EECS 16ML K-means Project

Team DarkMagic:
Kunal Adhia
Kevin Cruse
Deepshika Dhanasekar
Rayna Kanapuram

Introduction to K-Means Clustering

K-Means Clustering

Overview

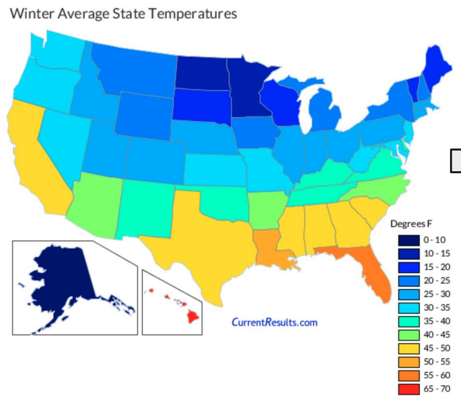
Motivation

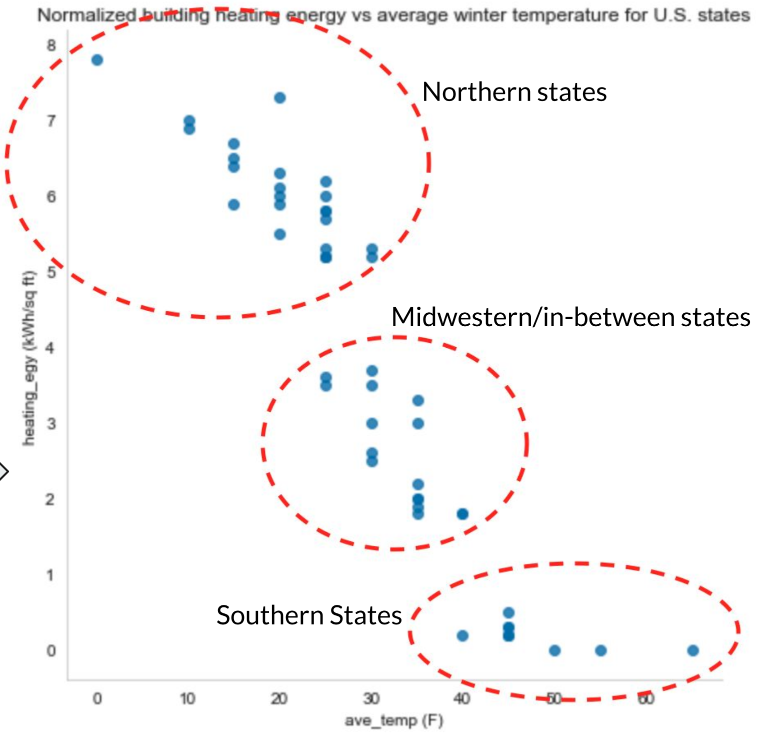
Before feeding a dataset into any predictive or classification algorithms, it is useful to *explore* the dataset and see how the data points are separated.

- Sometimes, the data points will already lie in distinct portions of a scatter plot based on dataset
- Most of the time, the data will be too rich with features to immediately discern any clustering pattern, so we will need to apply some dimensionality reduction first

Motivation

- Sometimes, the data points will already lie in distinct portions of a scatter plot based on dataset





Motivation

- Most of the time, the data will be too rich with features to immediately discern any clustering pattern, so we will need to apply some dimensionality reduction technique first (like PCA)

Motivation

- With a set of distinct clusters, we make the assumption that the given cluster of data points is distributed <u>equidistantly</u> about a mean (this would be circular or spherical if we're working in 2 or 3 dimensions, respectively)
- The goal of K-means clustering is to determine those means

Quick Note: Unsupervised Clustering

- When clustering, we usually do not have access to the labels of the data points

Otherwise, we wouldn’t need to use K-Means!

- This means K-Means is an *unsupervised* clustering method

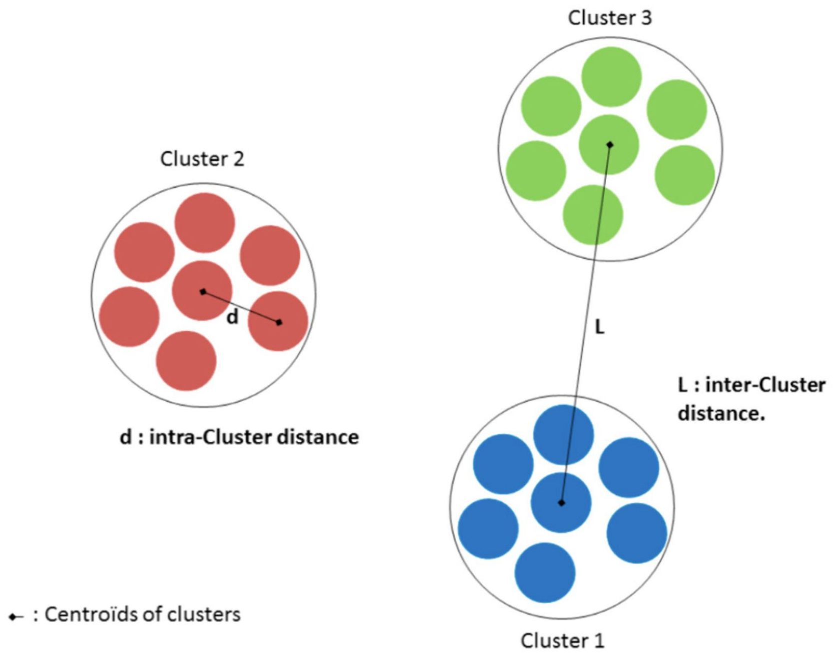
Choice of Cost

- Once we have our data reduced to an appropriate number of dimensions, we systematically infer the mean for each cluster (shown formally later)
- Since we do not have access to the actual labels of the data, how do we know when to stop K-Means? Impose two costs:
  - Inter-cluster distance (point-to-point within a cluster)
  - Intra-cluster distance (cluster-to-cluster)
- We iterate our algorithm until these costs no longer change significantly

How K-Means Works

Objectives & Goals of K-Means

1. Maximize inter-cluster distance
2. Minimize intra-cluster distance



Formal K-Means

Sum over all clusters

$$\underset{\{C_k\}_{k=1}^{K},\{\vec{c_k}\}_{k=1}^{K}:X=C_1\cup\ldots\cup C_K}{\arg\min}\ \sum_{k=1}^{K}\ \sum_{\vec{x}\in C_k} \|\vec{x}-\vec{c_k}\|^2$$

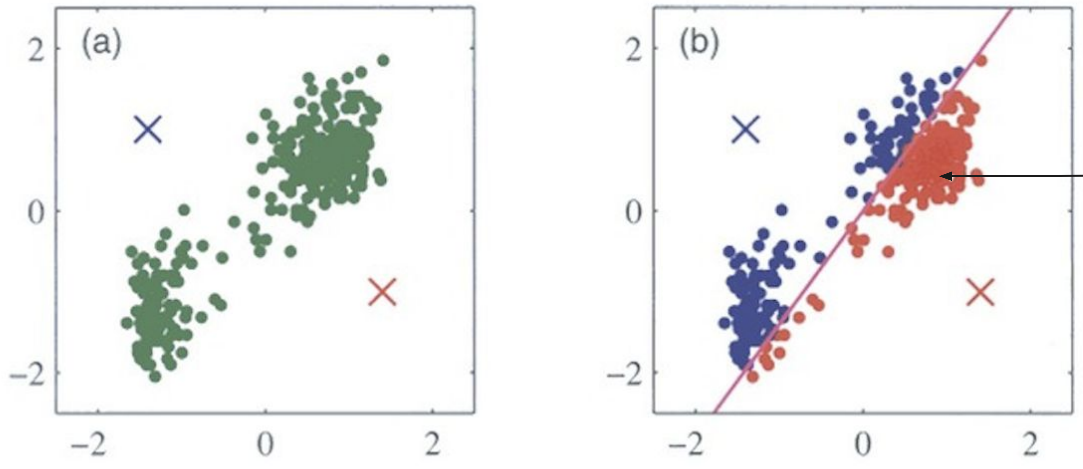
Each C_k is a set containing all the data points assigned to Cluster k

Each $\mathbf{c}_k$ is a point representing the center of Cluster k

Sum of squared distance from each point in Cluster k to Centroid k

Maximizing Inter-cluster distance

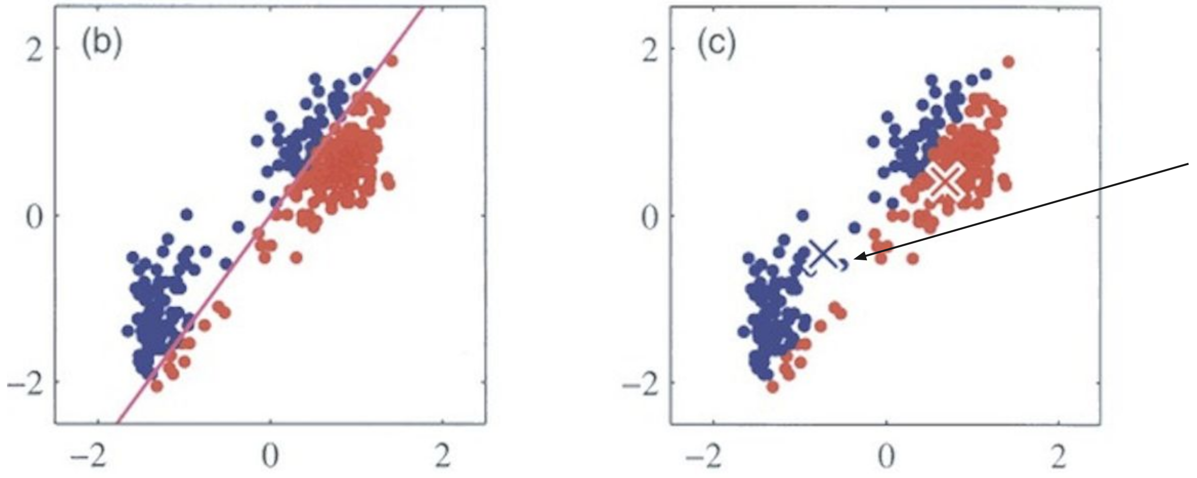
If we had centroid locations, how would we assignment points to maximize inter-cluster distance?



Assign each point to the centroid that it is closest to

Minimizing Intra-cluster distance

If we had cluster assignments, how would we pick centroids to minimize intra-cluster distance?

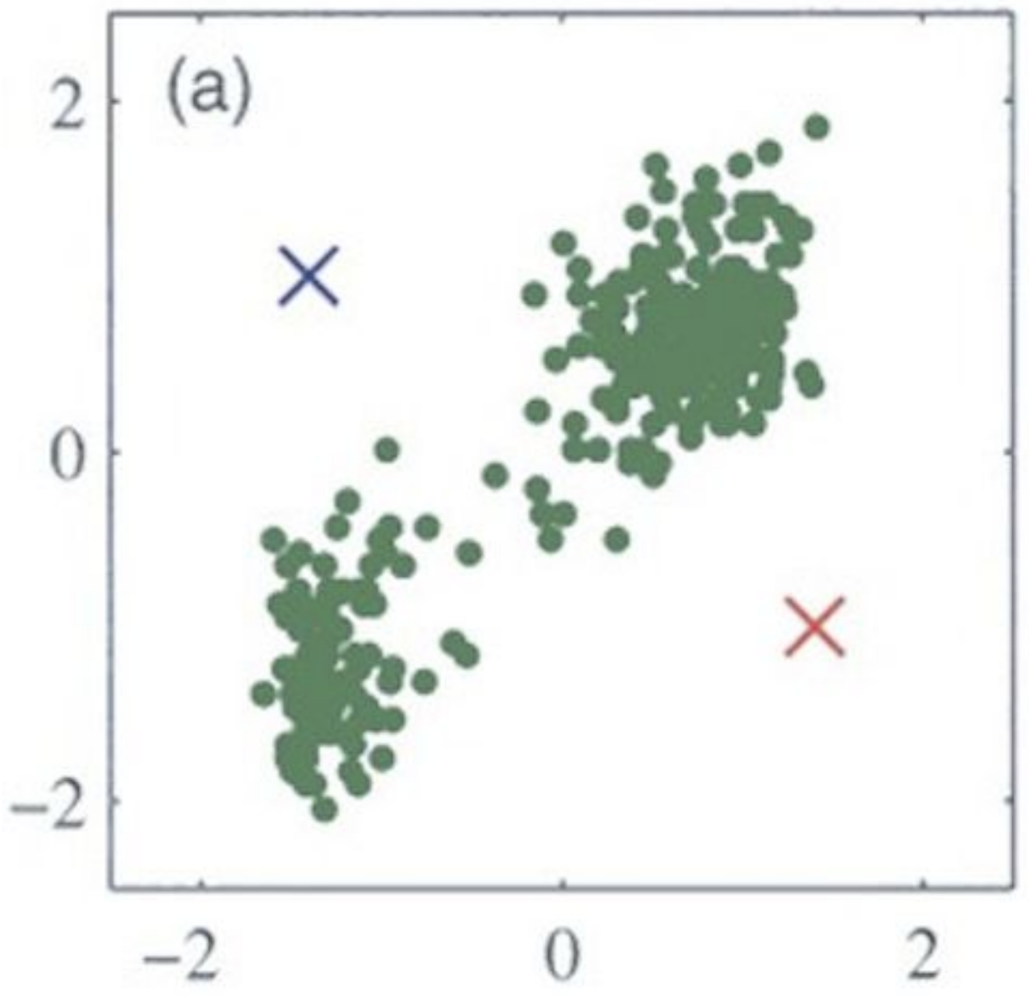


Set the new cluster centroids to be the average of all the points assigned to that cluster

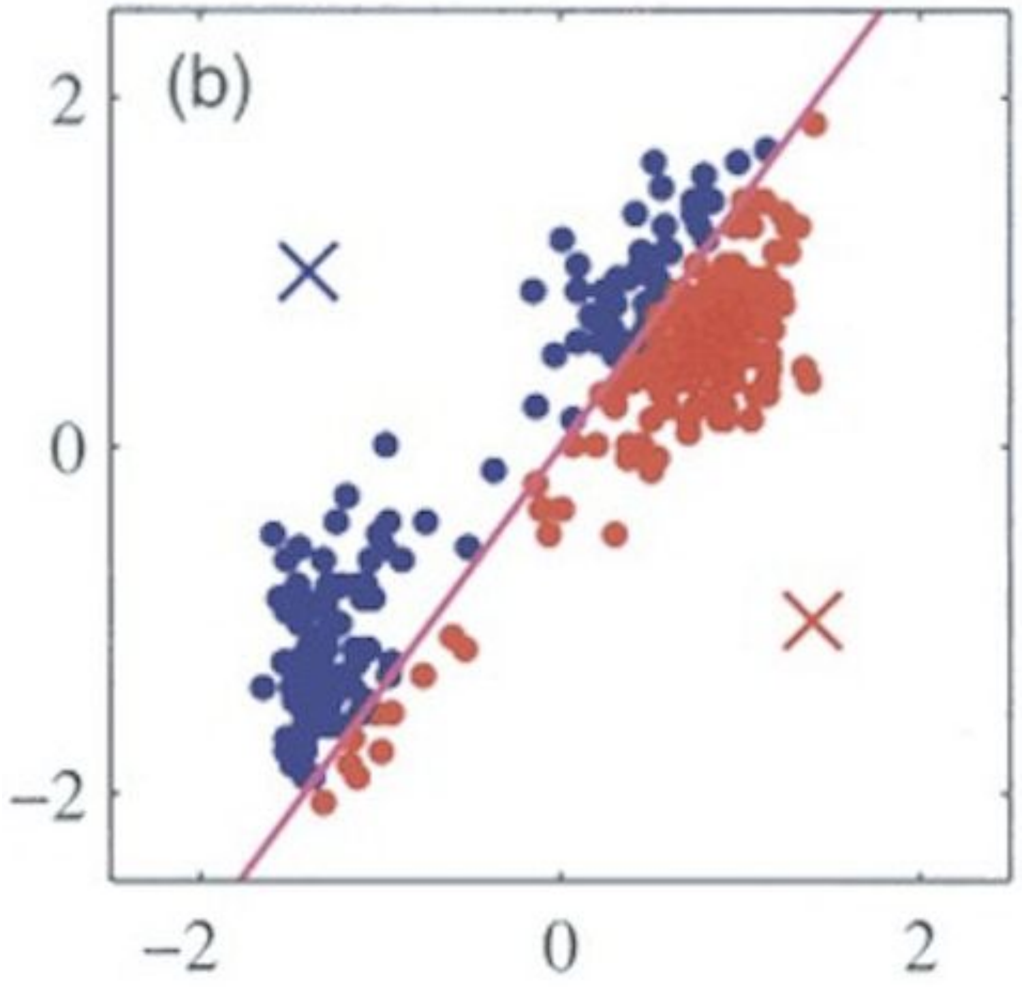
Efficient K-Means Algorithm

1. Initialize random centroids
2. While K-Means has not converged:
    a. Given the current centroids, find the best cluster assignments
    b. Given the new cluster assignments, find the best centroids

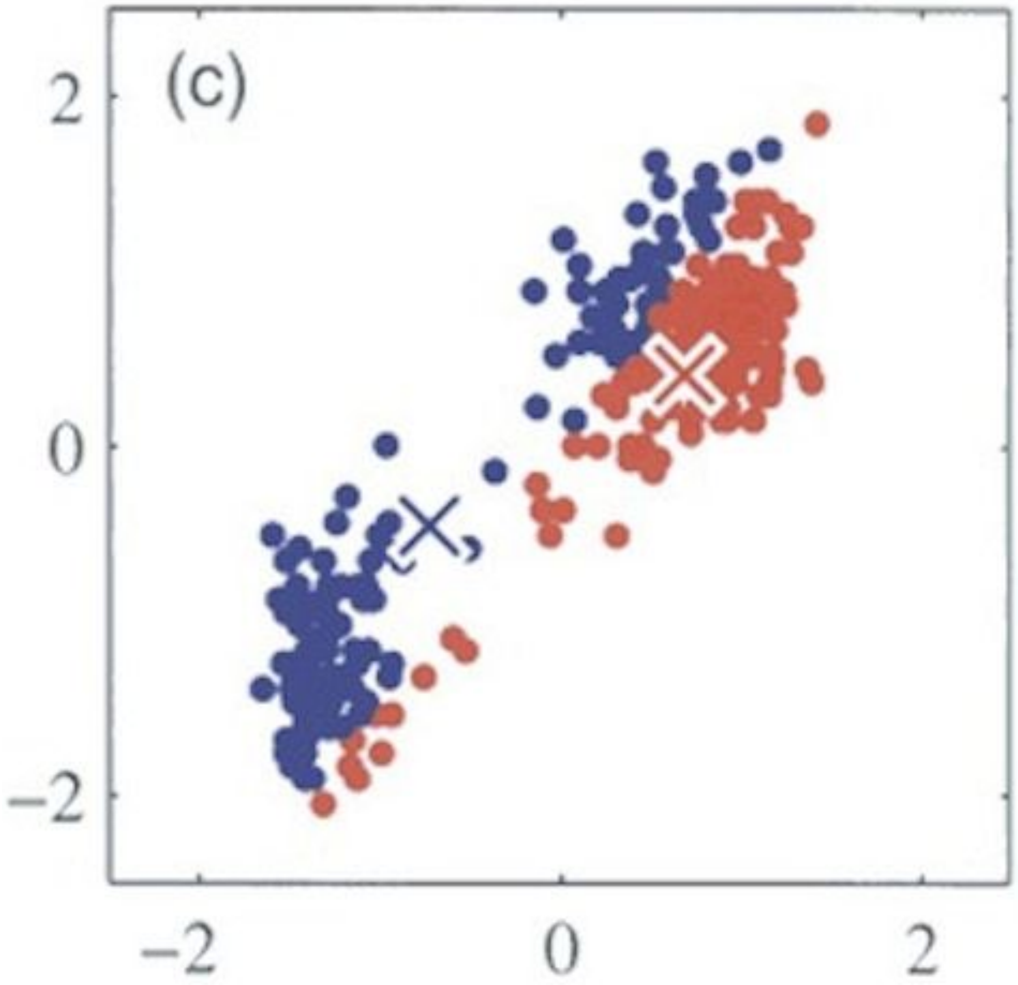
Initialize Centroids



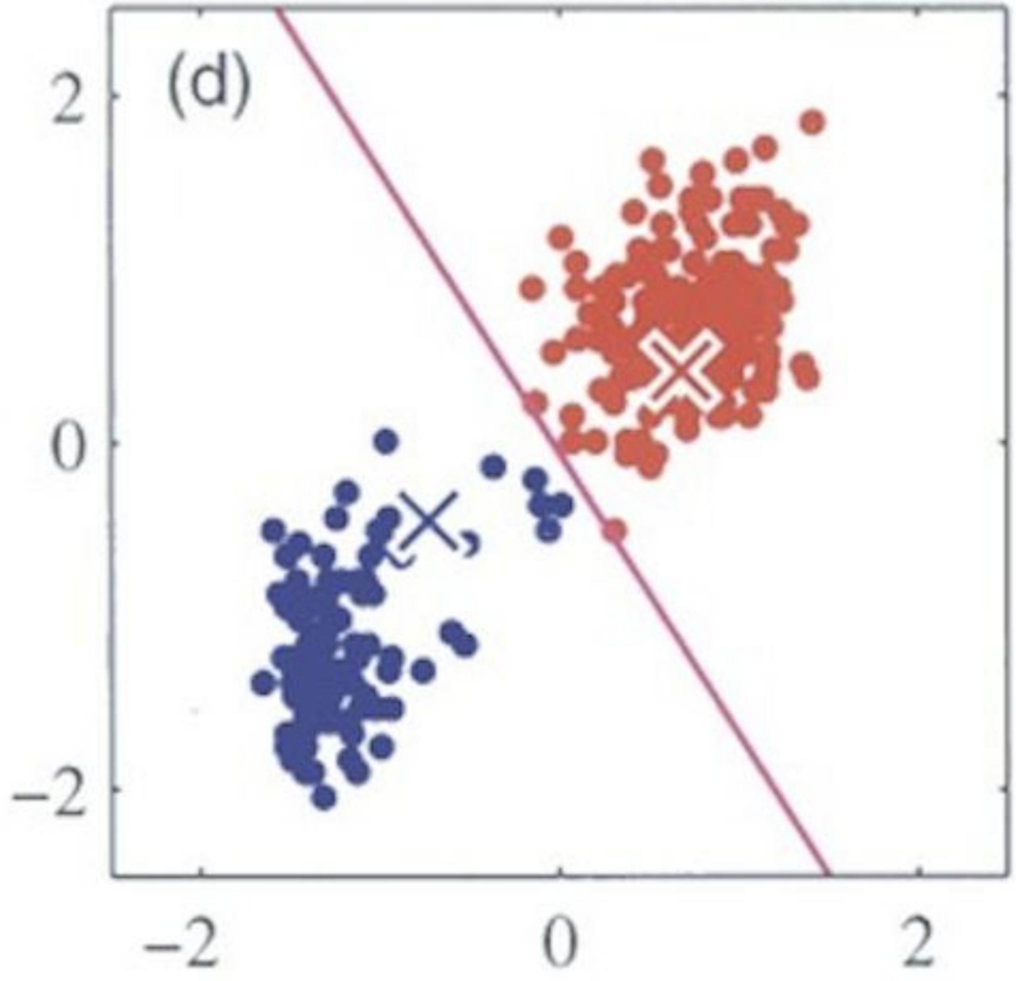
Assign Points to the Closest Cluster



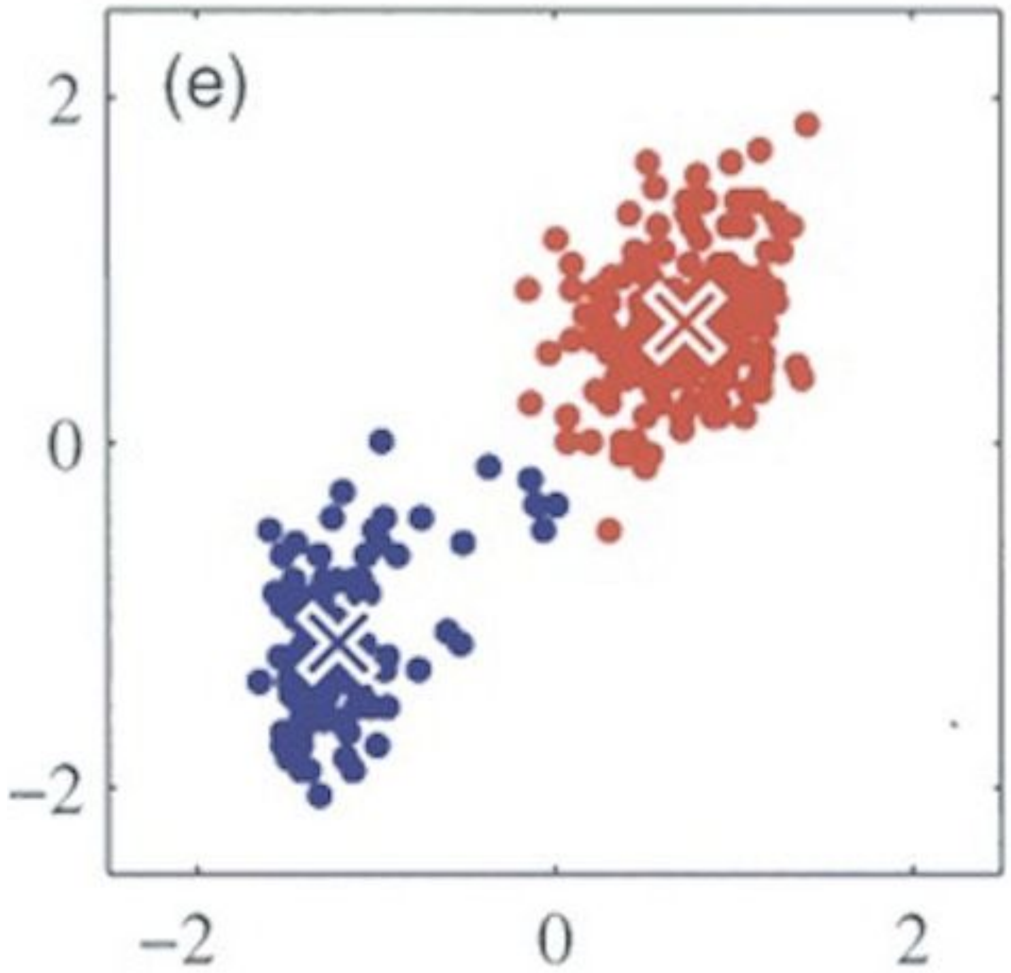
Find new centroids



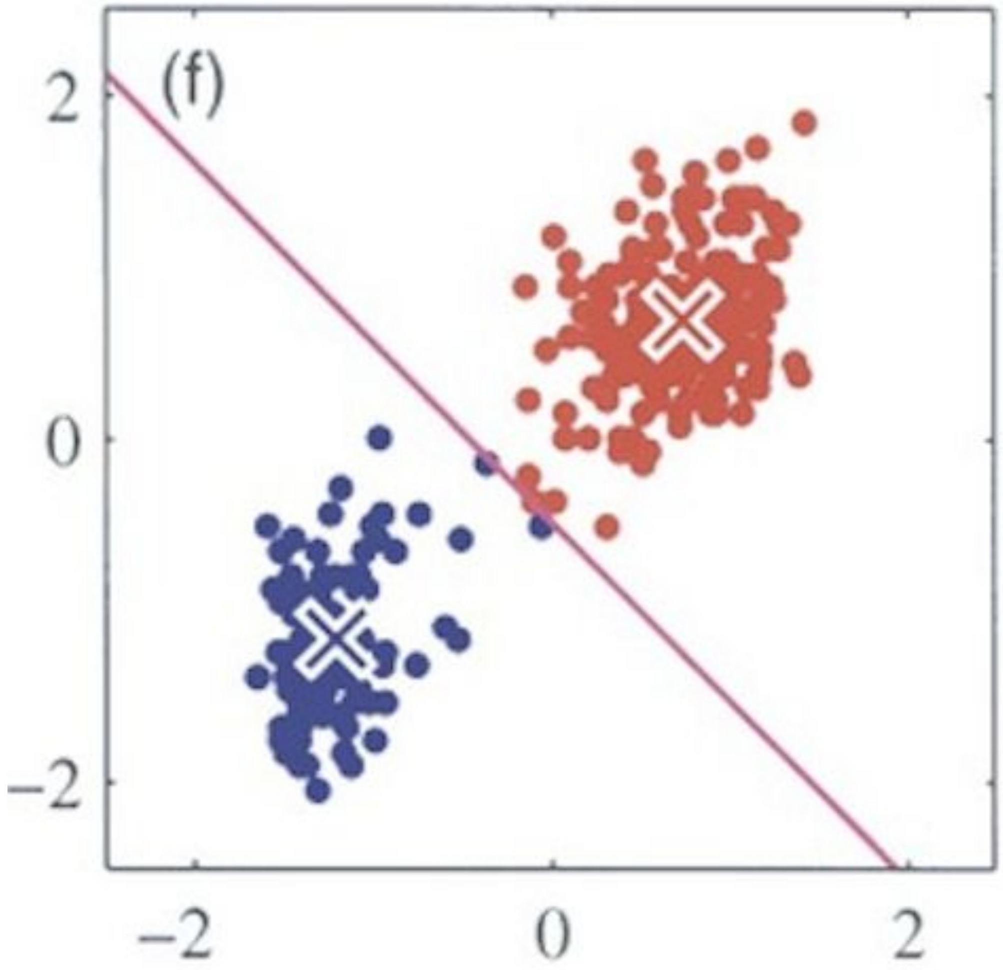
Re-assign Points to the Closest Cluster



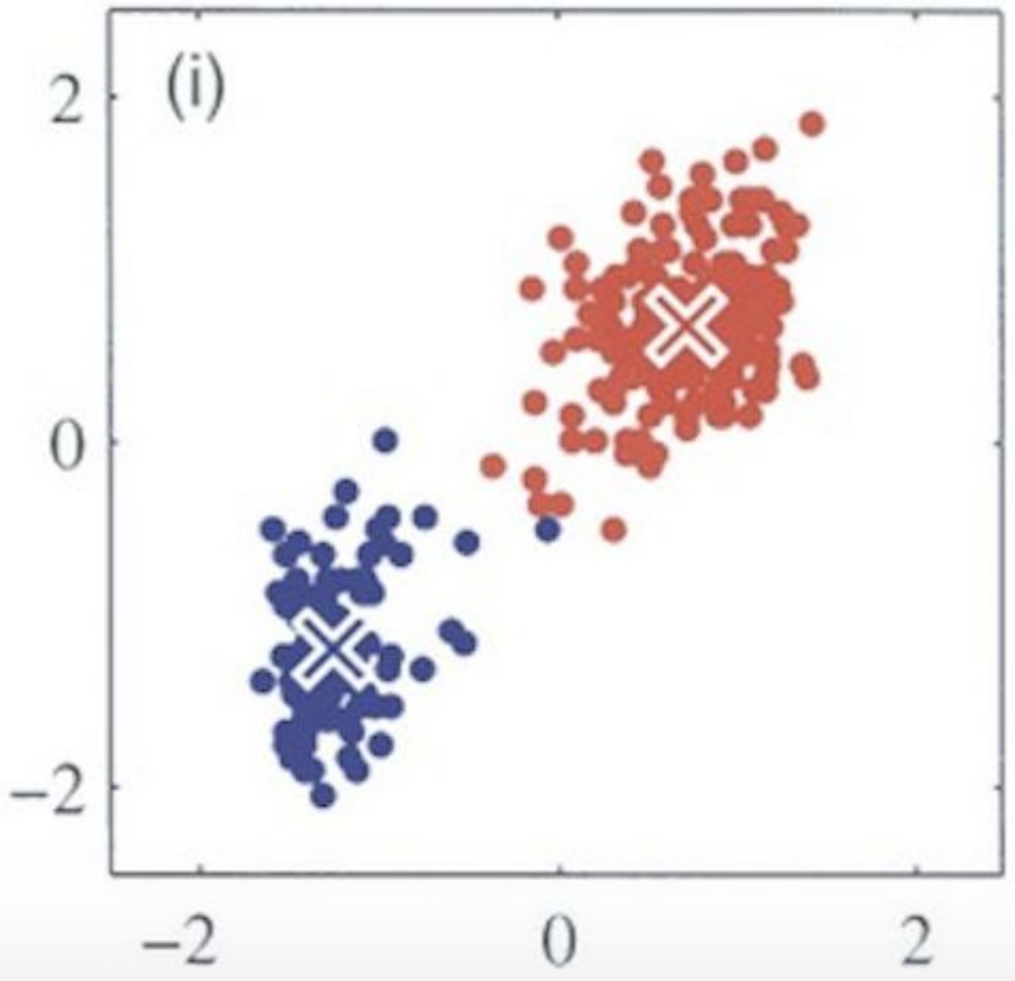
Find new centroids



Re-assign Points to the Closest Cluster



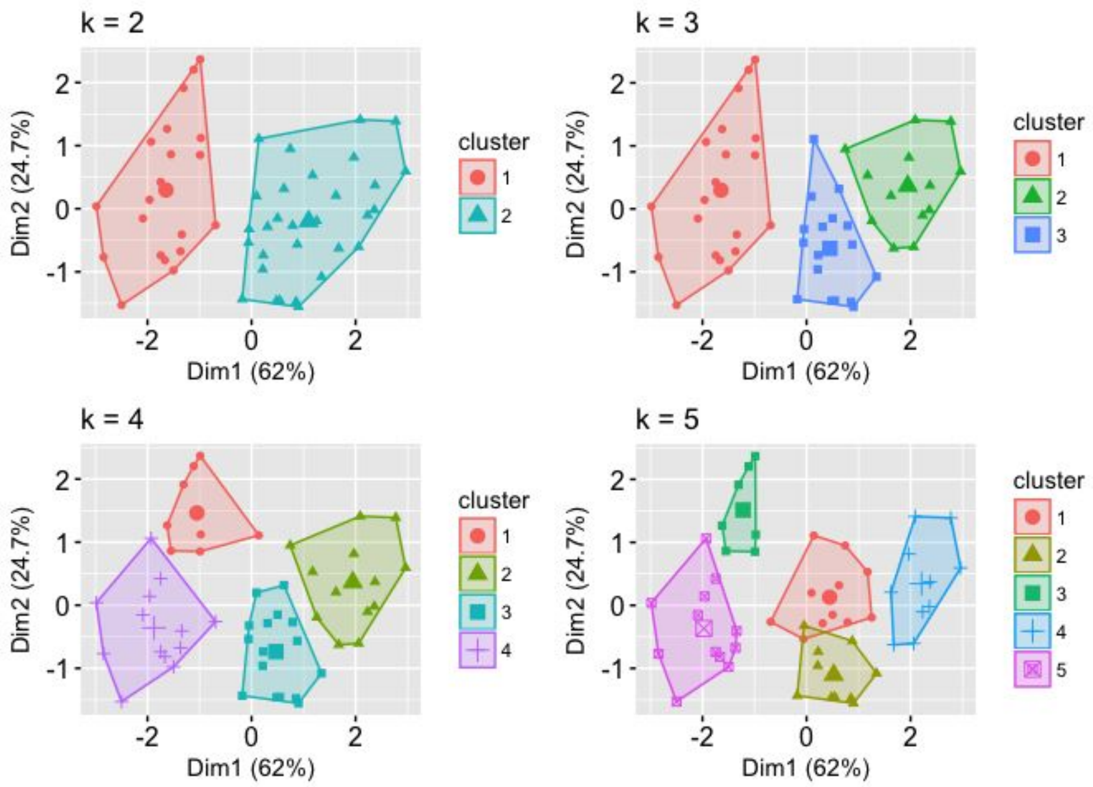
Repeat until the assignments & centroids stop changing between iterations



Tuning K-Means

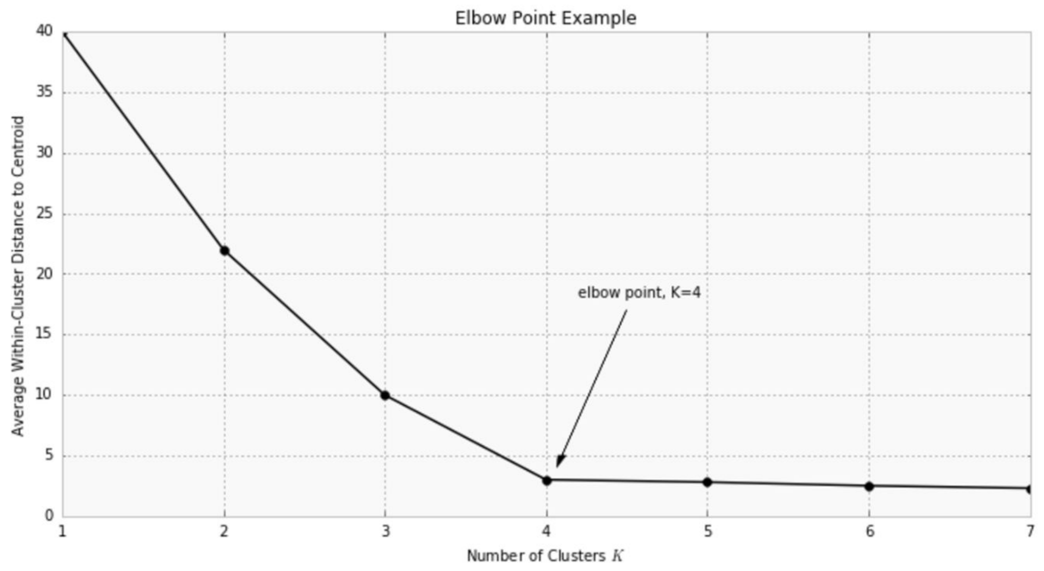
Picking “k”

- Unlabeled data means we have no reference
- Different k values can give vastly different clusters



Picking “k”

- Pick the right number of clusters k using validation and/or domain knowledge
- When using validation, graph the cost for multiple values of k & use the elbow method



Picking the right dimensionality

- Ideally, we want to cluster based on features that have the highest amount of variance between the data points we have
- Hard to visualize past 3-D, so we can’t just plot & look at which data looks the most separable
- Can use PCA to reduce the dimensionality of the data & see how K-Means performs

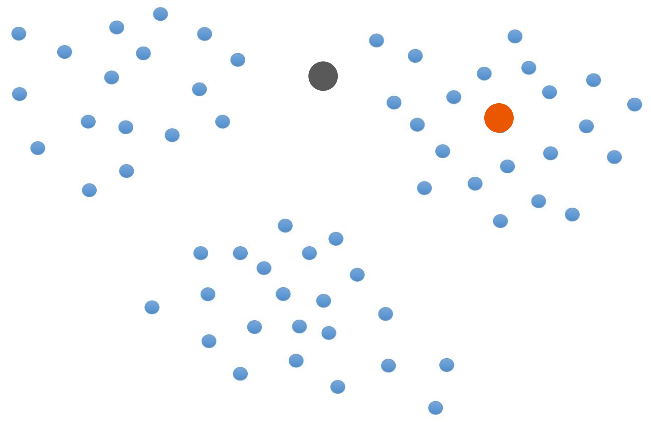
Limitations of K-Means

Limitations of K-means

There is no likelihood attached to K-means. While K-means allows us to assign a given data point to a cluster. We have no information about the likelihood or confidence that the assignment is correct.

K-means will tell us that the grey and orange points both belong to the top right cluster.

But, we won’t have information about the likelihood that the grey point belongs to the top left cluster instead (even though it is close to the boundary).

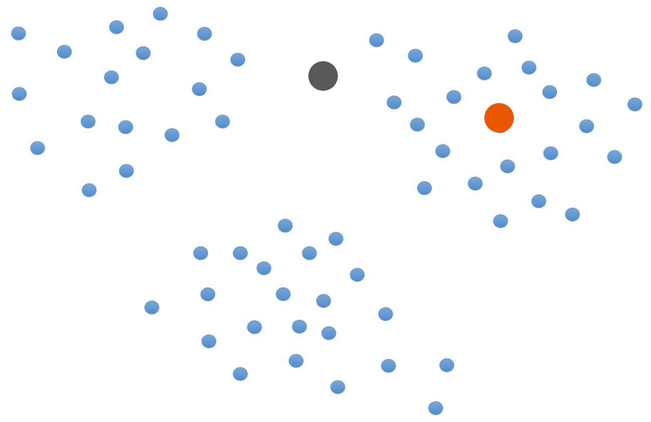


Limitations of K-means

The K-means algorithm is based on hard assignments. Each point belongs to exactly 1 cluster. A soft assignment would allow us to assign each point to a distribution over the clusters.

In the case of soft assignments, we would be able to know the probability that each point belongs to every cluster.

In our toy example, an algorithm based on soft assignments may tell us the grey data point belongs to the tail ends of the top right and top left cluster distributions. This helps give us a measure of confidence in our assignment.

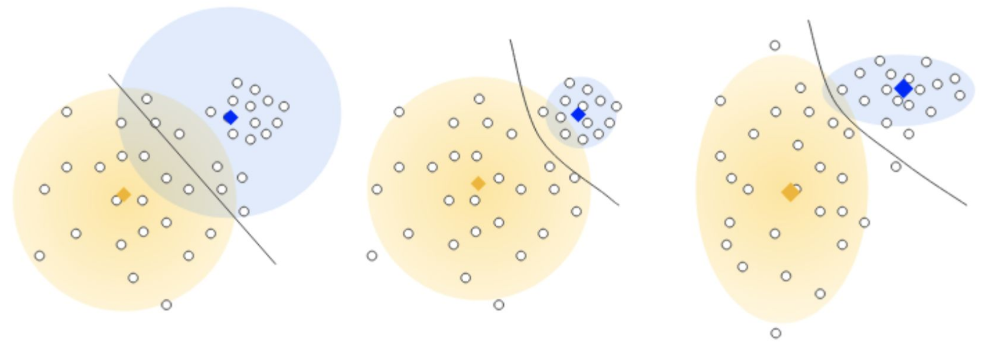


Limitations of K-means

Each feature is treated equally by K-means. This causes our output to be circular in 2-D, and spherical in 3-D (think back to formula for Euclidean distance).

As shown in the diagram to the right, being limited to spherical clusters prevents our algorithm from learning cluster boundaries as shown to the far right.

For some datasets, non-spherical boundaries may be the most sensible. But, ordinary K-means would be unable to produce such a clustering.


Plain k-means

Varying widths across clusters

Varying widths across clusters & dimensions

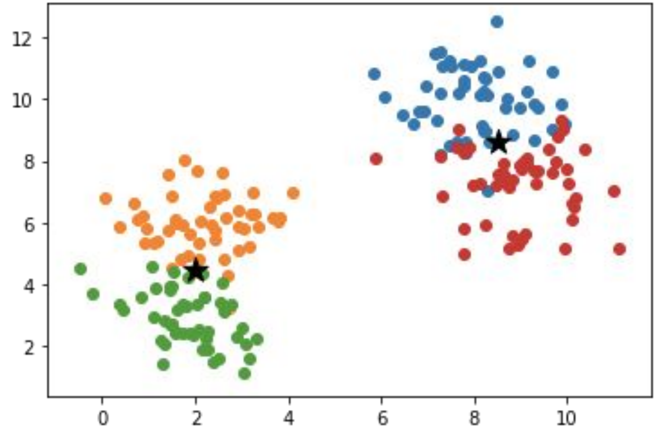
Limitations of K-means

The results of clustering are highly dependent on our choice of k.

While k=2 might appear to be reasonable, if we were looking at unlabeled data, the data is actually drawn from 4 separate clusters.

Our ability to appropriately tune k will significantly impact the results of the clustering algorithm.

In such cases, domain knowledge might be helpful to justify/explain the choice of k.



True clusters are denoted by the different colors. Stars denote the cluster centers for k=2.

Variations of K-Means

Variations of K-means: K-medians

K-medians works almost identically to k-means, but with a minor adjustment.

For k-means, clusters are determined by calculating the **mean** of each data point in the cluster by feature.

For k-medians, however, clusters are determined my calculating the **median** of each data point in the cluster by feature.

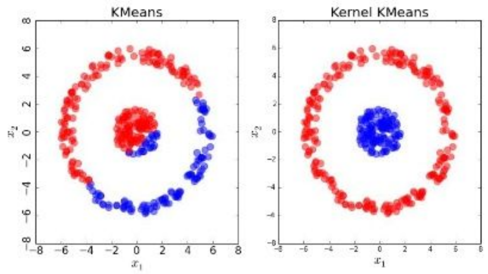
Advantage: Less prone to outliers.

Disadvantage: Each data point has less significance when determining the centroid.

Variations of K-means: Kernelized K-means

The k-Means algorithm can be kernelized by calculating inter-cluster and intra-cluster distances with inner products. In addition, as with other kernelized methods, one can specify the type of kernel.

Kernel K-means vs. K-means



Kernelized k-Means allows one to better cluster based on non-linear cluster separations with the use of non-linear kernels.

In addition, this method allows one to take advantage of the benefits of kernelization, including efficiency.

Works Cited

EECS 189 - Clustering Slides, Fall 2020

https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages