GBDi
Government Big Data Institute
สถาบันส่งเสริมการวิเคราะห์และบริหารข้อมูลขนาดใหญ่ภาครัฐ (สวข.)

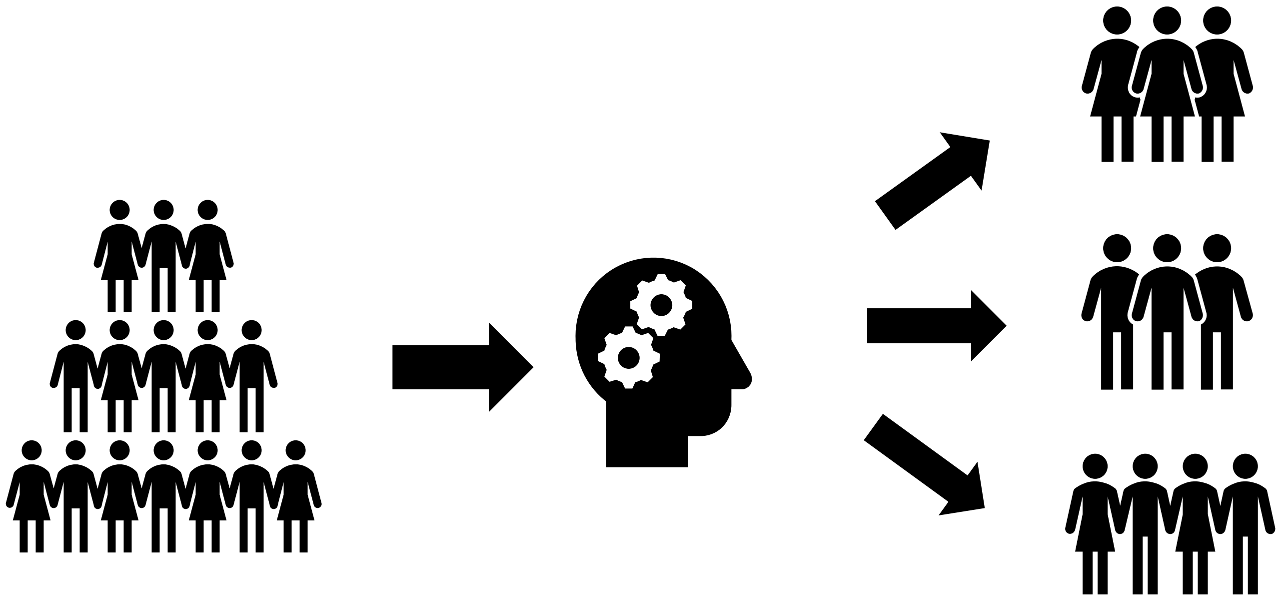
depa

Introduction to Unsupervised Learning: Basic Clustering

Patipan Prasertsom

Unsupervised Learning

- Machine learning tasks that attempts to uncover an underlying truth from the data.
- In other words, "given the data, what information can we extract from it?"
- Example tasks: Clustering, Dimensionality Reduction

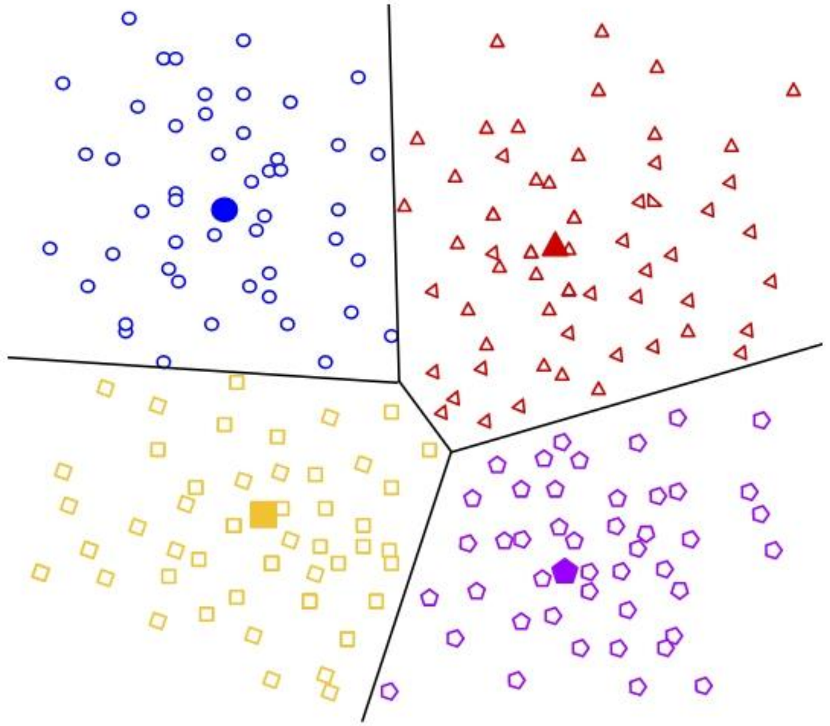


Clustering

- Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.
- It is basically a collection of objects on the basis of similarity and dissimilarity between them.

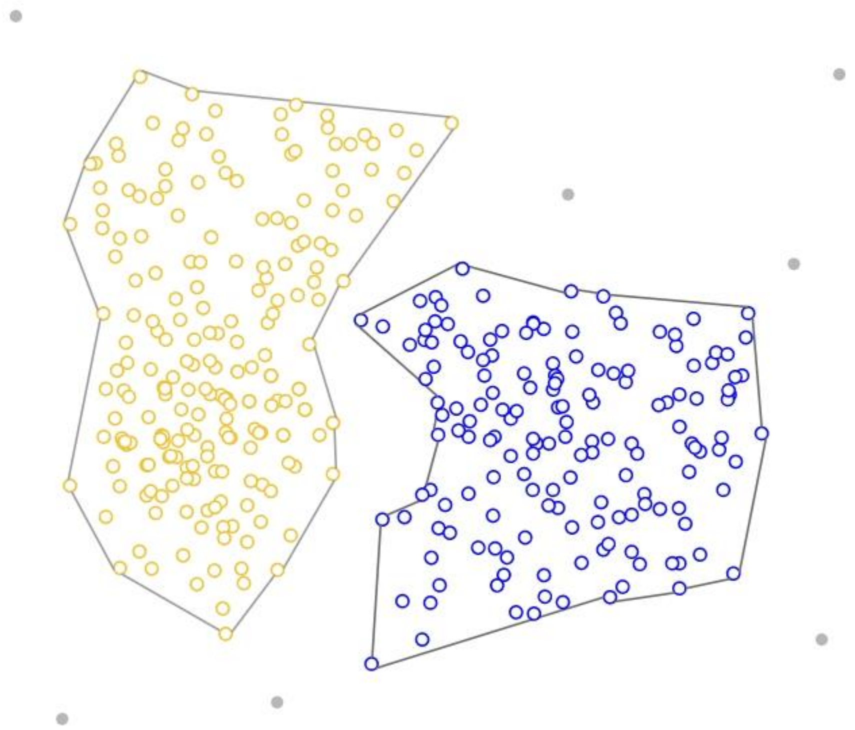
Types of Clustering

- Centroid-based clustering organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering.
- Centroid-based algorithms are efficient but sensitive to initial conditions and outliers.



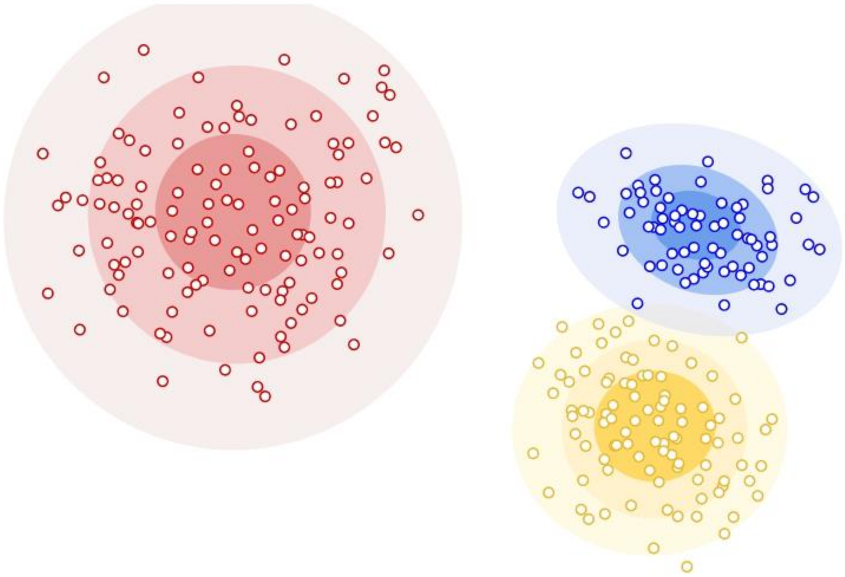
Types of Clustering

- Density-based clustering connects areas of high example density into clusters. This allows for arbitrary-shaped distributions as long as dense areas can be connected.
- These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters.



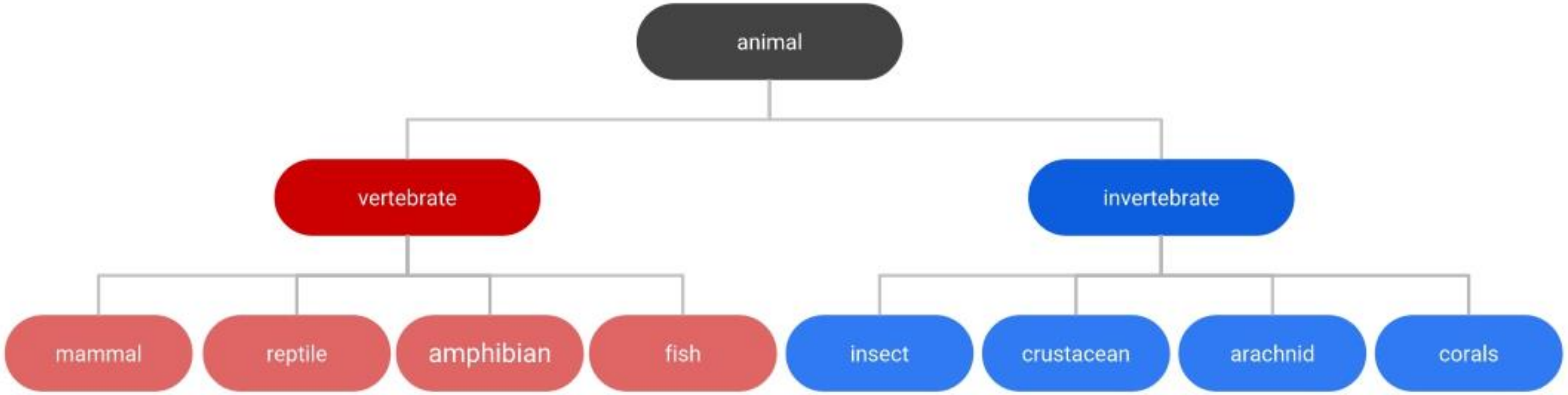
Types of Clustering

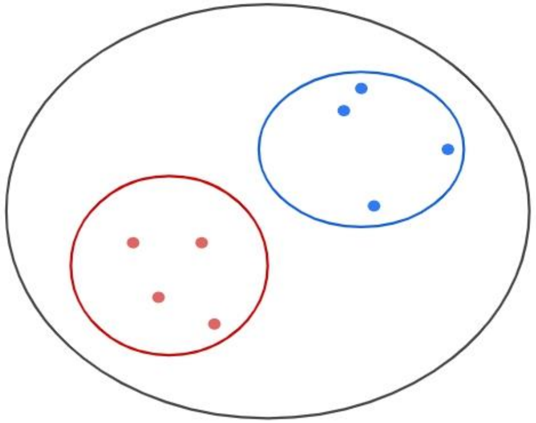
- Distribution-based clustering, as distance from the distribution's center increases, the probability that a point belongs to the distribution decreases. The bands show that decrease in probability.
- When you do not know the type of distribution in your data, you should use a different algorithm.



Types of Clustering

- Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies.
- In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.

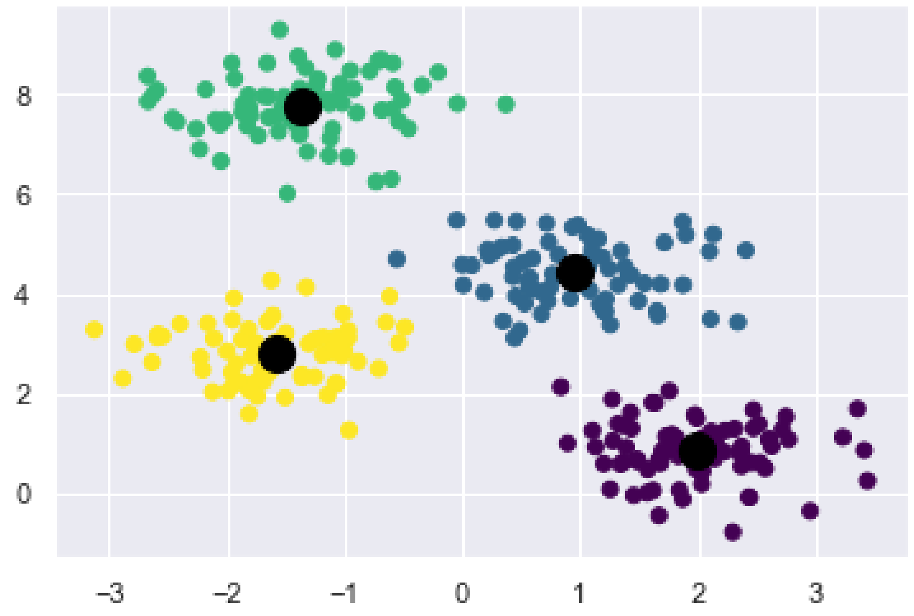




k-Means

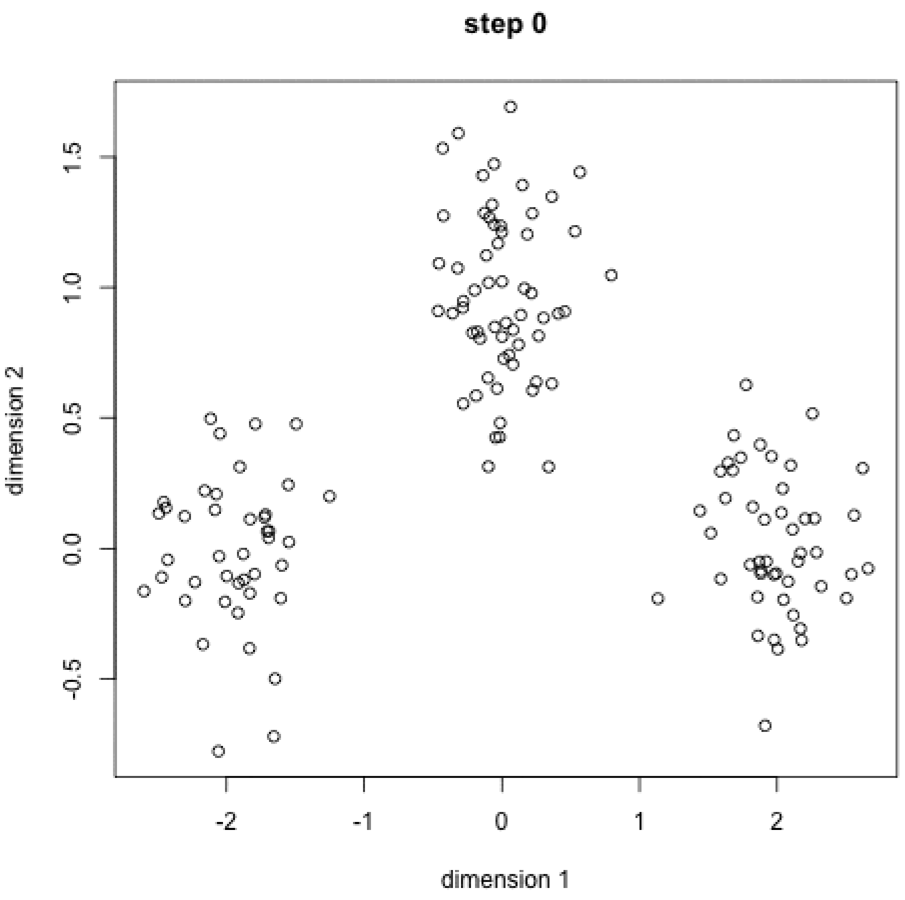
k-Means

- Organize data into groups in such a way that the data within each groups are as close to the centroids as possible.
- *The idea: "data that are close to each other are likely to be similar and form natural groups"*
- Pros:
  - Simple, fast, and scalable
  - Results are relatively simple to explain
- Cons:
  - *k* chosen manually.
  - Dependent on values at initializations
  - Suffered from outliers and *Curse of Dimensionality*



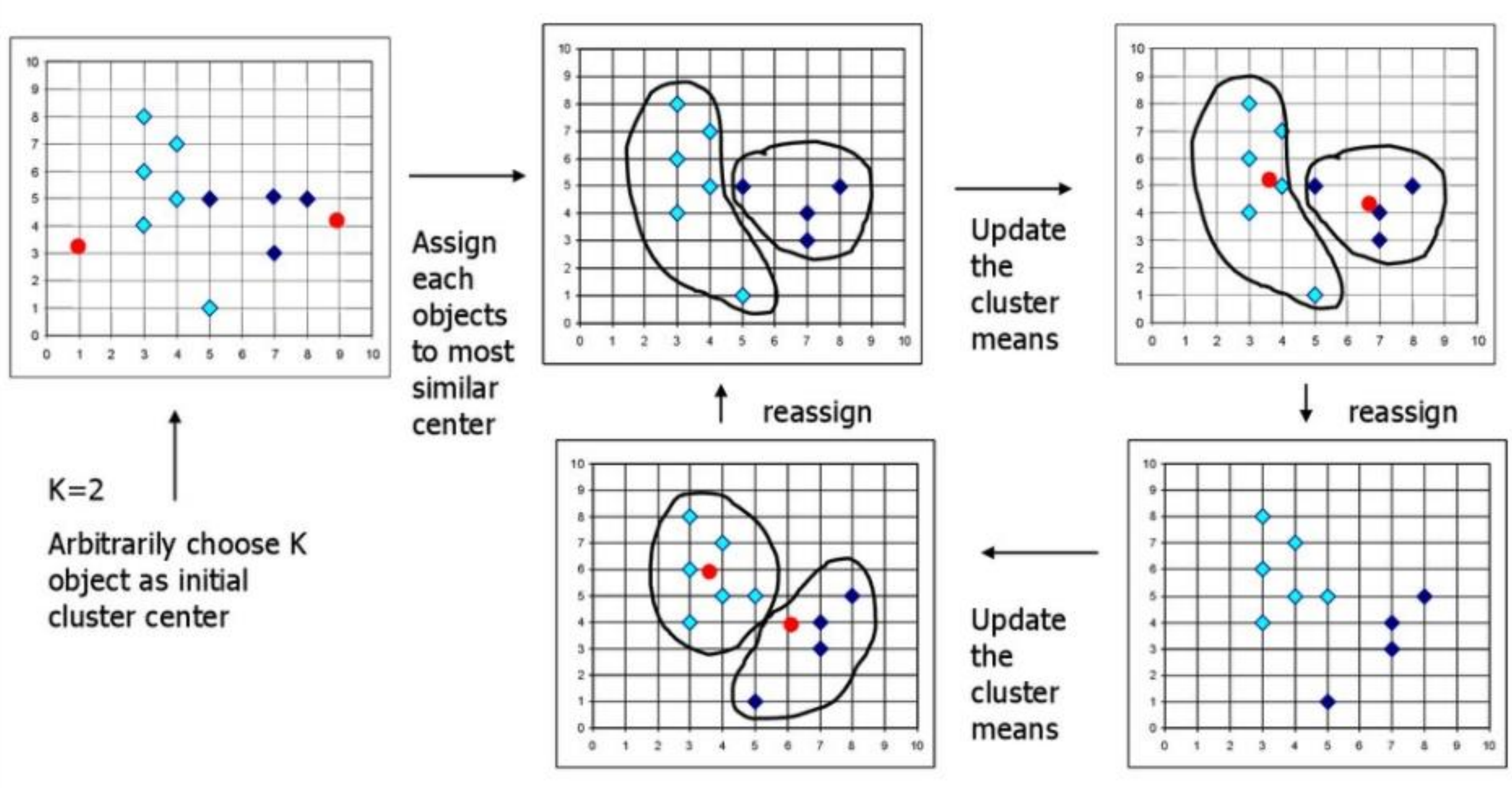
k-Means: the Algorithm

1. Partition object into k nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)
3. Assign each object to the cluster with the nearest seed point
4. Go back to Step 2, stop when no more new assignment



Demo:
https://stanford.edu/class/engr108/visualizations/kmeans/kmeans.html

k-Means



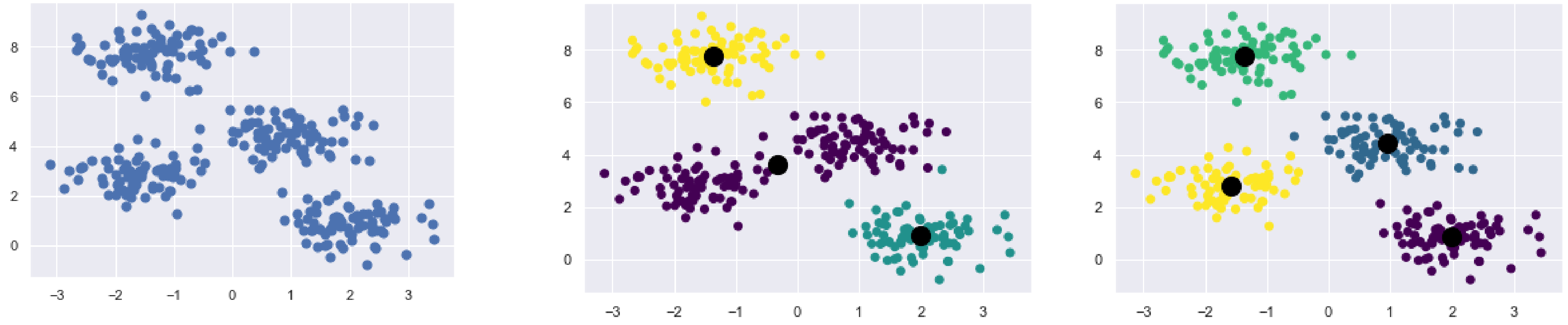
K=2
Arbitrarily choose K object as initial cluster center
Assign each objects to most similar center
Update the cluster means
reassign
reassign
Update the cluster means

k-Means Example

- Colab!

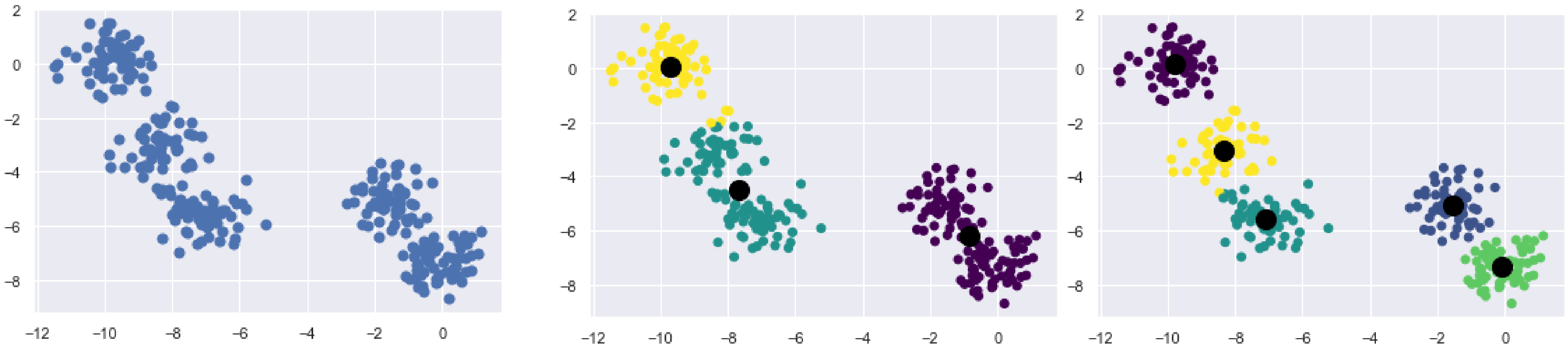
What are the proper clusters?

- k-Means will try to pick the most appropriate clusters for a given *k*



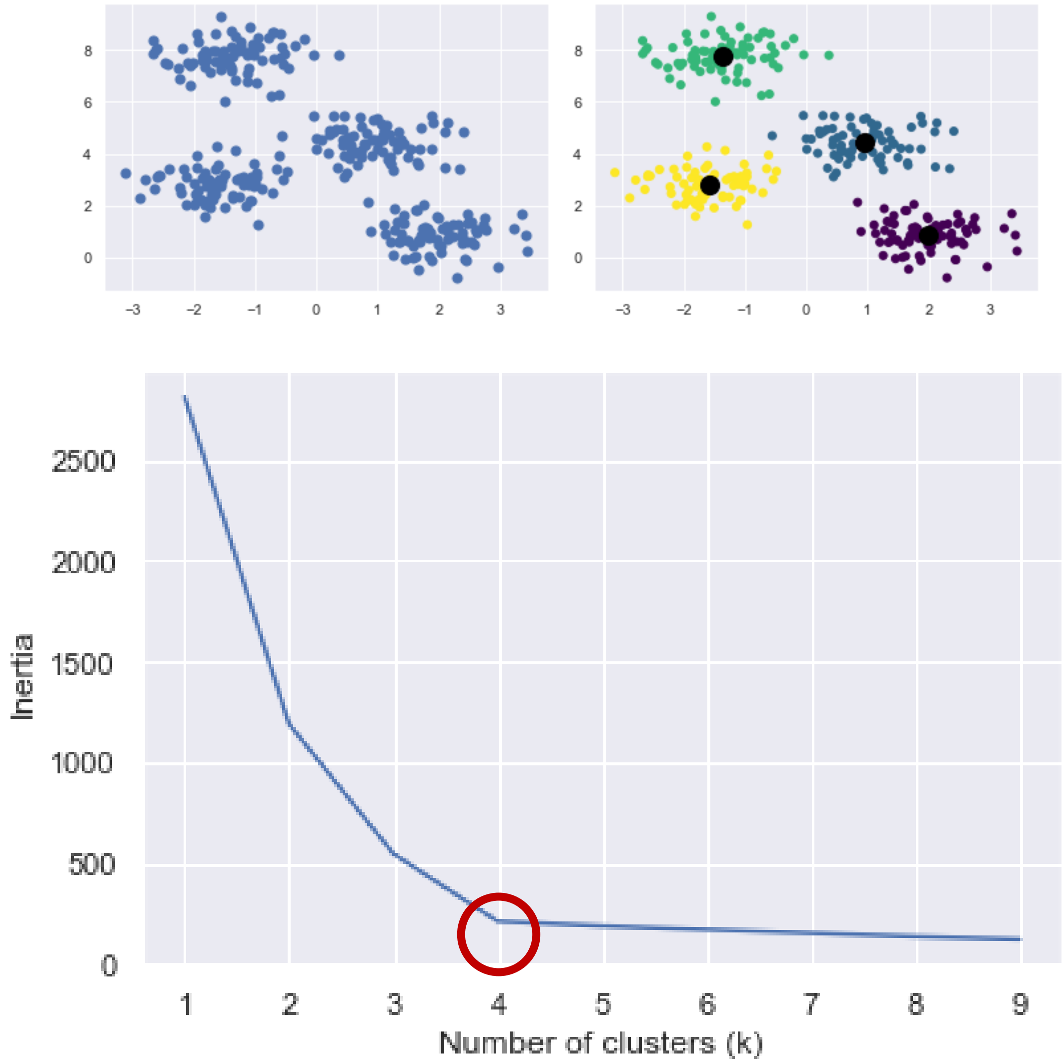
Which one is correct?

- What about this? How many clusters are there?



How can we know?

How do we decide?: The elbow method



- To decides on number of clusters, we observes the overall distances from each point to its cluster center
- In this case, we plot the inertia of each clustering result
  - Inertia: sum of squared distances of samples to their closest cluster center
- Pick *k* at the “elbow’ where splitting into more groups no longer provides noticeable improvements

Note on picking proper clusters

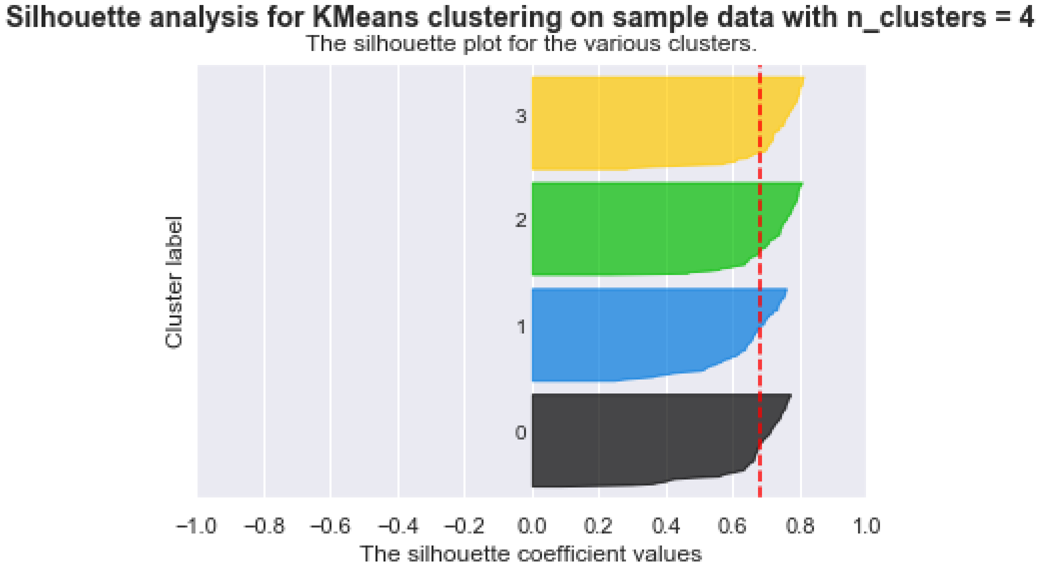
- Questions: I picked the number of groups at the elbow. Am I done?
- Answer: Maybe? There isn’t always a clear *right* answer.
- Good clusters are clusters that provide us with *actionable* insights
- Sometimes, the magnitude of the overall distances from centroids
- may not be the most representative measure for clustering.
  - A lot of time, it would also be helpful for us if we can *look* at it
    - However, when data is in higher dimensions, visualization isn’t always an option
  - There are many other metrics that can help us evaluate clusters.
    - One example of these metrics is *Silhouette Score*

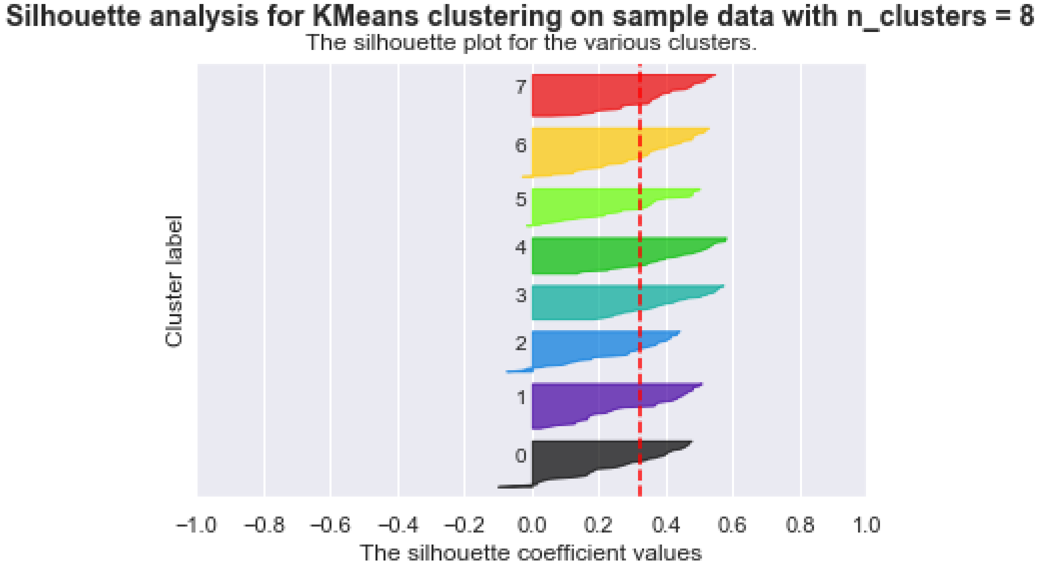
Silhouette Score

- **Silhouette Coefficient** is a measure of the level of *cohesion* of a cluster, given by a formula

$$s = \frac{b-a}{\max(a,b)} \text{ ,where}$$

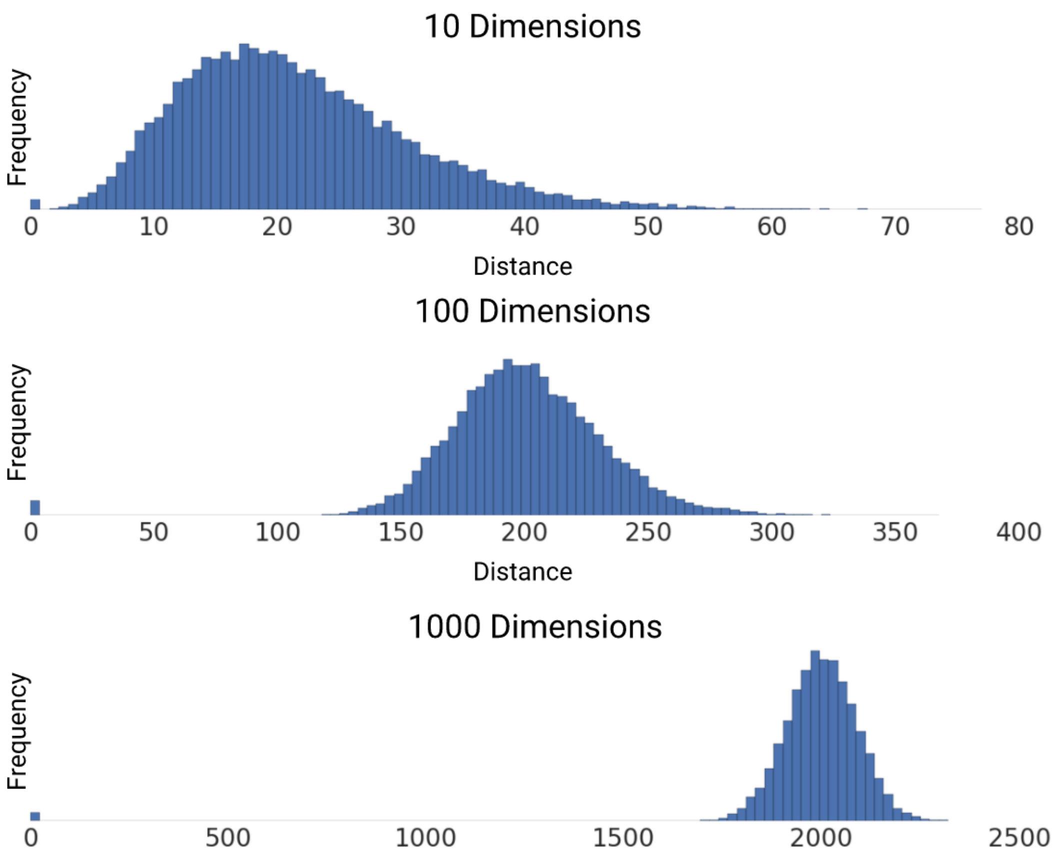
  - a is the mean intra-cluster distance
  - b is the mean nearest-cluster distance
- **Silhouette Score** is the mean of Silhouette Coefficient of all samples.
- The more positive, the better!
  - More positive generally means that the groups are tighter and better separated
- Can be helpful for visualization





Curse of Dimensionality

- The ratio of the standard deviation to the mean of distance between examples decreases as the number of dimensions increases.
  - Relative to the magnitude, the data points appears more tightly packed.
- This means it becomes harder to distinguish between examples
- In this case, more work needs to be done on k-Means to avoid this issue
  - *Spectral Clustering* (not covered in this class) is one of such methods



Additional Note

- Standard *k*-Means with Euclidean distance measure requires data to be numerical
- For categorical data, *k*-Modes (utilizing modes and Hamming Distance) can be used instead.
- *k*-Prototype is a mixture of *k*-Means and *k*-Mode. Therefore, it is applicable for situations with mixed type of data

k-Means Example

- Colab!

k-Means Exercise

- Colab!

Thank You