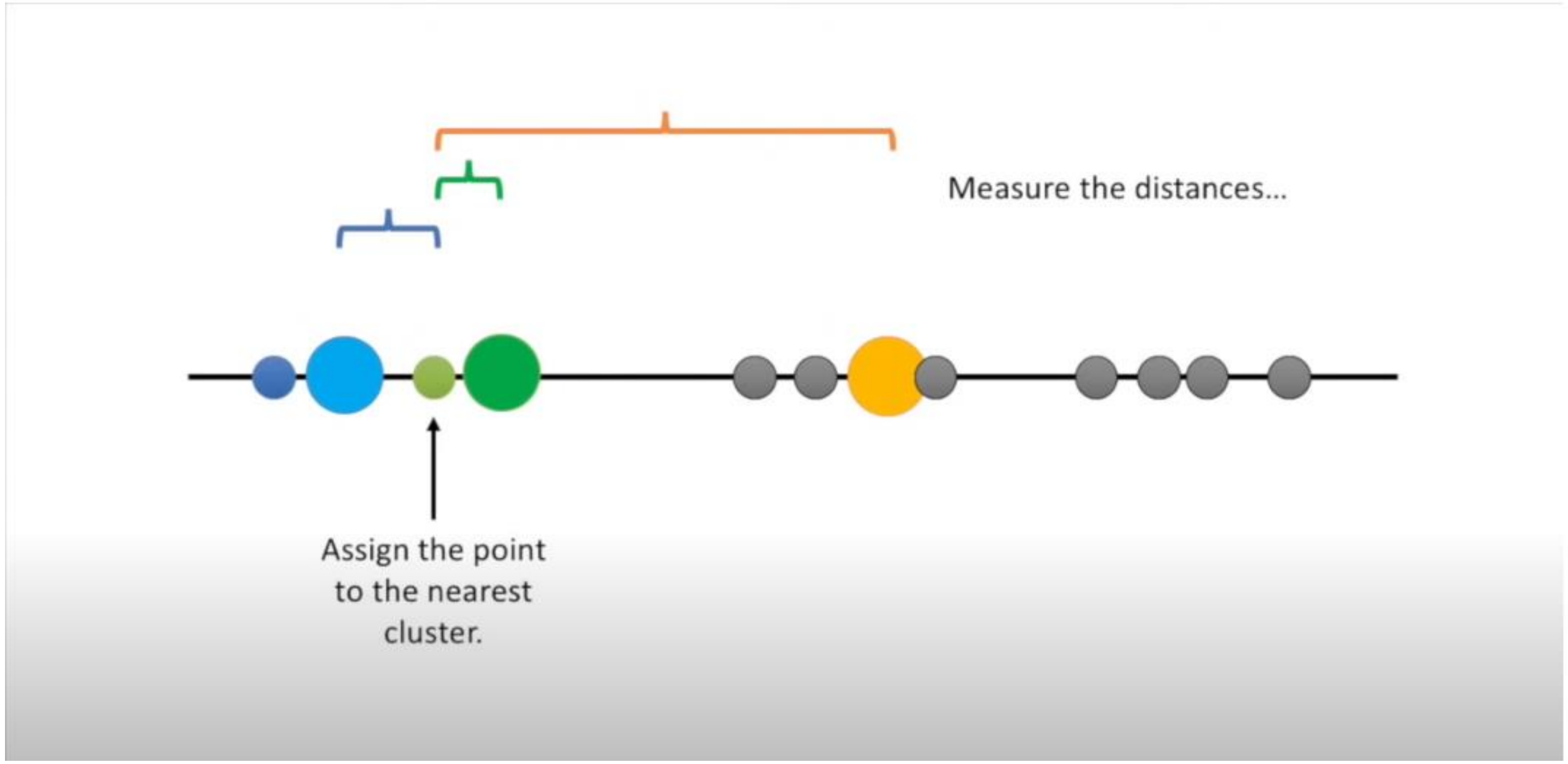
Step 3: Measure the distance between the 1st point and the three initial clusters.

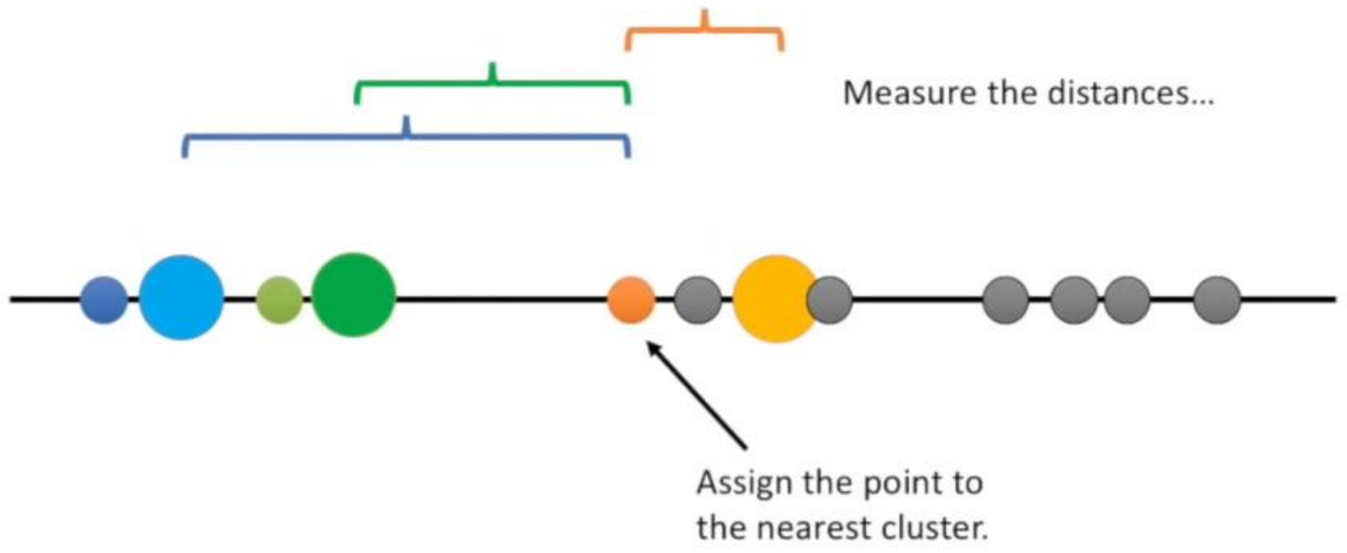


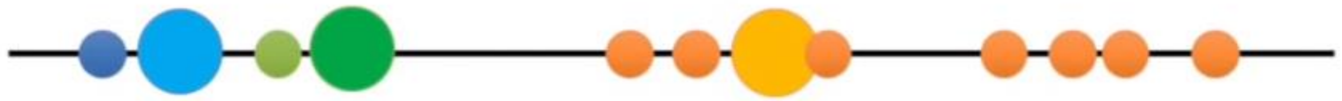
Step 4: Assign the 1st point to the nearest cluster. In this case, the nearest cluster is the **blue** cluster.

Now do the same thing for the next point.

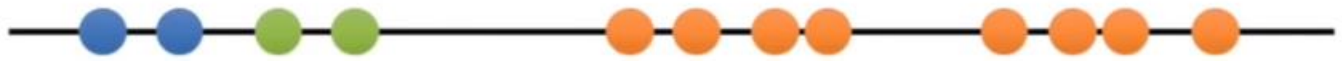



Measure the distances...
Assign the point to the nearest cluster.

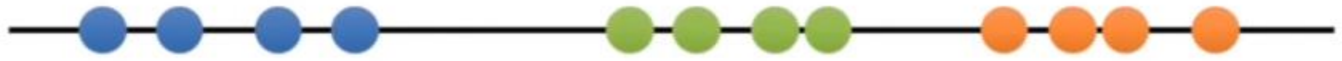

Measure the distances...
Assign the point to the nearest cluster.



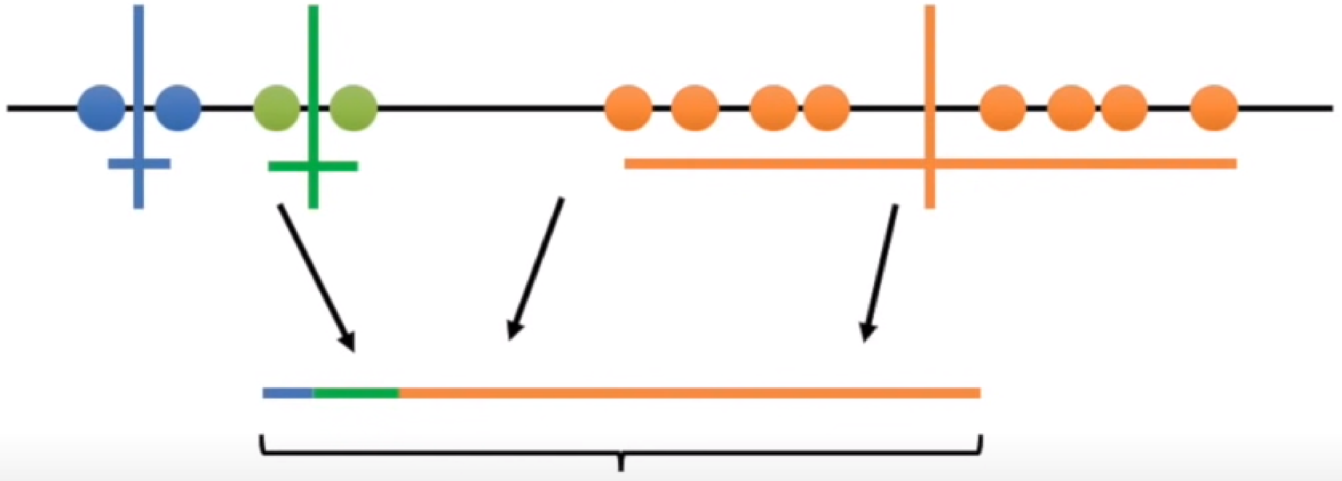
The rest of these points are closest to the **orange** cluster, so they'll go in that one, too.



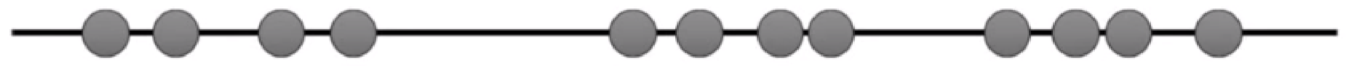
The K-means clustering is pretty terrible compared to what we did by eye.



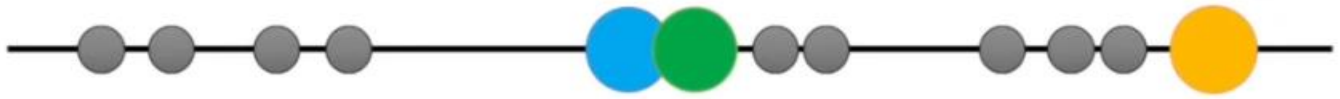
We can assess the quality of the clustering by adding up the variation within each cluster.



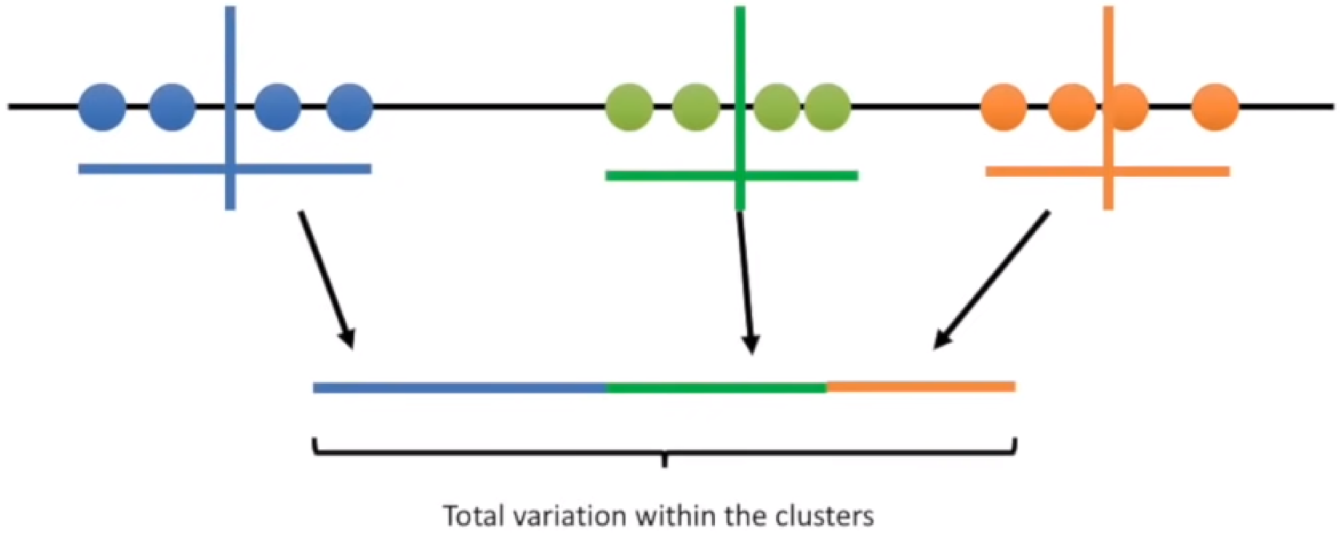
Since K-means clustering can't "see" the best clustering, its only option is to keep track of these clusters, and their total variance, So, here we are again, back at the beginning.



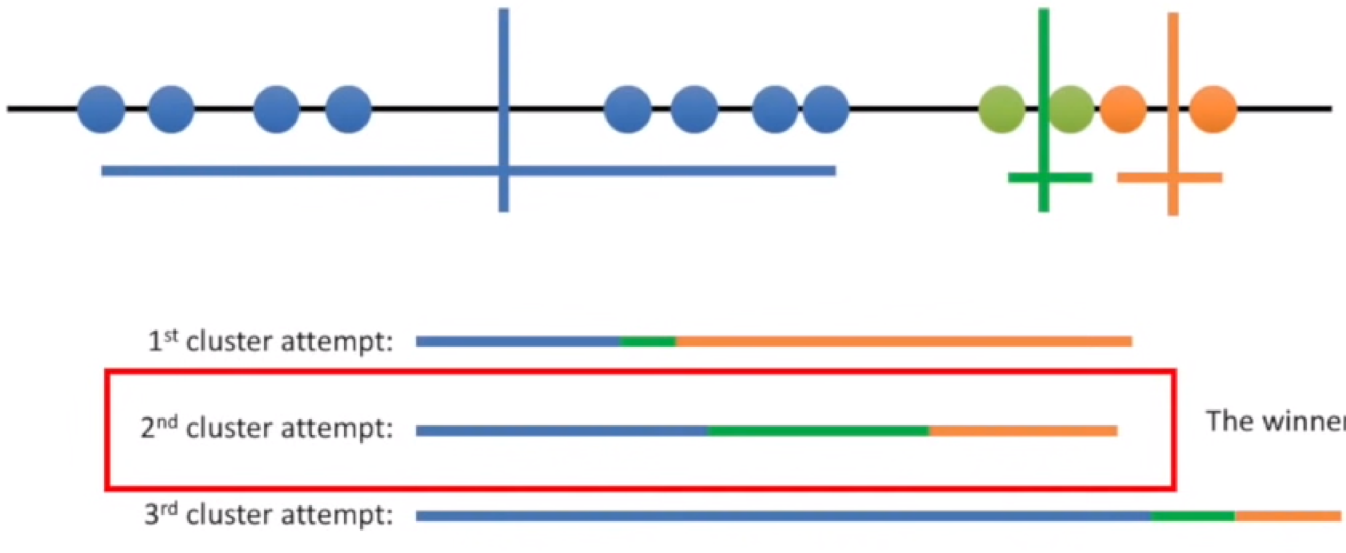
K-means clustering picks 3 initial clusters...



Now that the data are clustered, we sum the variation within each cluster.



At this point, K-means clustering knows that *the 2nd clustering is the best clustering so far*. But it doesn't know if it's *the best overall*, so it will do a few more clusters (it does as many as you tell it to do) and then come back and return that one if it is still the best.



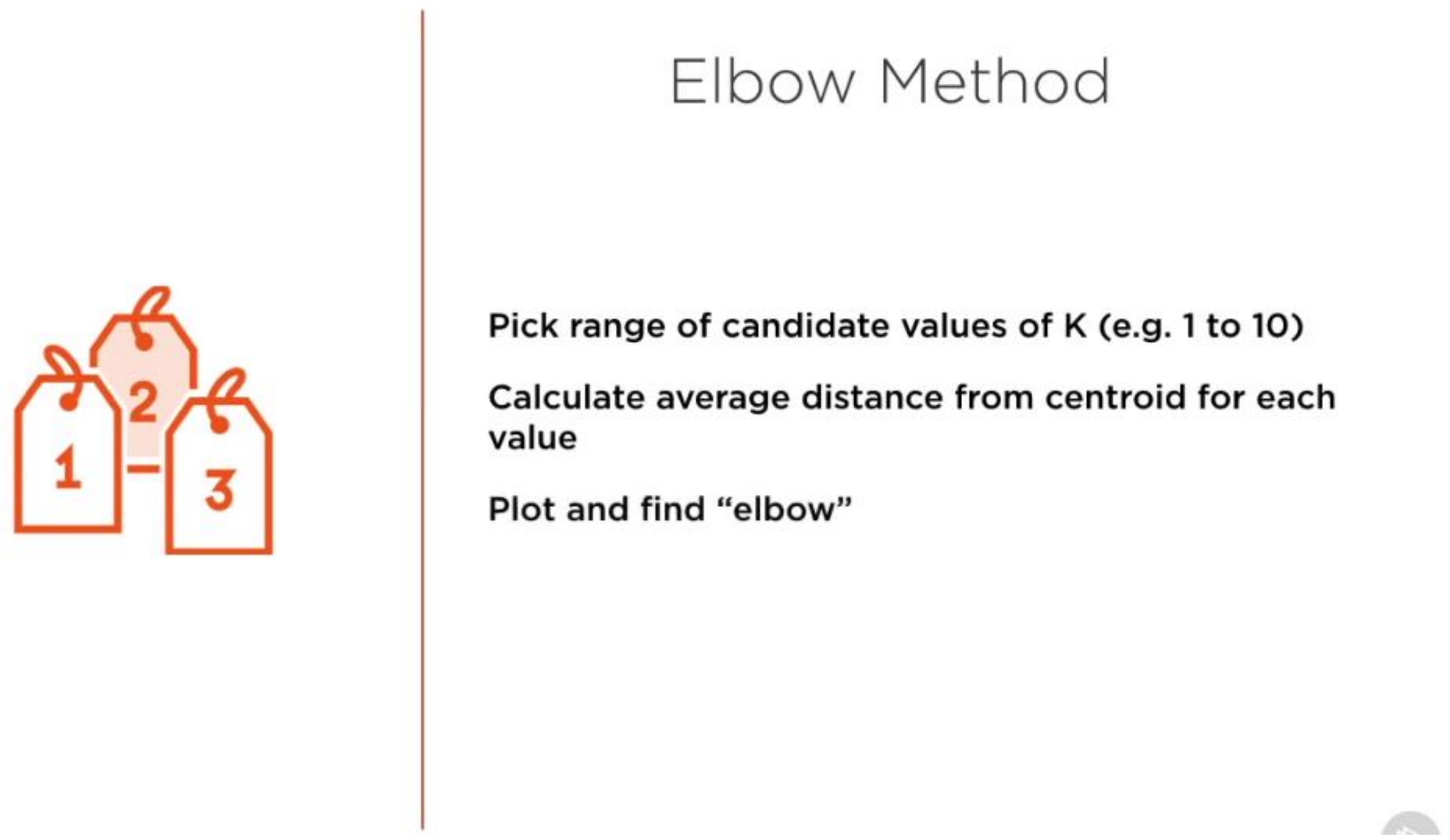
Number of Clusters

K is the most important hyperparameter

Sometimes obvious e.g. 10 in MNIST digit classification

Otherwise, apply standard method to find the “best” value of K

Elbow Method

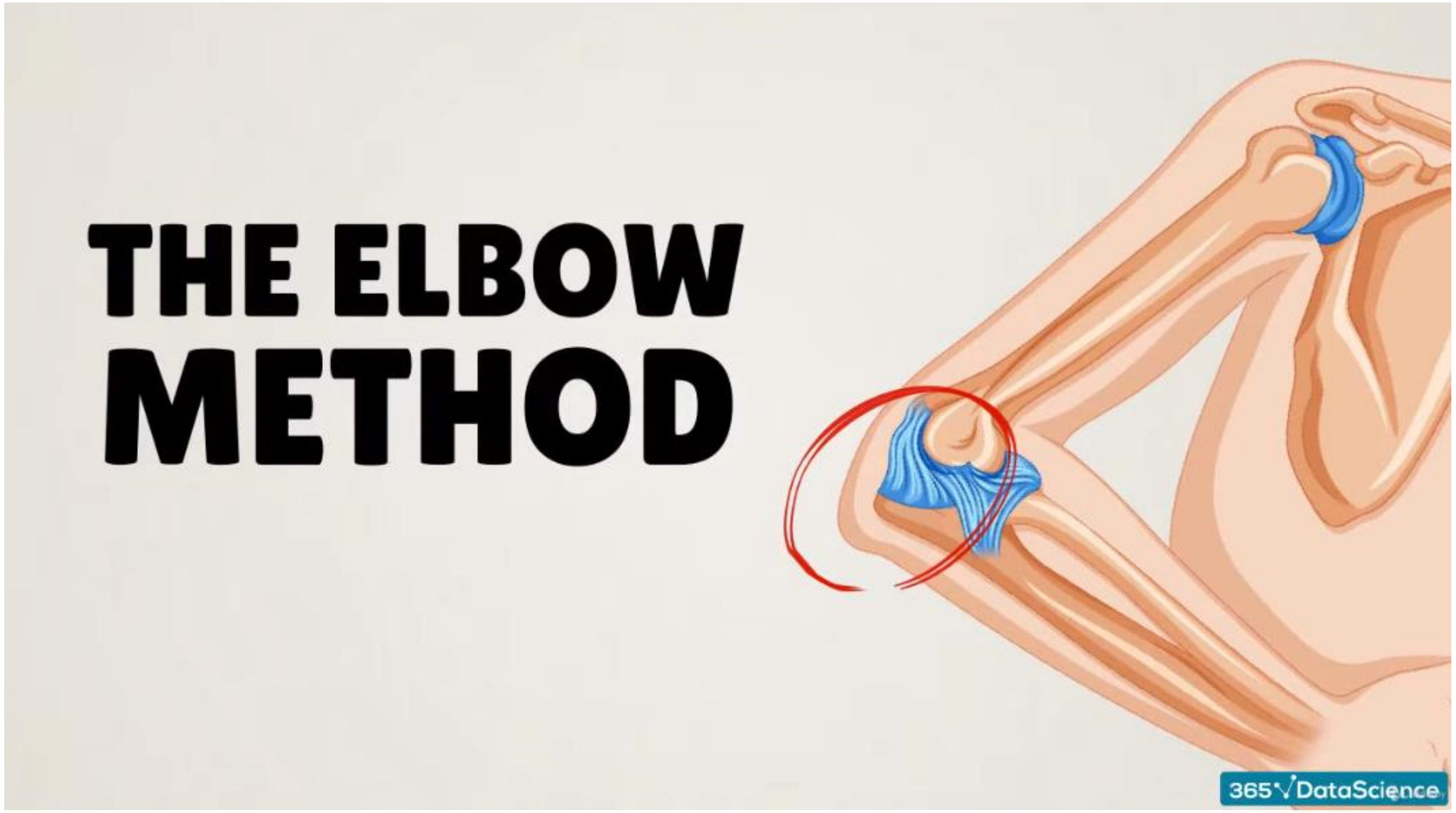


Elbow Method

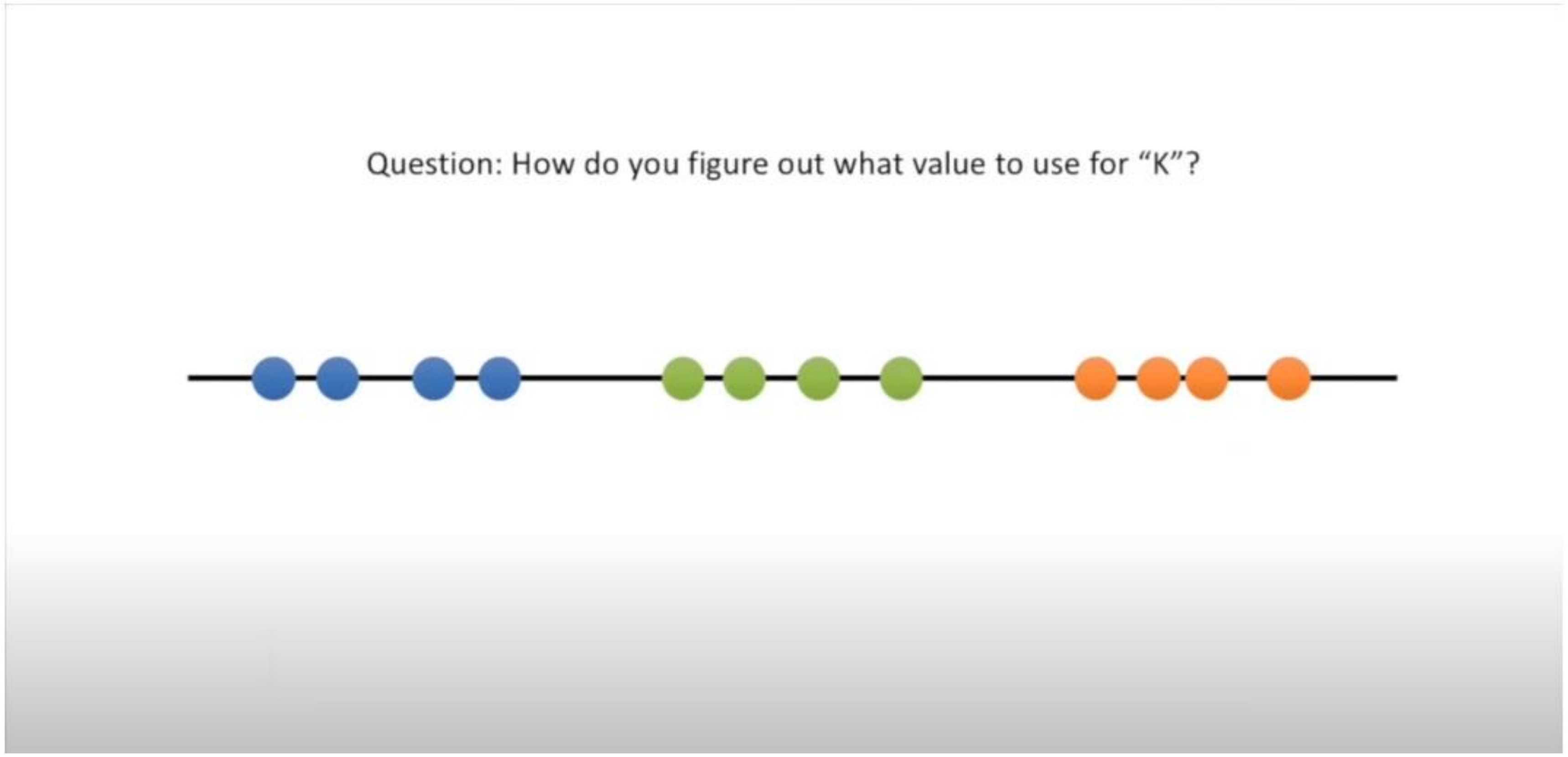
Pick range of candidate values of K (e.g. 1 to 10)

Calculate average distance from centroid for each value

Plot and find “elbow”

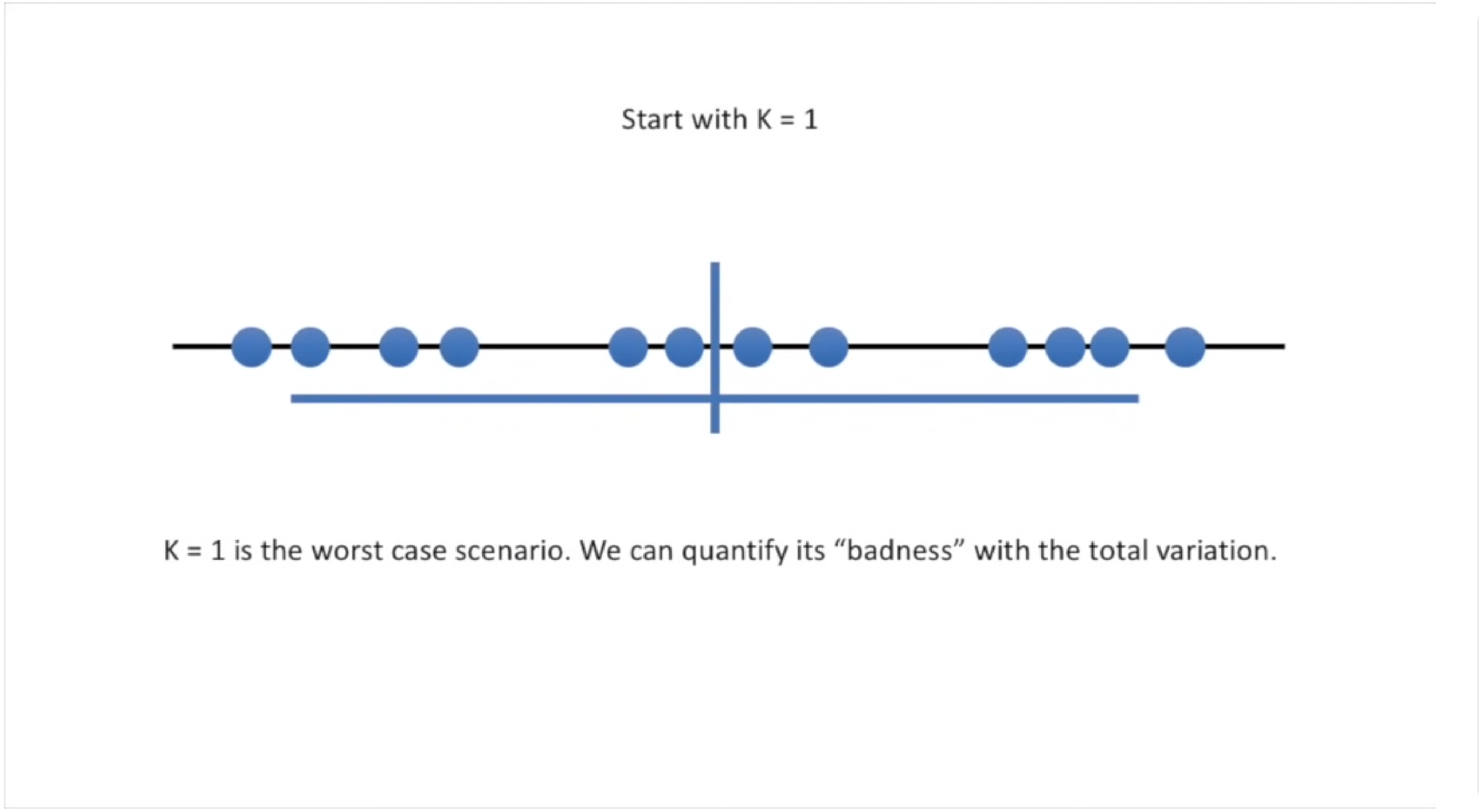

THE ELBOW METHOD
365√DataScience

How to select K?



Question: How do you figure out what value to use for "K"?

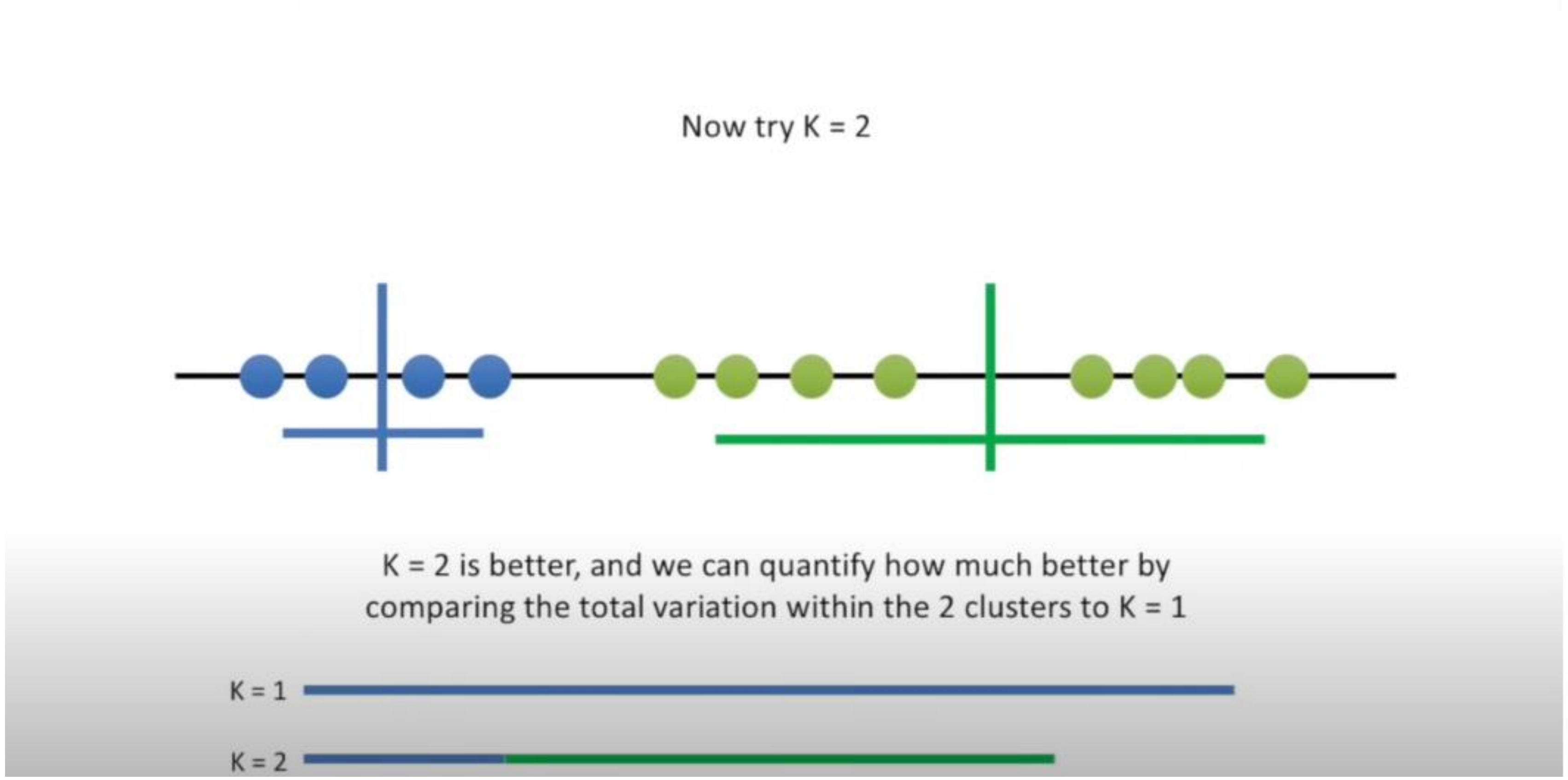
Start with K=1



Start with K = 1

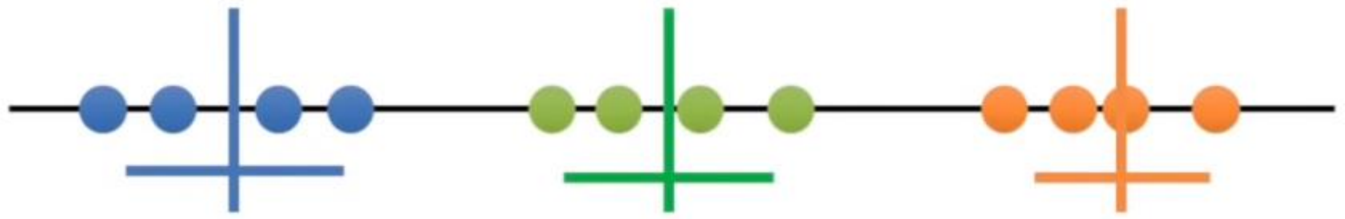
K = 1 is the worst case scenario. We can quantify its “badness” with the total variation.

Now try K=2

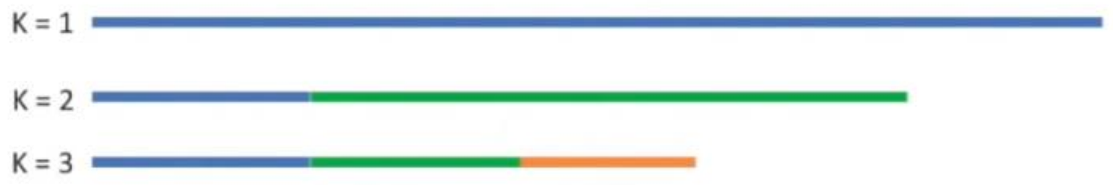

Now try K = 2

K = 2 is better, and we can quantify how much better by comparing the total variation within the 2 clusters to K = 1

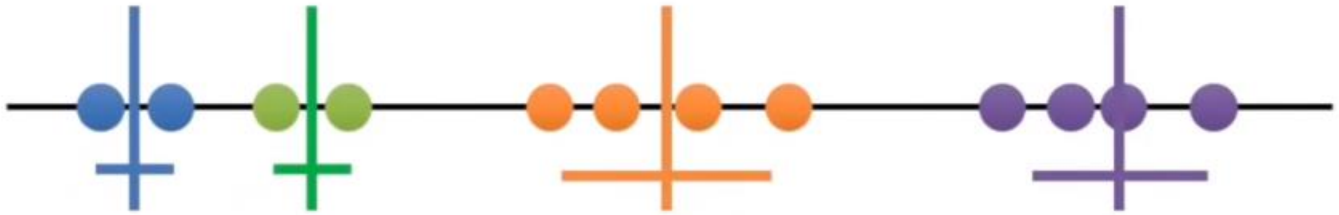
Now try K = 3



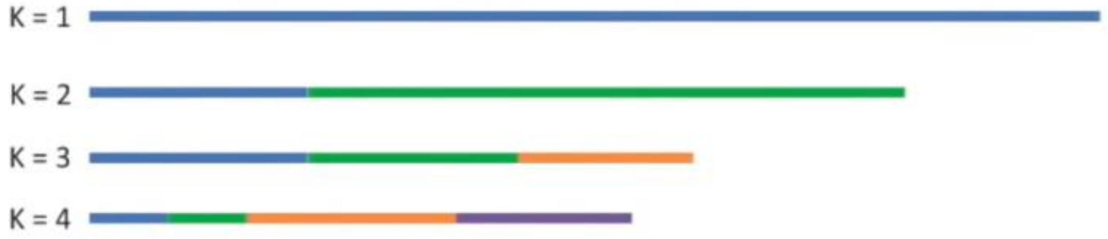
K = 3 is even better! We can quantify how much better by comparing the total variation within the 3 clusters to K = 2



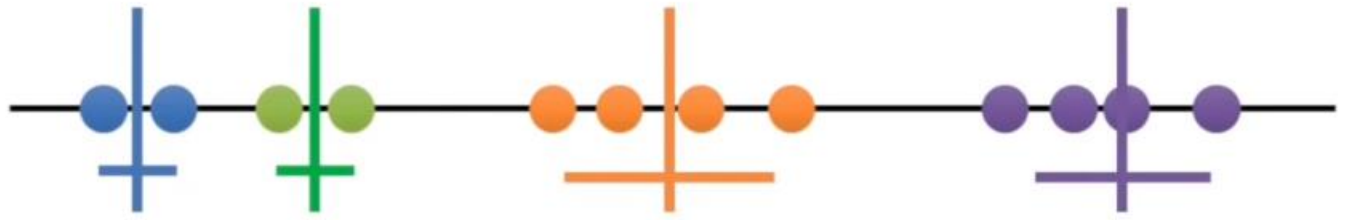
Now try K = 4



The total variation within each cluster is less than when K=3

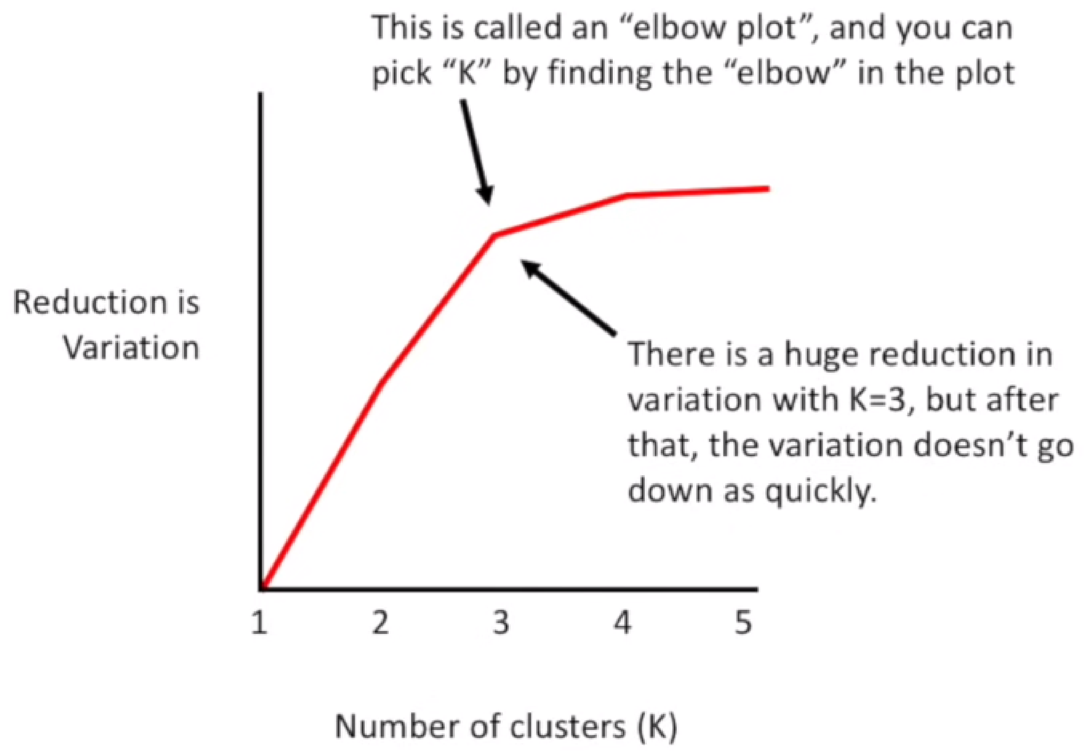


Now try K = 4

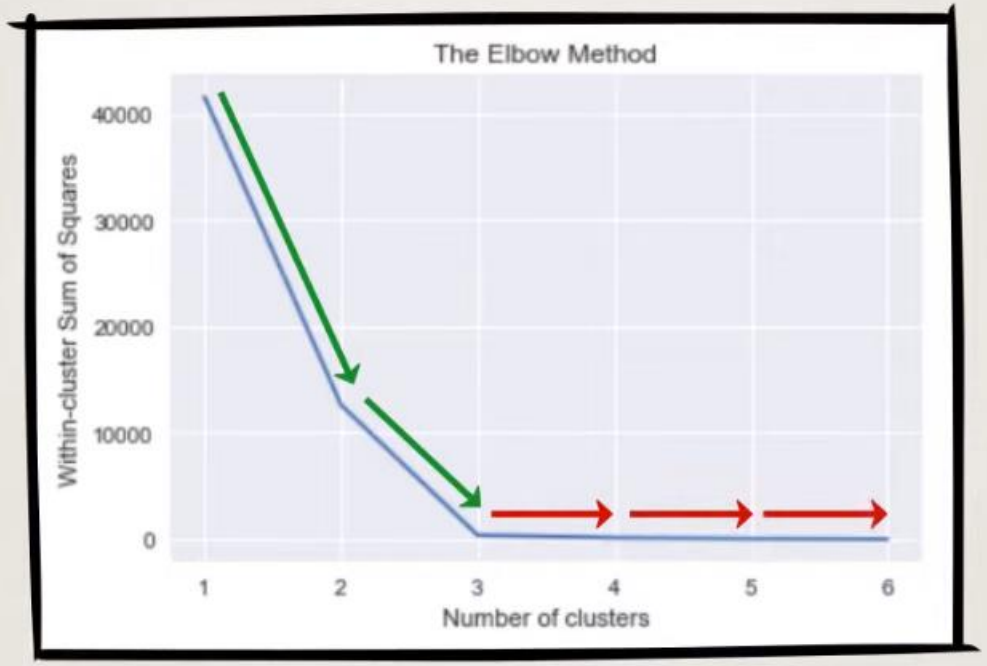


The total variation within each cluster is less than when K=3

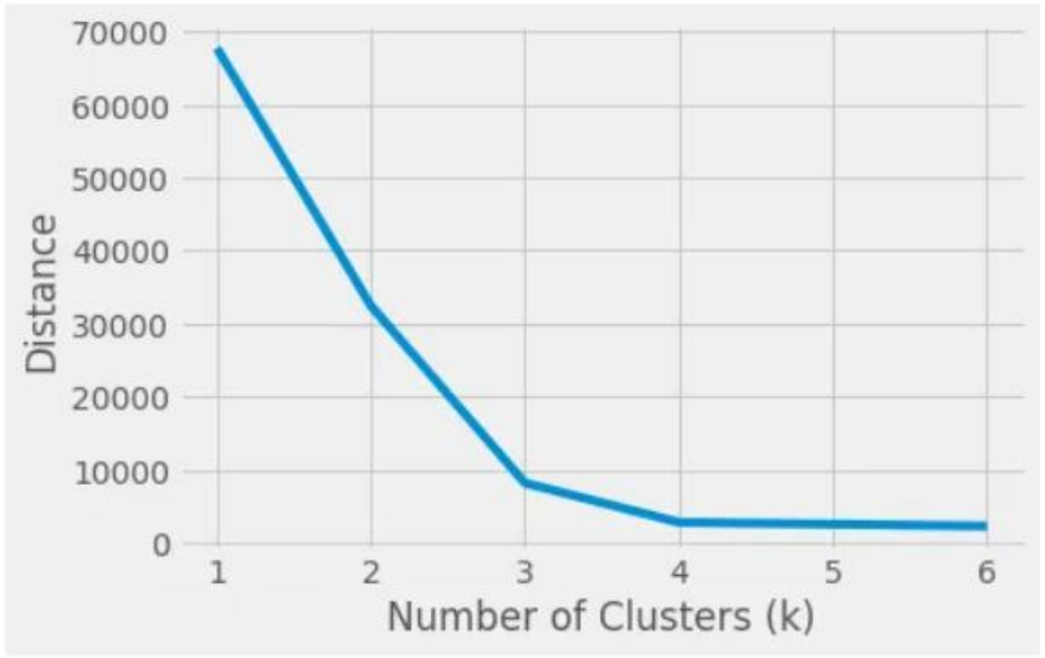
Each time we add a new cluster, the total variation within each cluster is smaller than before. And when there is only one point per cluster, the variation = 0.


This is called an "elbow plot", and you can pick "K" by finding the "elbow" in the plot
Reduction is Variation
There is a huge reduction in variation with K=3, but after that, the variation doesn't go down as quickly.
1
2
3
4
5
Number of clusters (K)

THE ELBOW METHOD

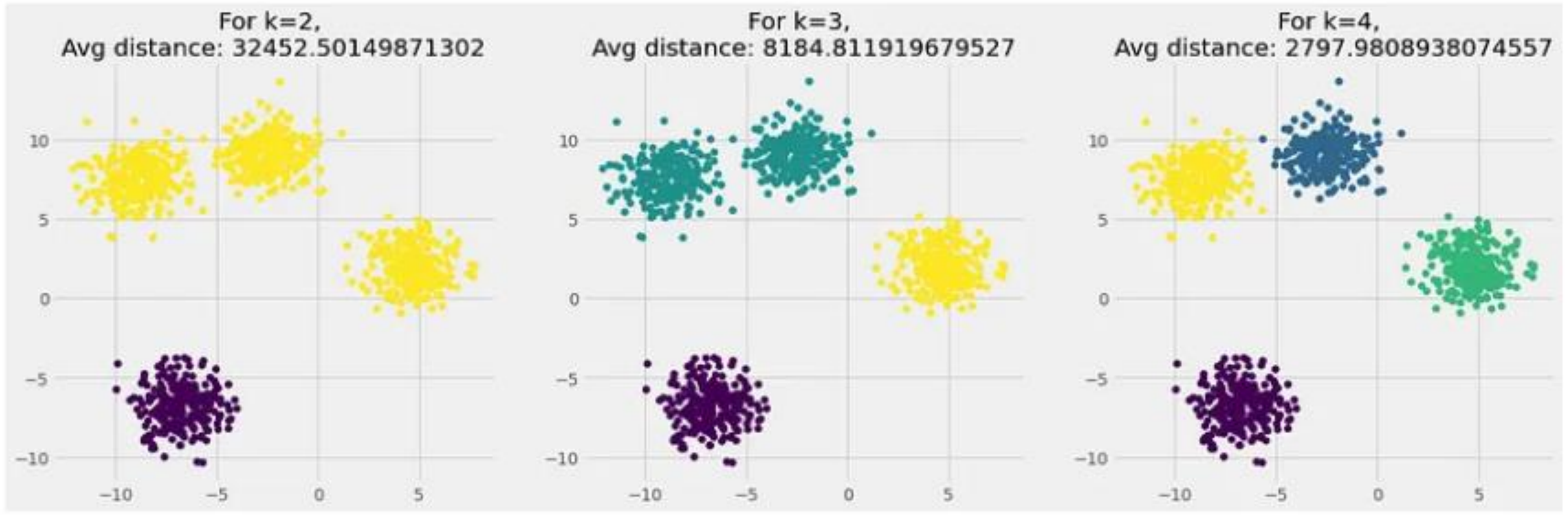


Another example



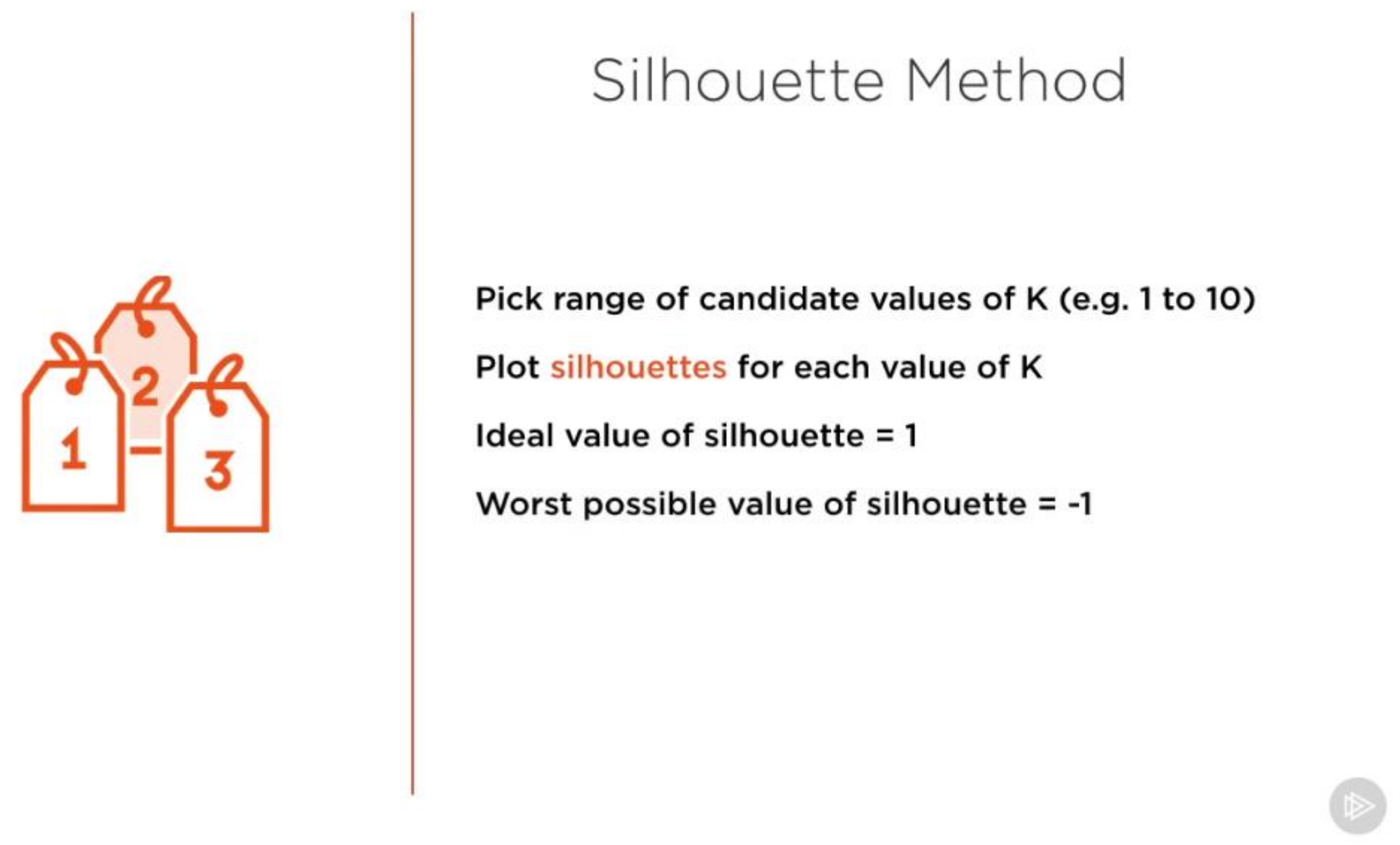
(Image by Author), Elbow Method to find optimal k

So plot and observe!!



(Image by Author), Scatter plot of clusters formed at k=2, 3, and 4

Silhouette Method



Silhouette Method

Pick range of candidate values of K (e.g. 1 to 10)

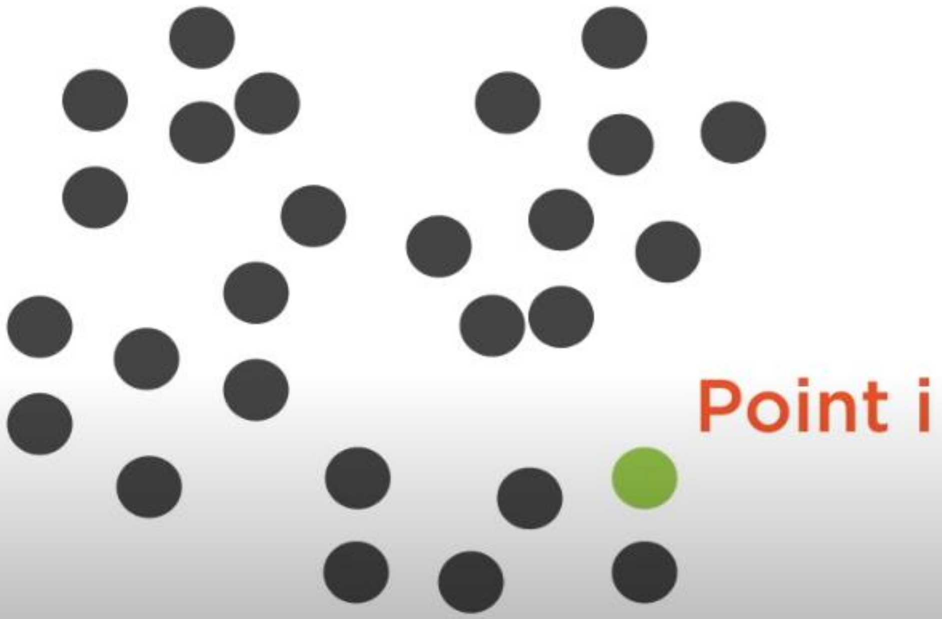
Plot silhouettes for each value of K

Ideal value of silhouette = 1

Worst possible value of silhouette = -1

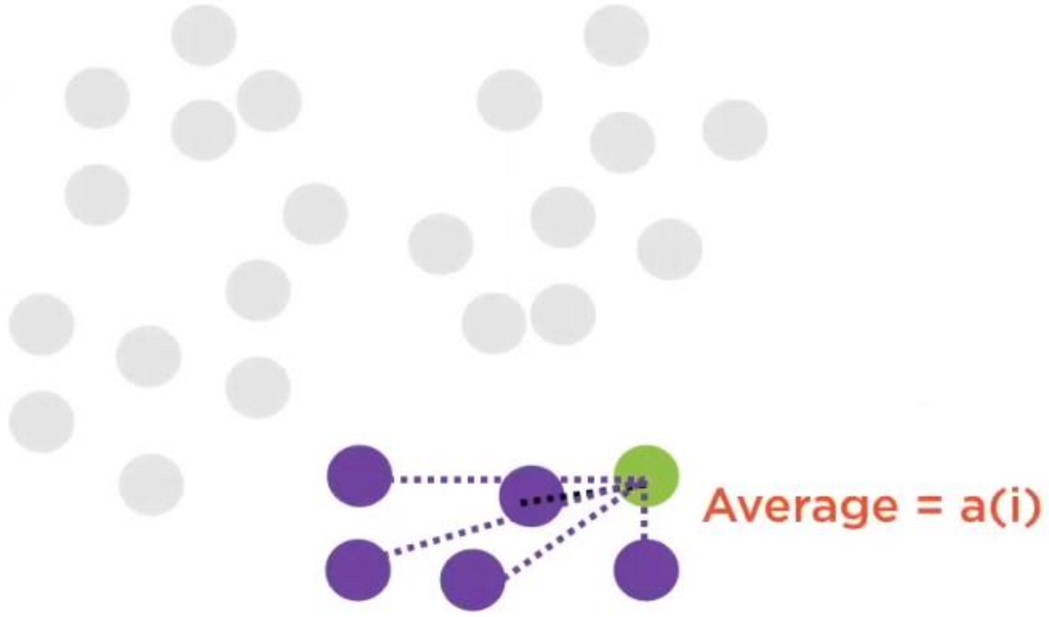
Silhouette Coefficient

For any point i, calculate silhouette coefficient



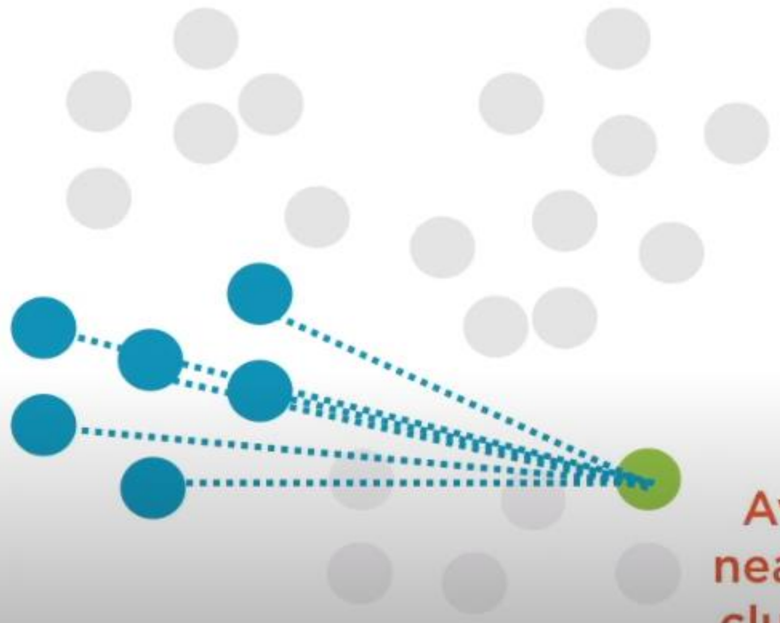
Silhouette Coefficient

Find a(i) = average distance of i to other points in same cluster



Silhouette Coefficient

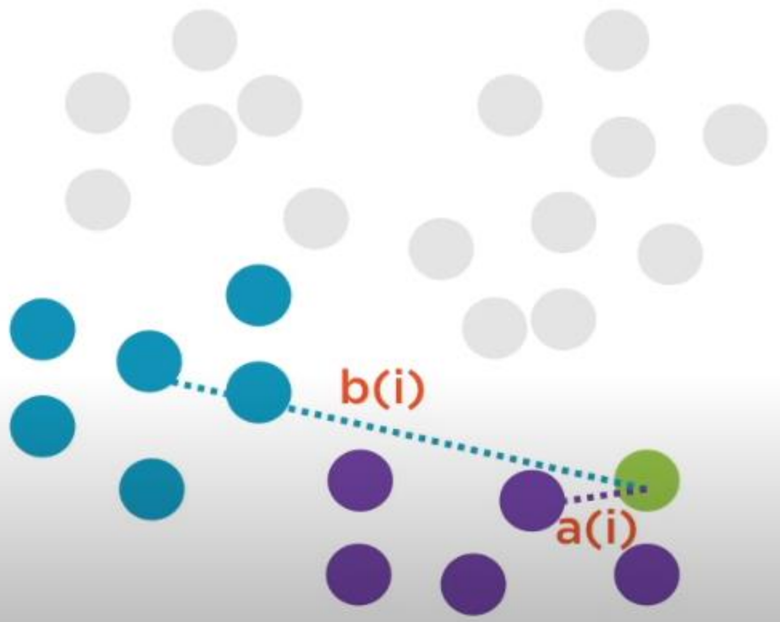
Find b(i) = average distance to nearest other cluster



Average to nearest other cluster = b(i)

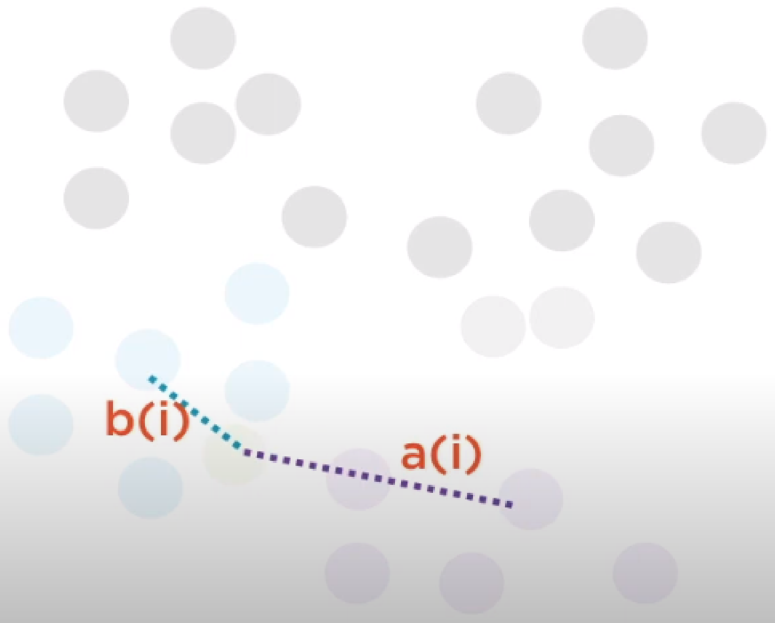
Silhouette Coefficient

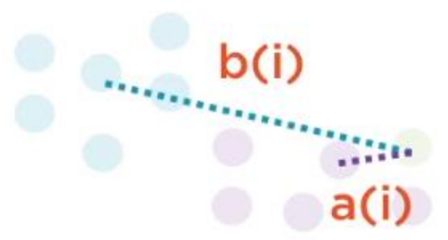
Ideally, a(i) << b(i)



Silhouette Coefficient

If a(i) > b(i), i is likely misclassified





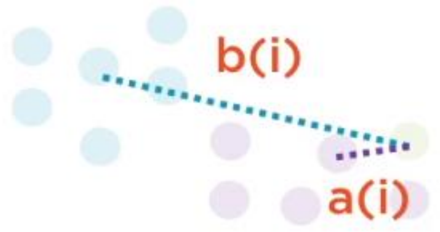
Silhouette Coefficient

For any point i

$$s(i) = \frac{b(i) - a(i)}{\text{Larger of } b(i) \text{ and } a(i)}$$

a(i) = Average distance inside cluster

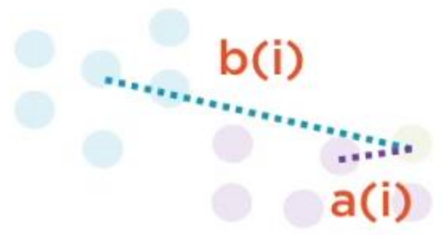
b(i) = Average distance to nearest other cluster



Ideally s(i) = 1

Ideally, a(i) = 0, b(i) = Infinity

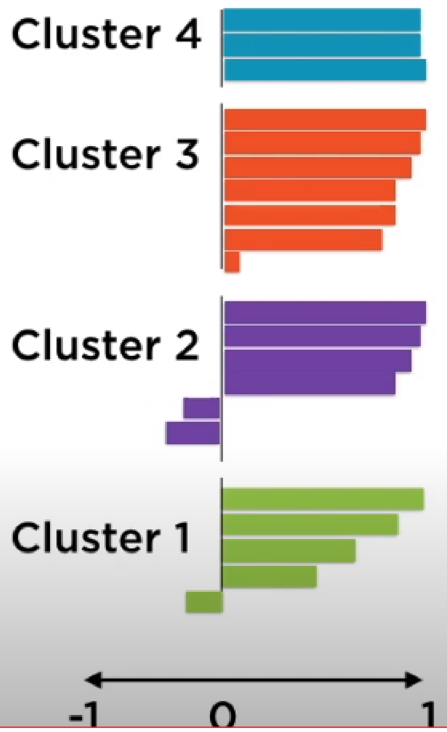
$$s(i) = \frac{b(i) - a(i)}{\text{Larger of } b(i) \text{ and } a(i)}$$



Worst-case s(i) = -1

Worst case, a(i) = Infinity, b(i) = 0

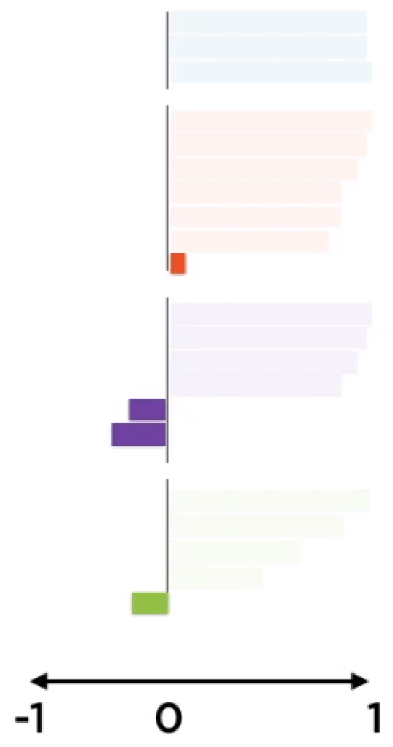
$$s(i) = \frac{b(i) - a(i)}{\text{Larger of } b(i) \text{ and } a(i)} = -1$$



Silhouette Plot

Calculate s(i) for each point

Plot value of s(i) to identify outliers



Outliers

Ideally, s(i) = 1

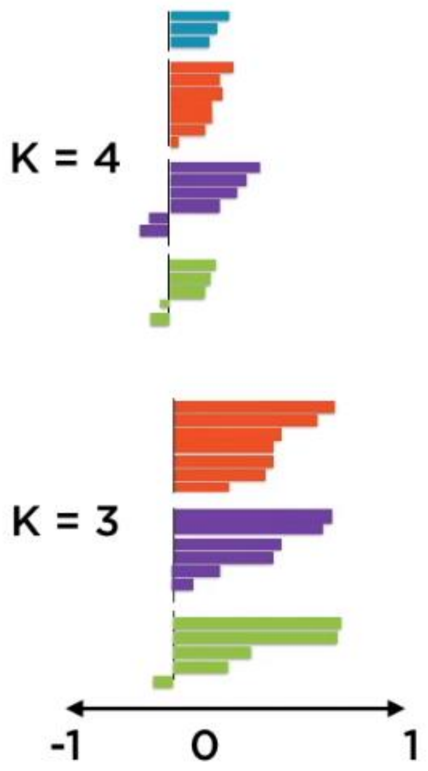
So, s(i) < 0 indicates outliers

“Best” K

Extend the same idea

Replicate plot for different values of K

Pick K where average silhouette is closest to 1

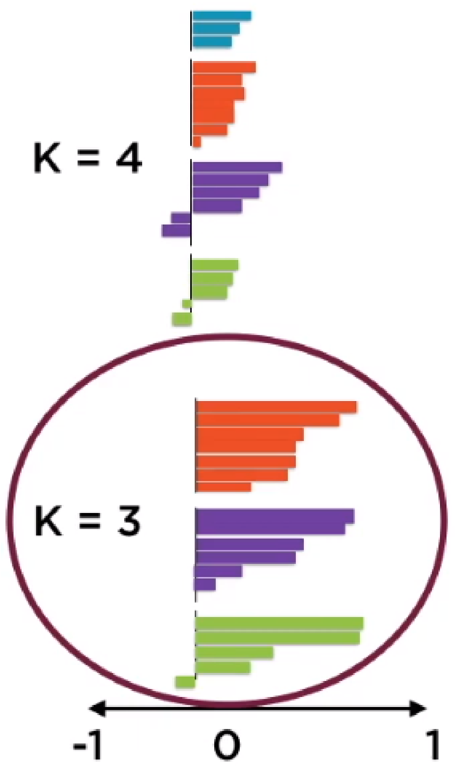


“Best” K

Extend the same idea

Replicate plot for different values of K

Pick K where average silhouette is closest to 1



“Best” K

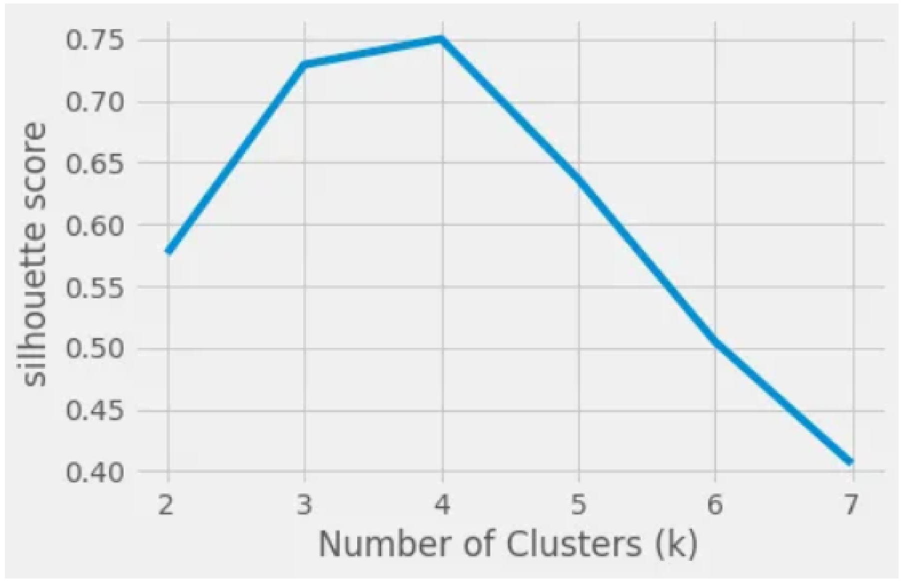
Here K = 3 is noticeably better than K = 4

K = 3 has noticeably larger positive values

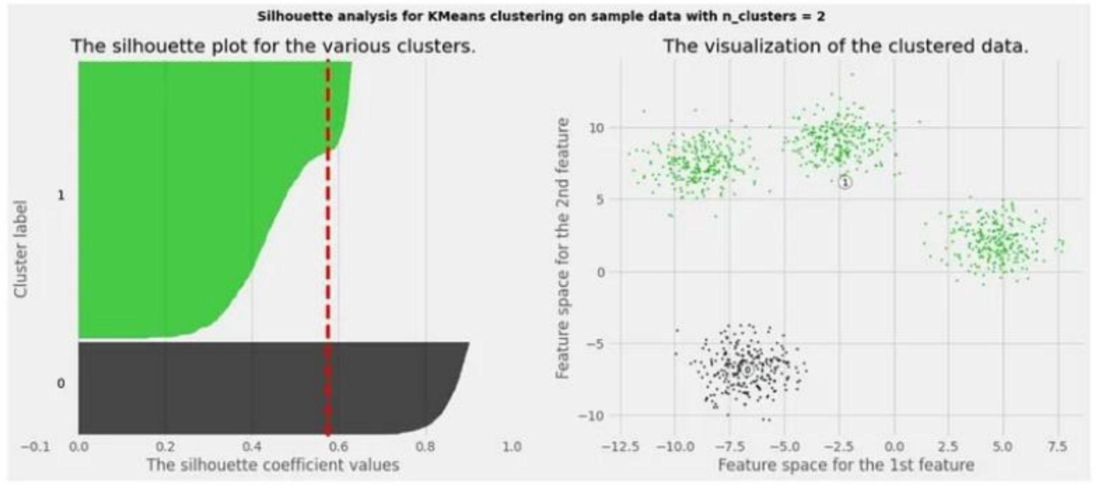
Find the optimal value of ‘k’ using Silhoutte Analysis:

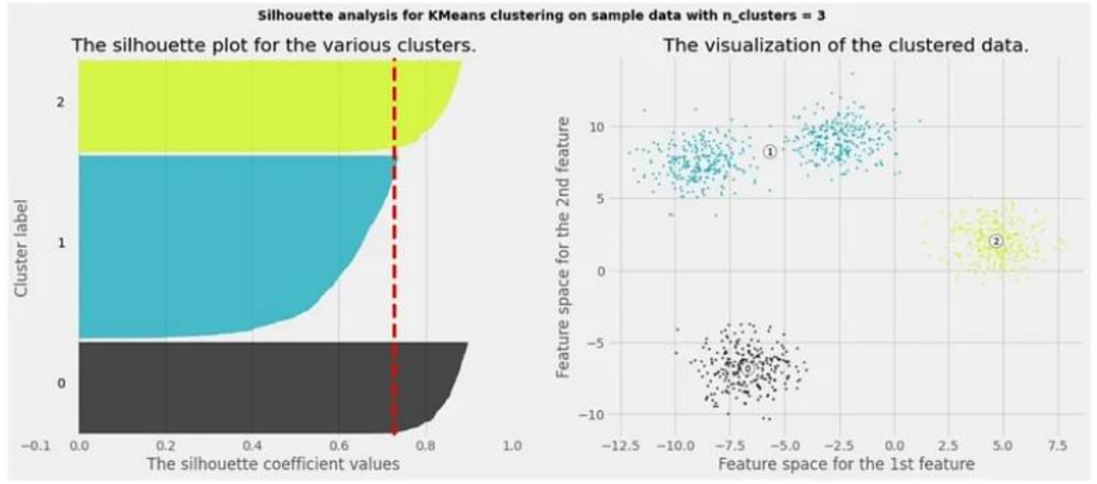
Similar to the previous Elbow method, we pick a range of candidate values of k (number of clusters), then train K-Means clustering for each of the values of k. For each k-Means clustering model represent the silhouette coefficients in a plot and observe the fluctuations and outliers of each cluster.

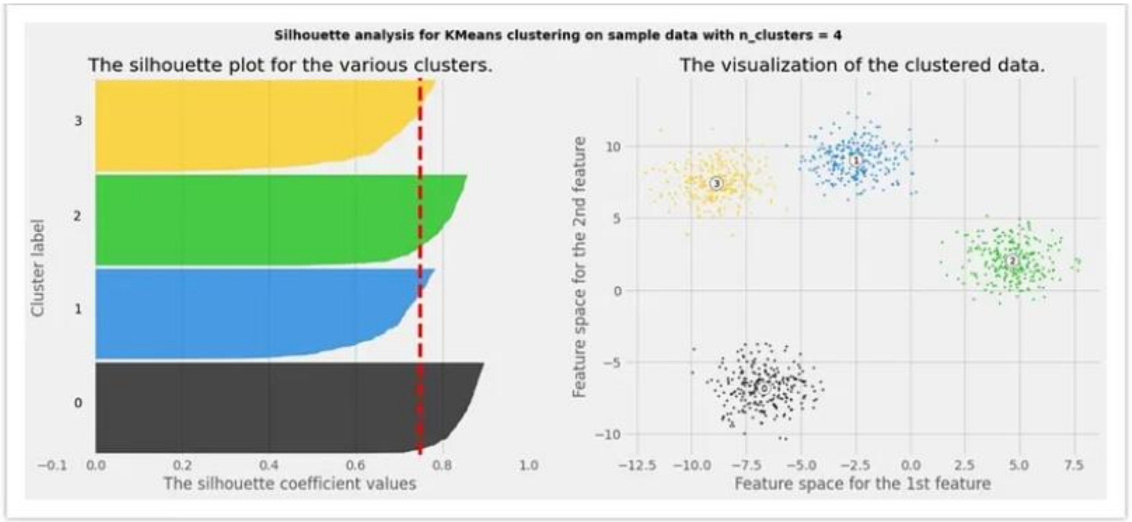
Top highlight

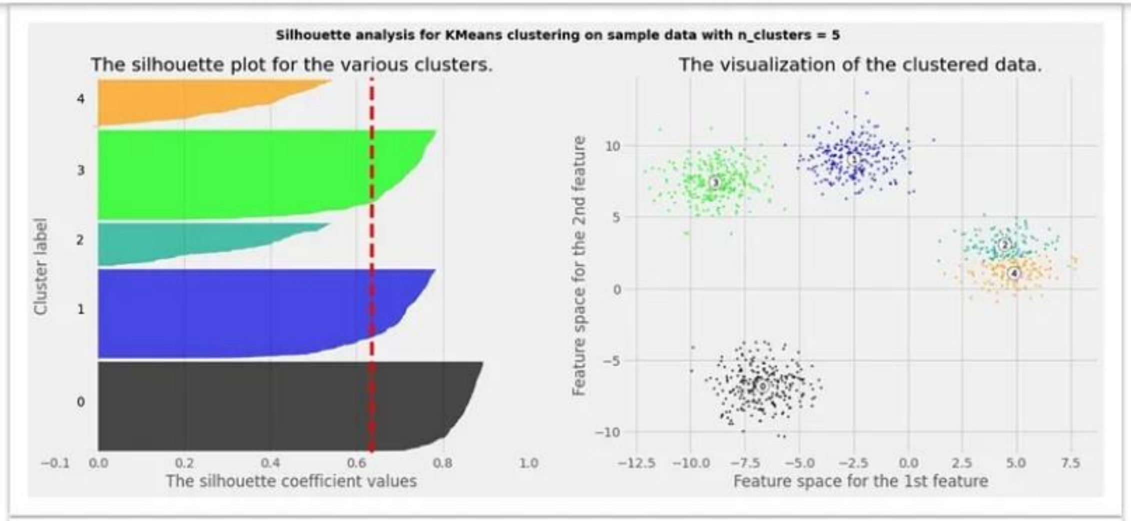


(Image by Author), Silhoutte score vs the number of clusters


Silhouette analysis for KMeans clustering on sample data with n_clusters = 2
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 1st feature
Feature space for the 2nd feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 3
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 1st feature
Feature space for the 2nd feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 4
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 2nd feature
Feature space for the 1st feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 5
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 2nd feature
Feature space for the 1st feature

K-means summary

- Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and
  - other clustering algorithms have their own lists of weaknesses (C-Means, Dbscan,…)
- No clear evidence that any other clustering algorithm performs better in general
  - although they may be more suitable for some specific types of data or applications.
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!