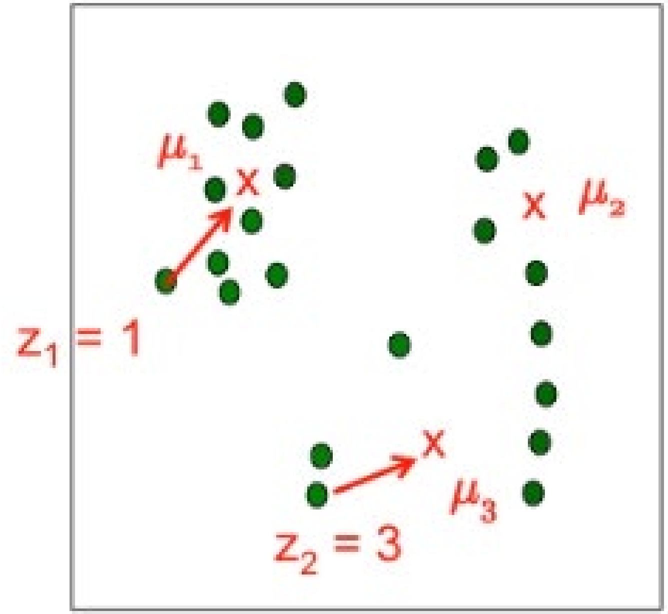
k-means clustering

- Partition data samples *k* clusters (pre-determined *k*) belonging to the cluster with the nearest mean/cluster center
- The centers serve as a prototype of the cluster
- Partitioning data space into Voronoi cells
- It minimizes within-cluster variances, the mean optimizes squared errors

K-Means Clustering

- A simple algorithm that iterates between 2 steps
  1. updating the assignment of data to clusters (sample assignment)
  2. updating the cluster's summarization (center update)

- Given the i-th data sample with features x_i
- Assume k clusters
- Each cluster c is represented by a center μ_c
- Each cluster serves as a prototype for a set of nearby samples
- Assignment of the i-th sample to an index $z_i \in \{1, 2, \ldots, k\}$
- Update the centers after assignment of all samples



K-Means Clustering

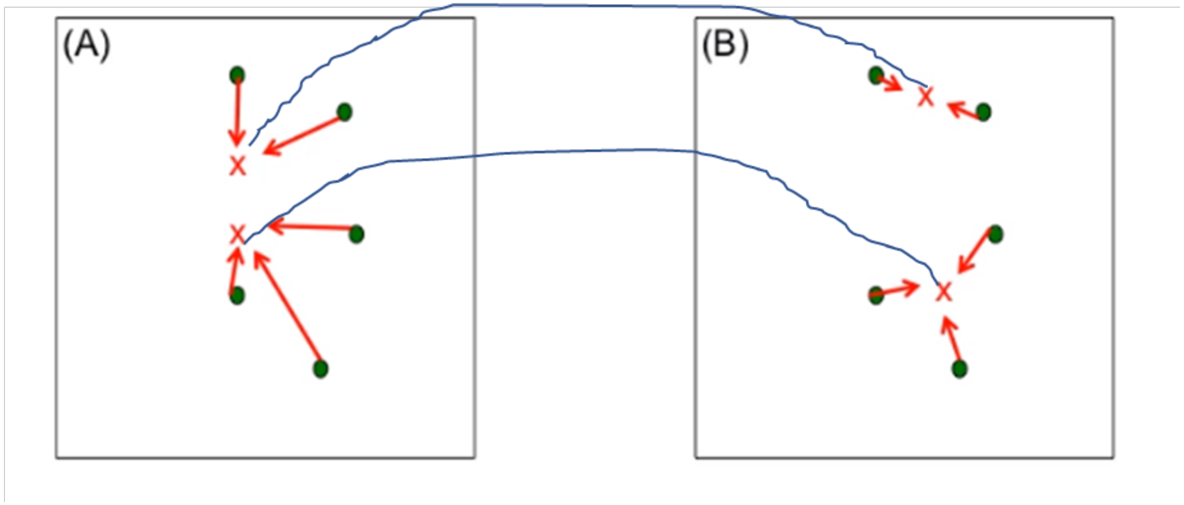
- Iterate until convergence

  1. For each sample, find the closest cluster

  $z_i = \arg\min_c \|x_i - \mu_c\|^2 \quad \forall i$

  2. Computer the mean of all assignment samples and set it as cluster center

  $\forall c \quad \mu_c = \frac{1}{m_c} \sum_{i \in S_c} x_i \qquad S_c = \{i : z_i = c\}, m_c = |S_c|$

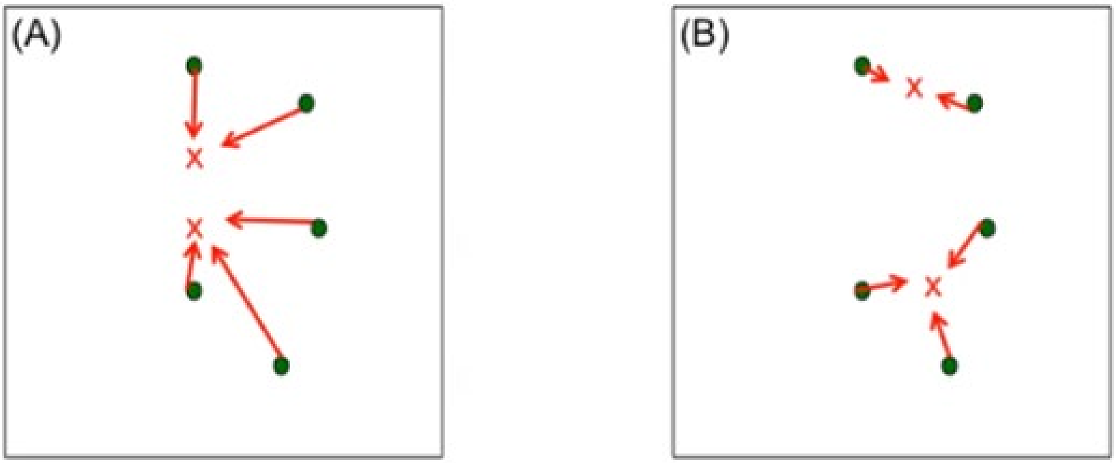


K-Means Clustering

- Cost function to be minimized:

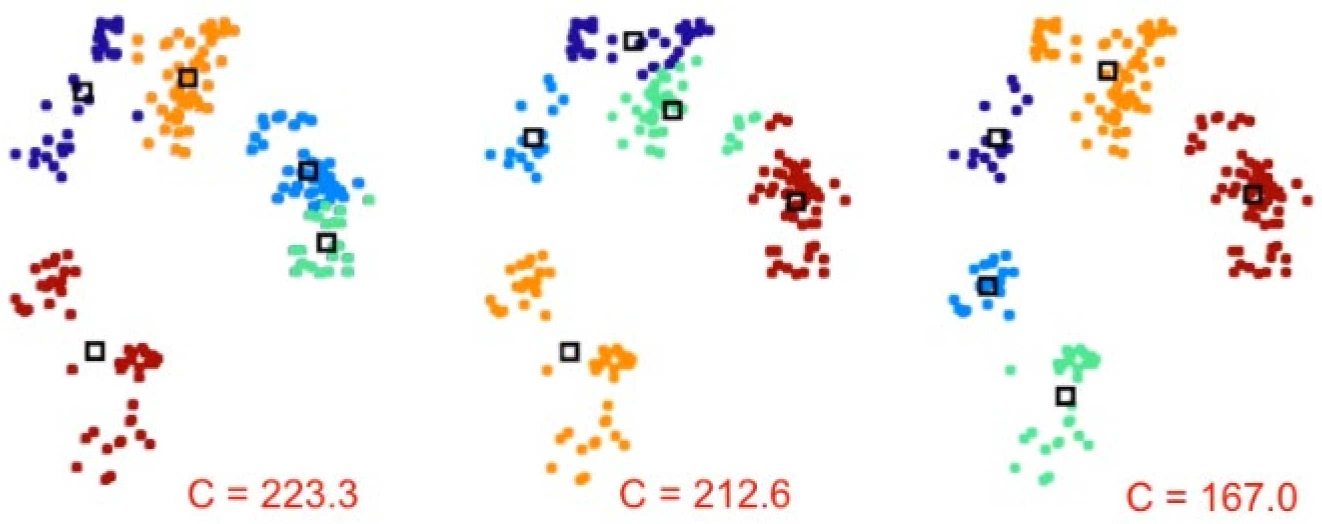
$$C(\underline{z}, \underline{\mu}) = \sum_i \left\| x_i - \mu_{z_i} \right\|^2$$

- Cost descent over the 2 steps:
  Selecting the closest center minimizes the sum each time a sample is assigned
  cluster center updates reduces the cost as each cluster only includes samples closest to it

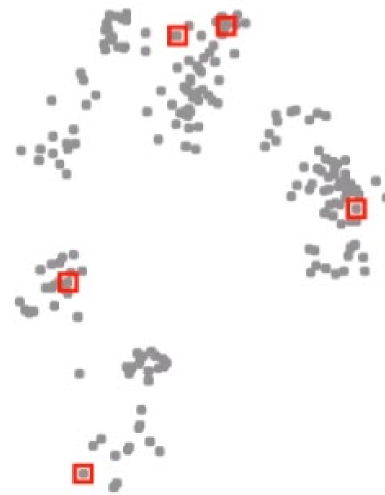


K-Means Clustering – How to initialize

- Initialization matters, resulting in different local minima
- Use different/random initialization, and perform a diagnosis based on the cost value

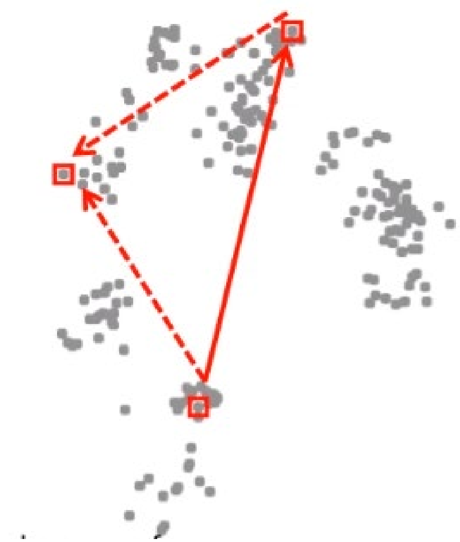


K-Means Clustering – How to initialize



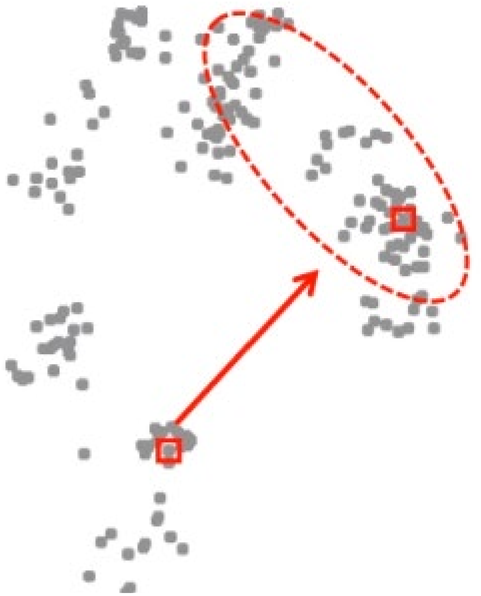
- Random initialization – to cover some data samples; but may end up choosing nearby samples

- Distance based initialization – first center is random, then find the point away from the clusters already chosen; but may end up outliers as centers



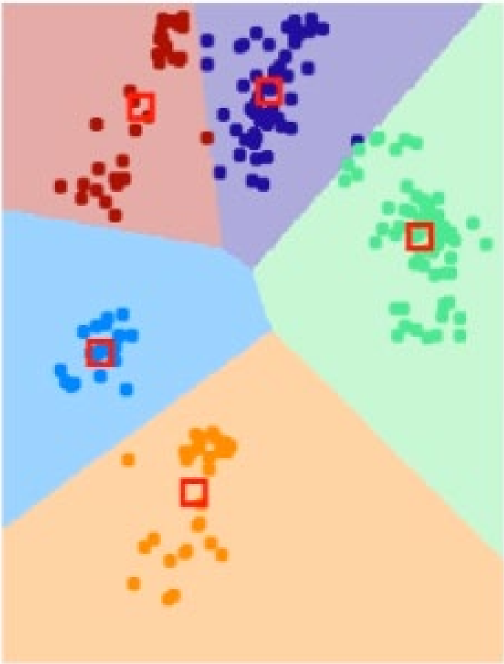
K-Means Clustering – How to initialize (e.g., K-means ++)

- K-means ++ takes advantage of the strength of each (random and distance based)
- Choose a center from far, but with randomness, i.e., select a new center according to the density of distances between samples and existing centers
- Increased chance of resulting in centers with high density of data



Using k-means to cluster new data & Voronoi tessellation

- Perform clustering by k-means
- Determine cluster labels (indices)
- Assign new samples by nearest neighbors

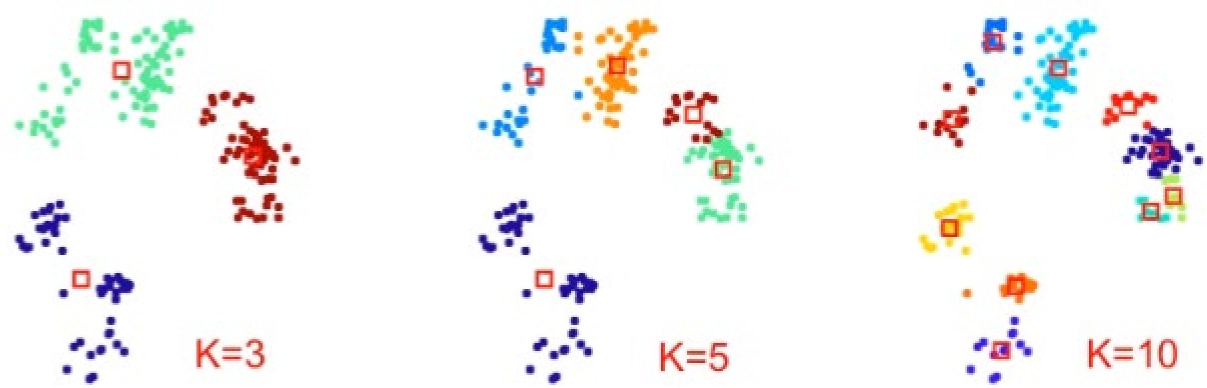


K-Means Clustering – How to choose K

With the cost function below

$$C(\underline{z}, \underline{\mu}) = \sum_i \left\| x_i - \mu_{z_i} \right\|^2$$

What is the optimal value of k?

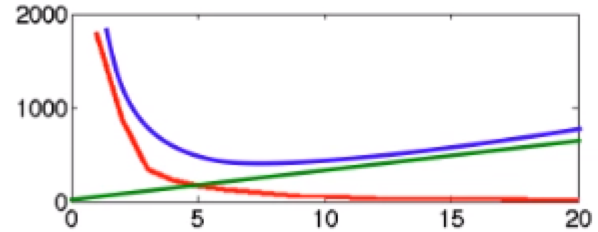


K-Means Clustering – How to choose K

- With cost function

$$C(\underline{z}, \underline{\mu}) = \sum_i \|x_i - \mu_{z_i}\|^2$$

  what is the optimal value of k?



- Cost always decreases with k!
- A model complexity issue…

- One solution is to penalize for complexity
  - Add penalty: Total = Error + Complexity
  - Now more clusters can increase cost, if they don't help "enough"
  - Ex: simplified BIC penalty

$$J(\underline{z}, \underline{\mu}) = \log \left[\frac{1}{m\,d} \sum_i \|x_i - \mu_{z_i}\|^2 \right] + k \frac{\log m}{m}$$

  - More precise version: see e.g. "X-means" (Pelleg & Moore 2000)

Highlights of K-Means

- K-Means clustering
  - Clusters described as locations ("centers") in feature space
- Procedure
  - Initialize cluster centers
  - Iterate: assign each data point to its closest cluster center
  - : move cluster centers to minimize mean squared error
- Properties
  - Coordinate descent on MSE criterion
  - Prone to local optima; initialization important
- Out-of-sample data
- Choosing the # of clusters, K
  - Model selection problem; penalize for complexity (BIC, etc.)