K-means. Clustering Goodness. Soft K-means. Gaussian Mixture Models. Kernel K-means.

Radu Ionescu, Prof. PhD.

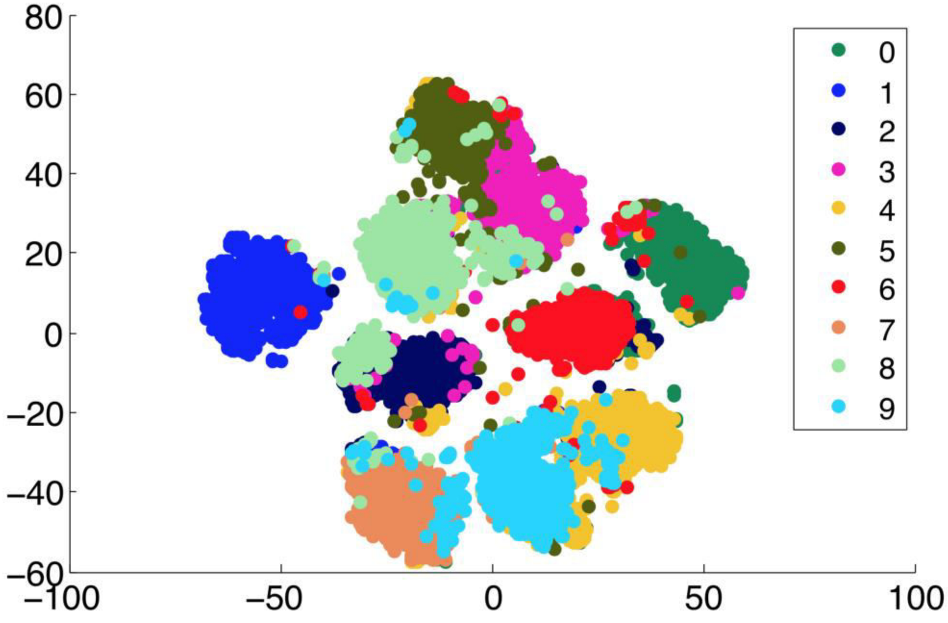
raducu.ionescu@gmail.com

Faculty of Mathematics and Computer Science

University of Bucharest

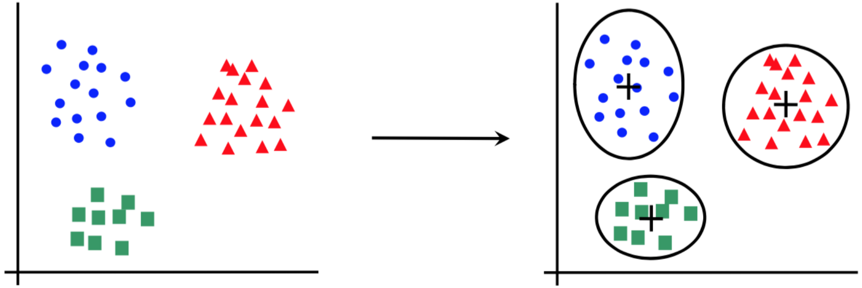
Reminder: Unsupervised Learning

- There are no labels for the training phase
- Our goal is to discover structure in data



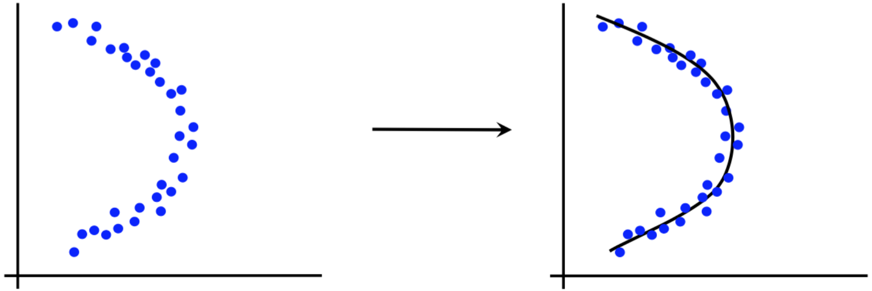
Canonical forms of unsupervised learning problems

- Clustering



- K-means
- DBSCAN
- Hierarchical Clustering
- …

- Dimensionality Reduction



- Principal Component Analysis
- t-SNE
- …

Clustering

- Clustering or Cluster Analysis is the task of grouping a set of objects such that objects in the same group (cluster) are more similar to each other than to objects in other groups
- There are several types of clustering methods:
  - Centroid-based: each cluster is represented by a prototype object (a center), e.g. k-means
  - Density-based: clusters are dense regions of space, e.g. DBSCAN
  - Distribution-based: clusters are modeled by statistical distributions, e.g. GMM
  - Graph-based: clusters are cliques in a graph, e.g. HCS
  - Hierarchical models: there is a hierarchical relationship between clusters
- Based on the relation between objects and clusters:
  - Hard-clustering (partitioning): one object can belong to a single cluster
  - Soft-clustering (fuzzy-clustering): each object has a degree of membership to each cluster

K-means

K-means

- K-means is a clustering algorithm that partitions the data points into a fixed number of clusters k
- Being a centroid-based method, each cluster is represented by a prototype point (centroid) and every point is assigned to the cluster with the nearest centroid
- Want to minimize the sum of Euclidean distances between feature vectors $\mathbf{x}_i$ and the nearest cluster centroids $\mathbf{m}_k$ (i.e. within-cluster sum of squares, variance of clusters):

$$L(X, M) = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - m_k)^2$$

- K-means uses an iterative method and it converges to a local minimum
  - ➢ Finding the global optimum is NP-hard

K-means clustering

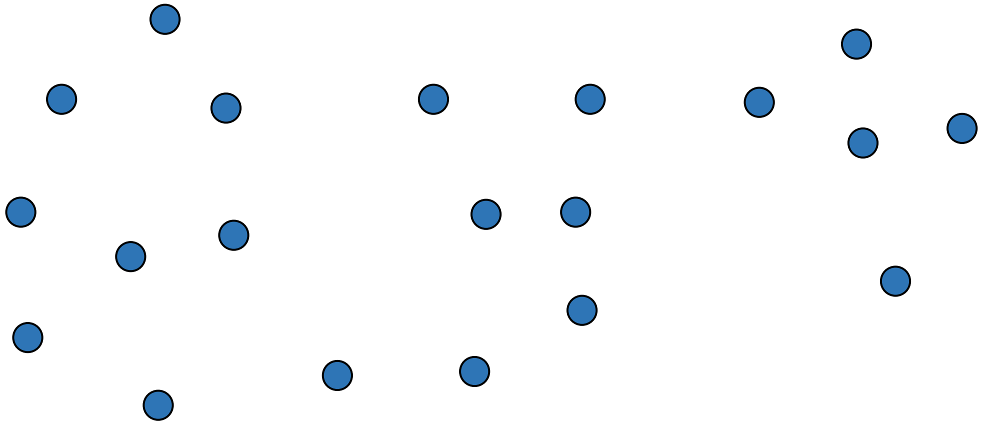
- Want to minimize the sum of Euclidean distances between feature vectors $\mathbf{x}_i$ and the nearest cluster centroids $\mathbf{m}_k$:

$$L(X, M) = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - m_k)^2$$

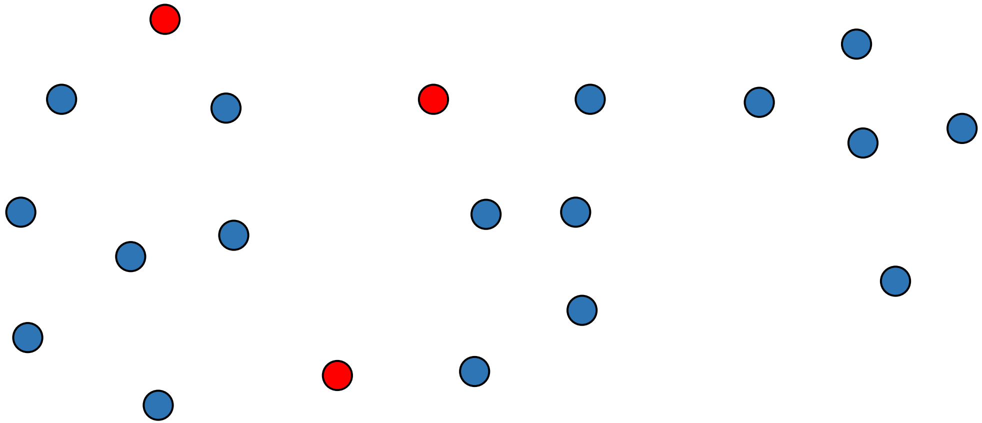
- Algorithm (Expectation-Maximization):

1. Initialize the K cluster centroids randomly
2. Iterate until clusters converge:
   - a. (E) Label each vector based on the nearest cluster centroid
   - b. (M) Recompute the centroid of each cluster = mean of all feature vectors assigned to a cluster

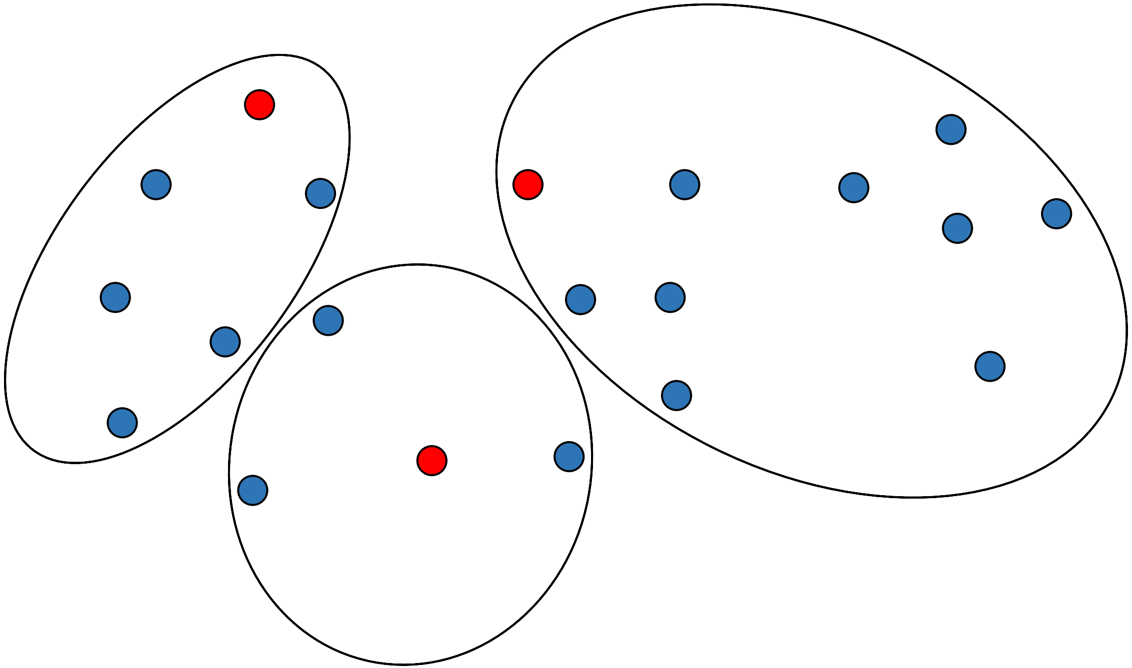
K-means clustering – 1.



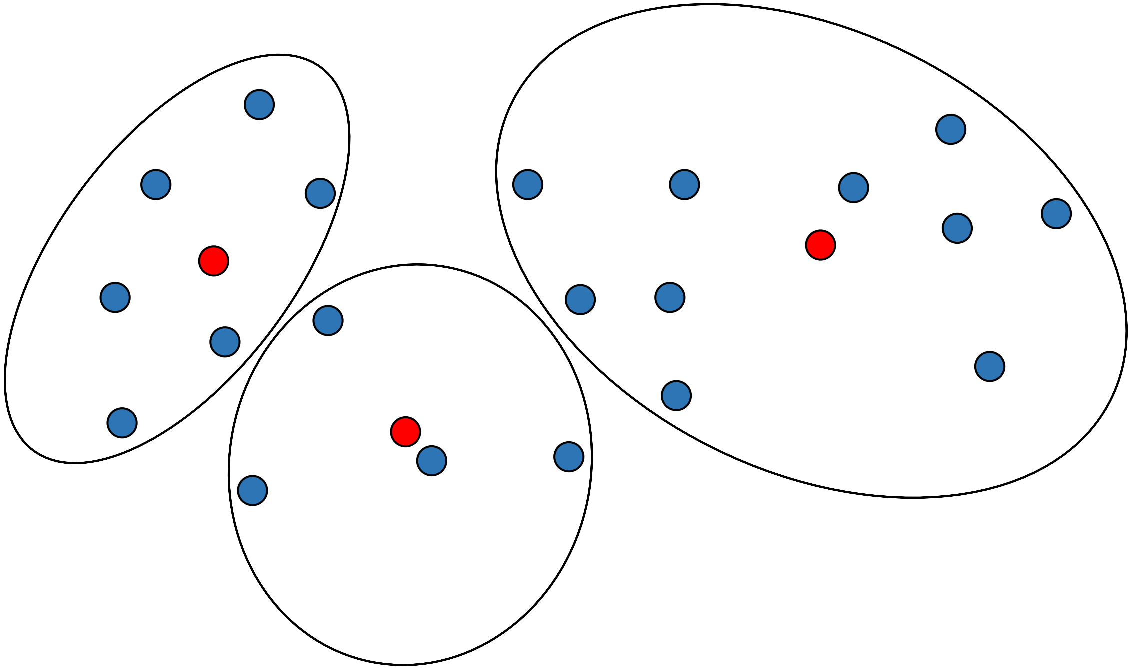
K-means clustering – 1.



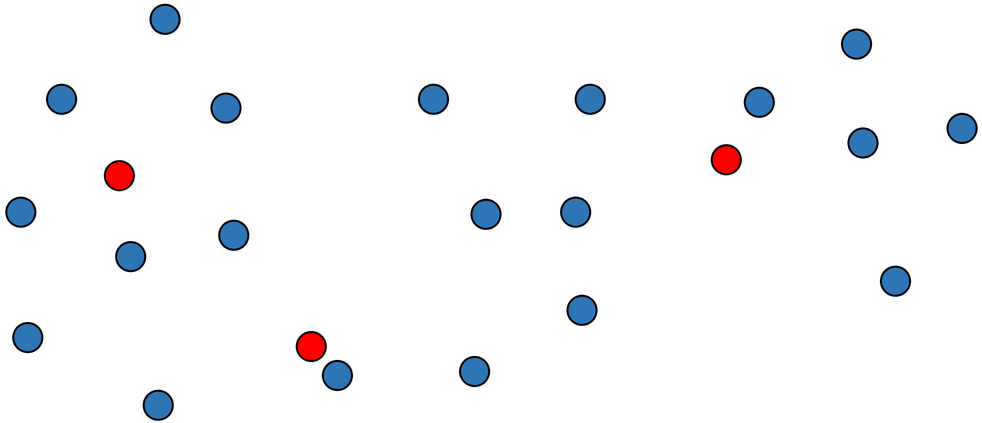
K-means clustering – 2.a.



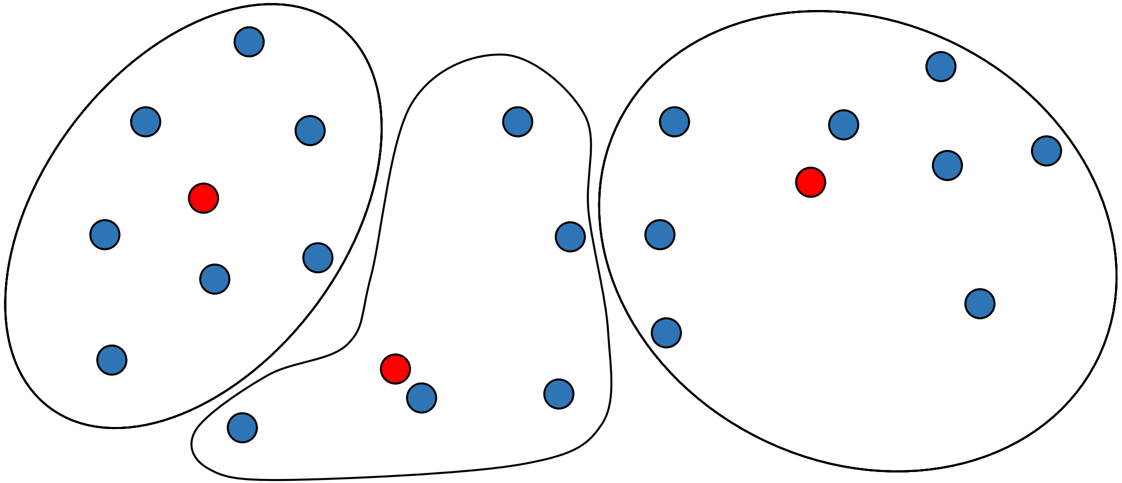
K-means clustering – 2.b.



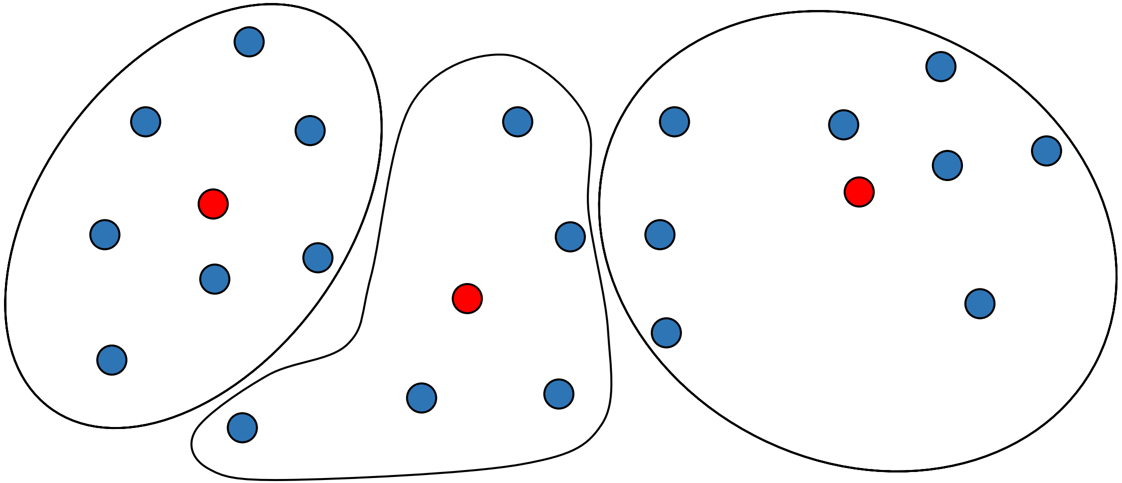
K-means clustering – 2.b



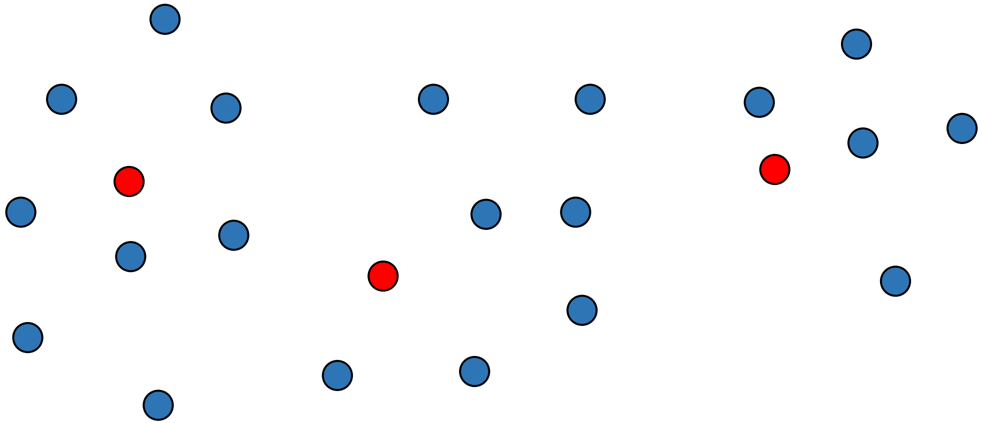
K-means clustering – 2.a



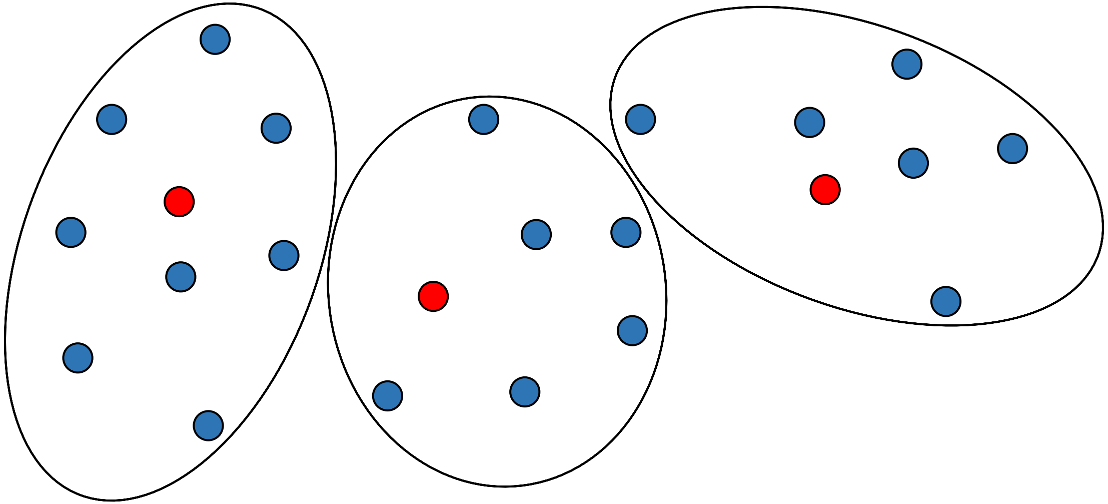
K-means clustering – 2.b.



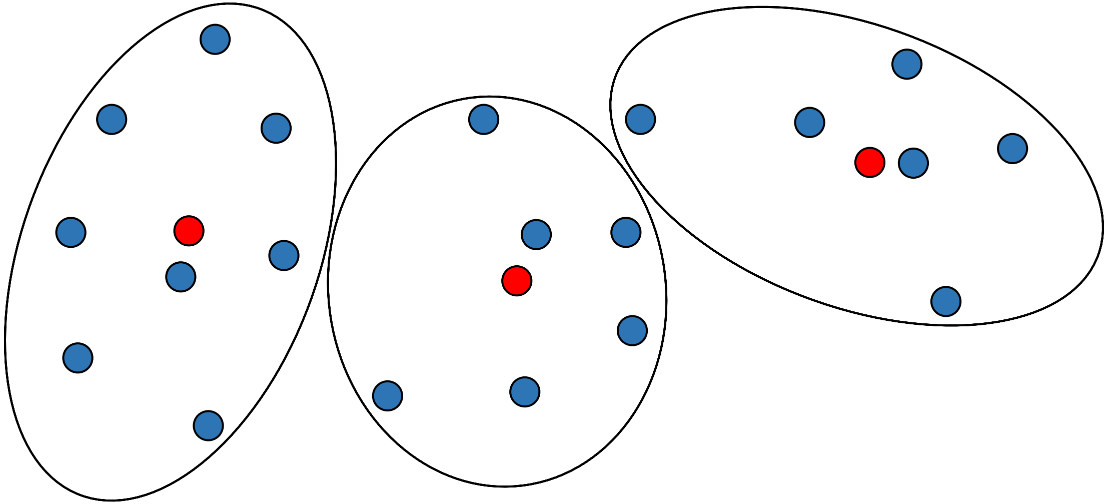
K-means clustering – 2.b.



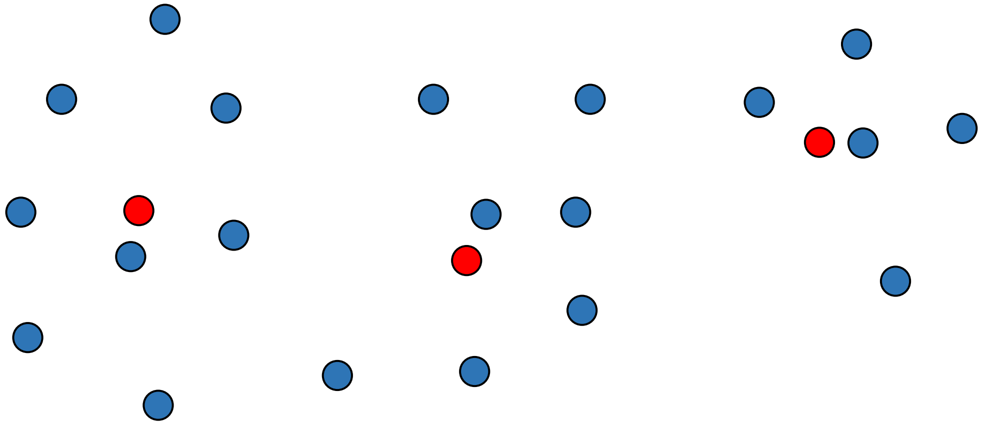
K-means clustering – 2.a.



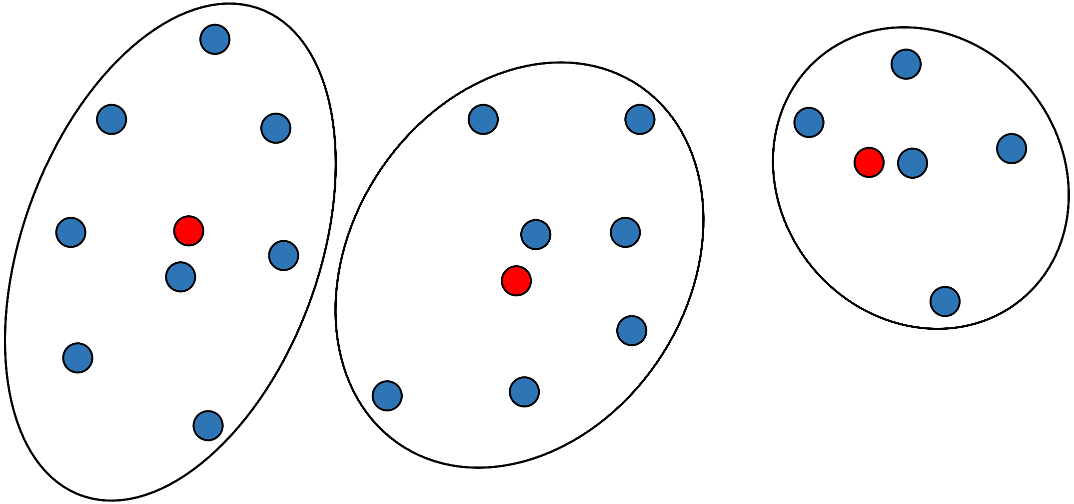
K-means clustering – 2.b.



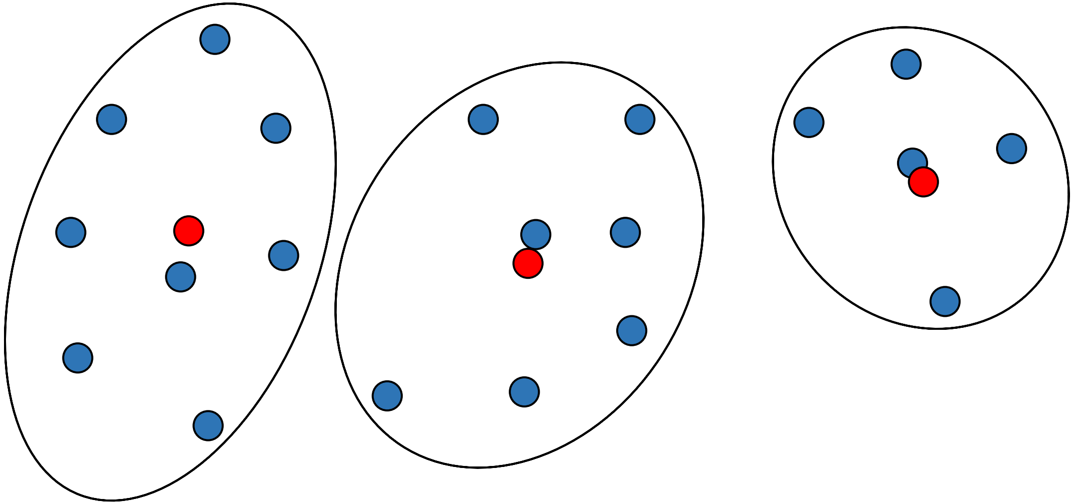
K-means clustering – 2.b.



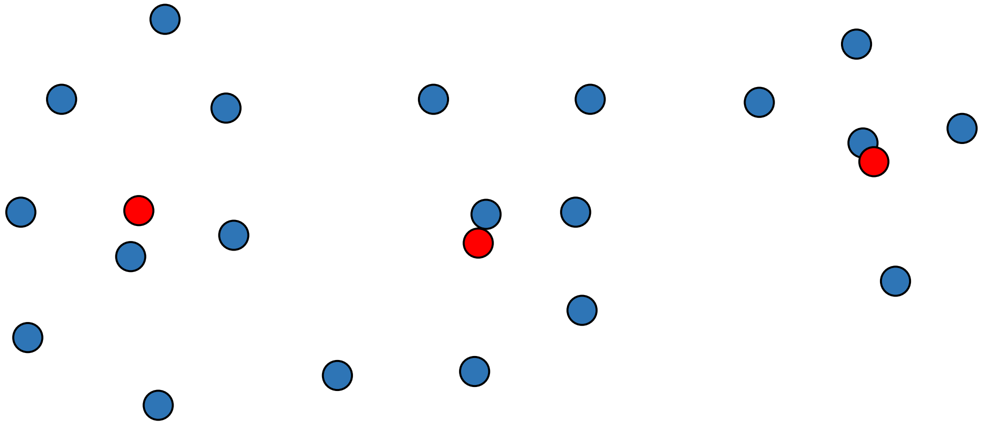
K-means clustering – 2.a.



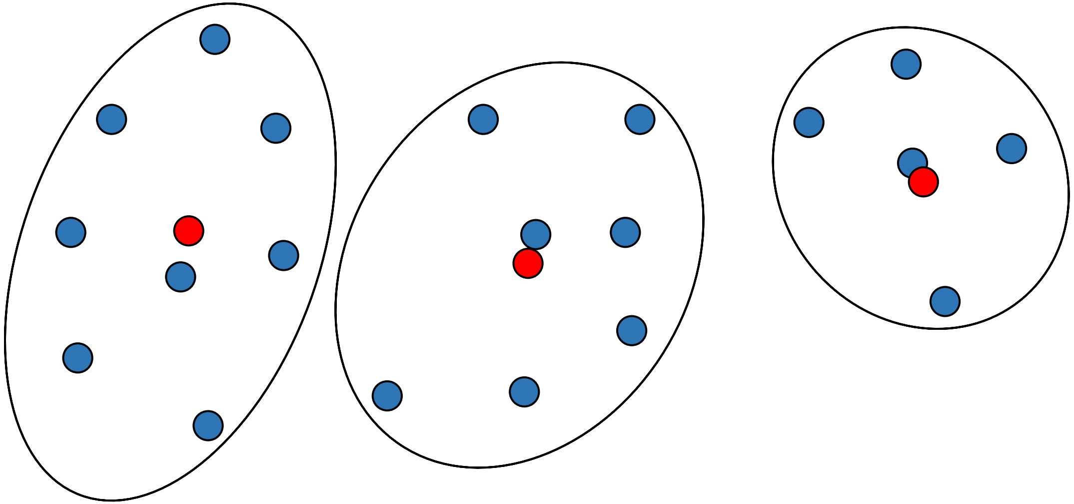
K-means clustering – 2.b.



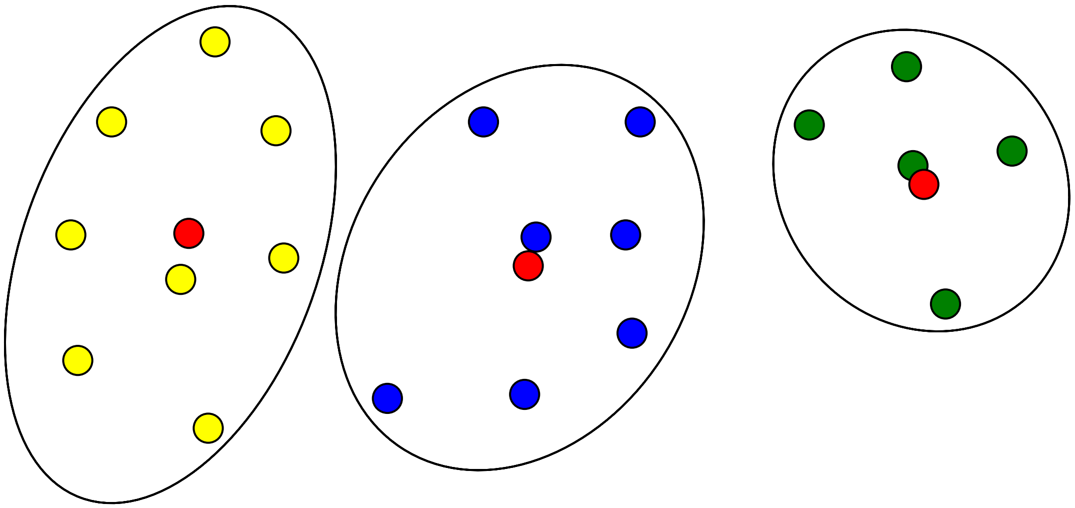
K-means clustering – 2.b.



K-means clustering – 2.a.

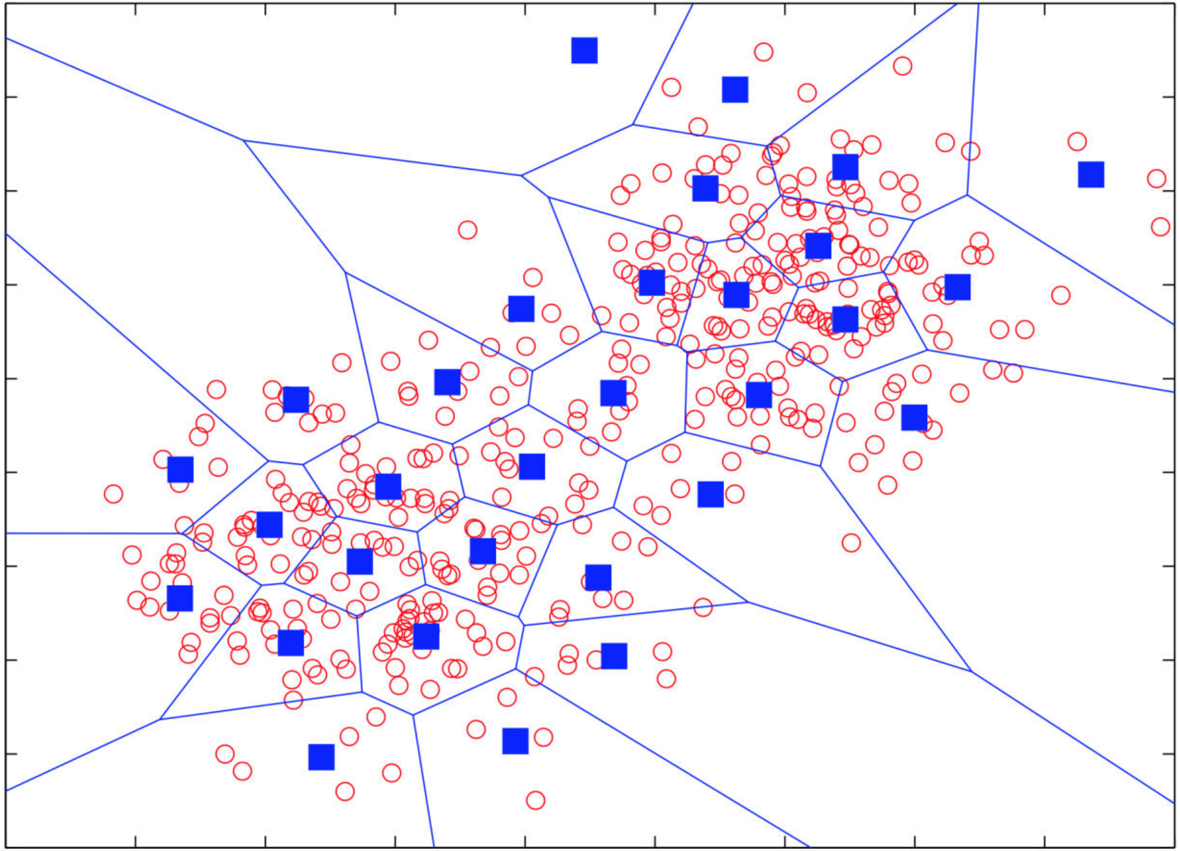


K-means clustering - Output



K-means clustering - Output

- A Voronoi tessellation!



Mathematical formulation

- For $X = \{x_1, x_2, \ldots, x_m\} \subset \mathbb{R}^n$, $K \in \mathbb{N}^+$ and M=$\{\mu_1, \mu_2, \ldots, \mu_K\} \subset \mathbb{R}^n$, we define the distortion measure:

$$L(X, M) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 = \sum_{i=1}^{m} \sum_{k=1}^{K} z_{ik} \|x_i - \mu_k\|^2$$

where:

$$z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$$

$\mu_k \in \mathbb{R}^n$ is the centroid of cluster C_k

$\sum_{k=1}^{K} z_{ik} = 1, \forall i$

Hard-clustering (partition)

- Goal: minimize L with respect to z_{ik} and μ_k

K-means algorithm: in depth

- Algorithm (Expectation-Maximization):

1. Initialization:

➢ choose random values for μ_k, for all $k \in \{1,2,\ldots,K\}$

2. Iterate until clusters converge:

- **Expectation** step:

➢ minimize L with respect to z_{ik}, keeping μ_k fixed

- **Maximization** step:

➢ minimize L with respect to μ_k, keeping z_{ik} fixed

K-means algorithm: in depth

- **Expectation** step:

  Minimize

$$L = \sum_{i=1}^{m} \sum_{k=1}^{K} z_{ik} \|x_i - \mu_k\|^2$$

w.r.t. z_{ik}, keeping μ_k fixed

- There must be only one $z_{ik} = 1, \forall i,$ (i.e. x_i will be assigned to a single cluster):
  - To minimize L, we must set:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| \\ 0, otherwise \end{cases}$$

  - In order words, we assign x_i to the cluster with the nearest centroid

K-means algorithm: in depth

- **Maximization** step:

  Minimize

$$L = \sum_{i=1}^{m} \sum_{k=1}^{K} z_{ik} \|x_i - \mu_k\|^2$$

w.r.t. μ_k, keeping z_{ik} fixed

- L is a quadratic function of $\mu_k \Longrightarrow$ minimize by setting $\frac{\partial L}{\partial \mu_k} = 0$

$$\frac{\partial L}{\partial \mu_k} = 2 \sum_i z_{ik}(x_i - \mu_k) = 2 \sum_i z_{ik} x_i - 2 \sum_i z_{ik} \mu_k = 0 \Longrightarrow$$

$$\mu_k = \frac{\sum_i z_{ik} x_i}{\sum_i z_{ik}} = \frac{1}{|C_k|} \sum_{x \in C_k} x$$

➢ In order words, we set μ_k to the mean of all points in C_k

Parameters and Evaluation

How to choose k?

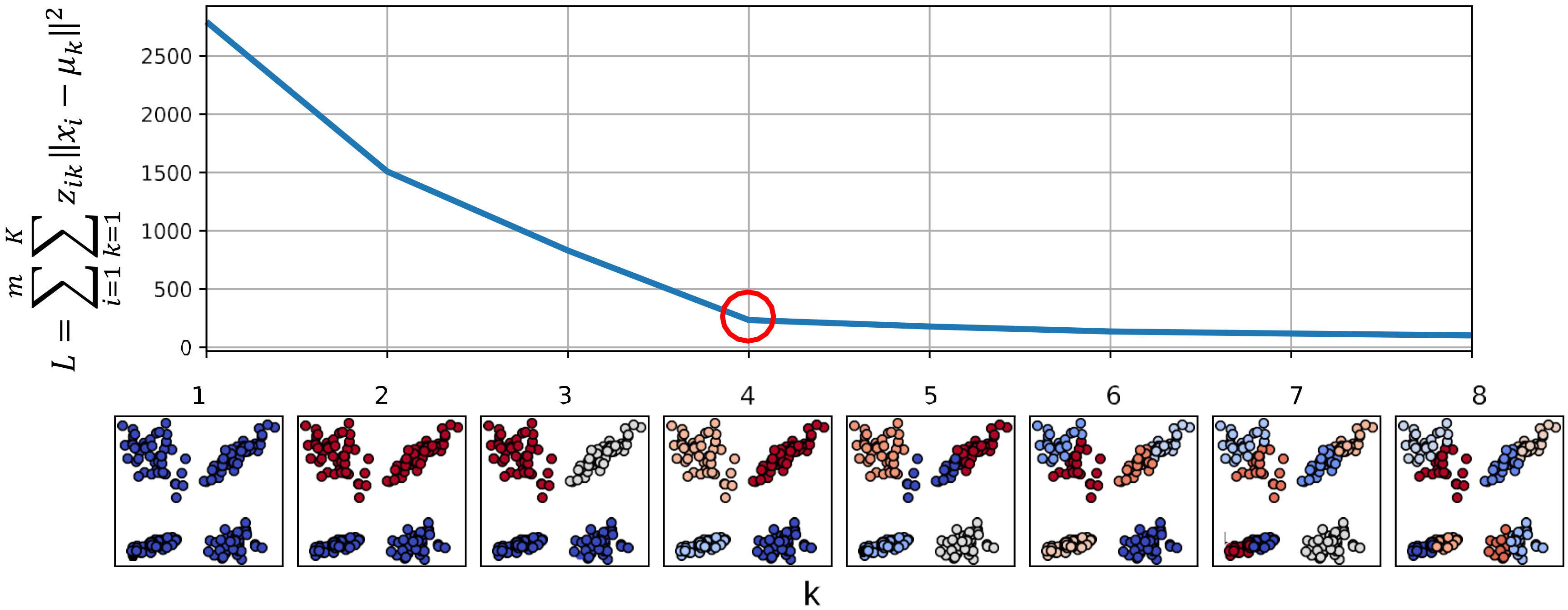
- The number of clusters k is a hyperparameter. How do we find a good k?

1. Elbow method:
   - Start with a small k value and increase it until adding another cluster does not result in a much lower distortion value
   - In other words, the new cluster does not explain so much the variance in data

2. Silhouette Coefficient:
   - A measure of how tight each cluster is and now fart apart clusters are from each other
   - Choose a value k that results in clustering with a large silhouette coefficient

The Elbow Method

- Choose k such that adding another cluster will not explain the variance in data by much (i.e. does not give a much lower distortion value)



The Silhouette Coefficient

- Measures the tightness of clusters and separation between clusters:

$$S = \frac{1}{m}\sum_{i=1}^{m} s(x_i)$$

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

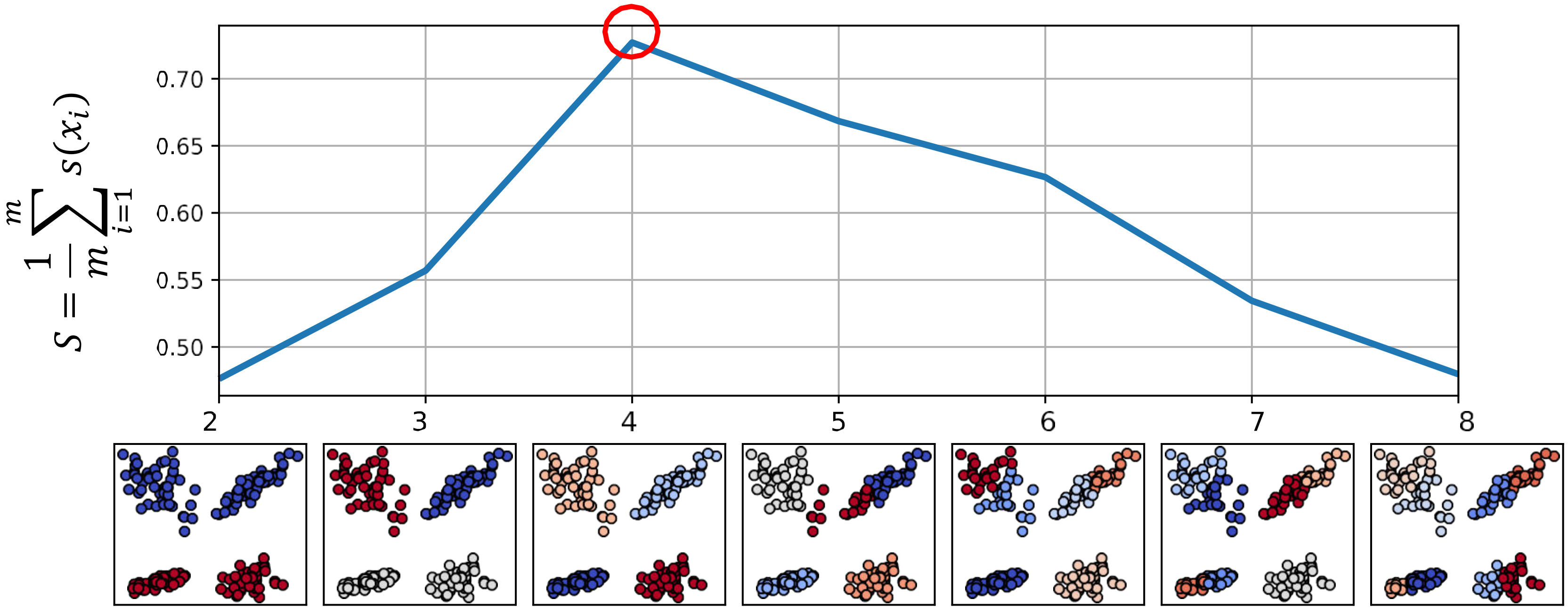
where:

- ➢ $a(x_i)$ is the average distance between x_i and all other points in the same cluster
- ➢ $\mathrm{b}(x_i)$ is the lowest distance to all points in a different cluster

(i.e., the average distance to the nearest neighboring cluster)

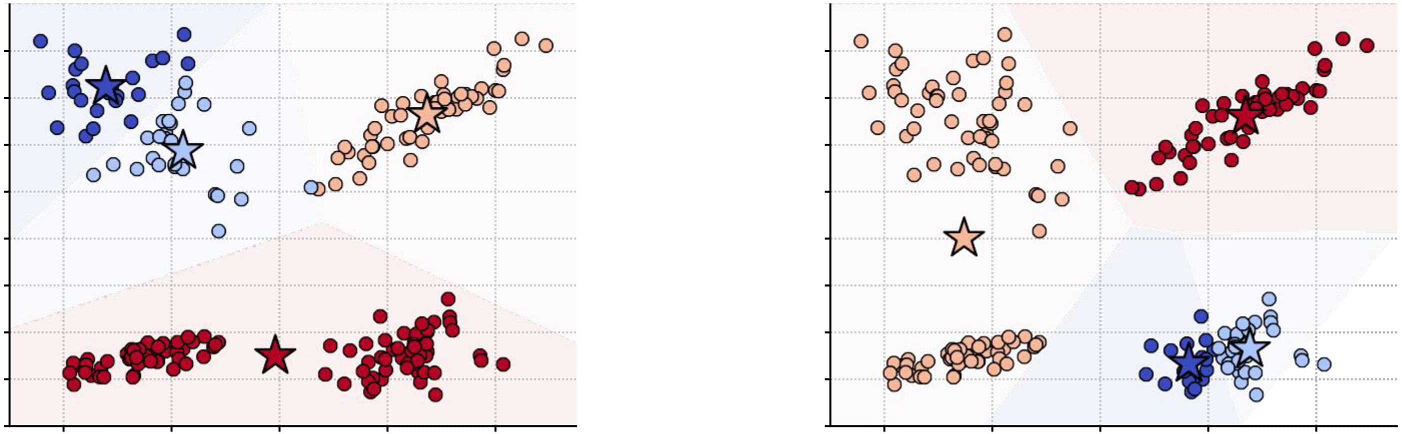
The Silhouette Coefficient

- Choose k that gives the highest mean silhouette



Local Minima

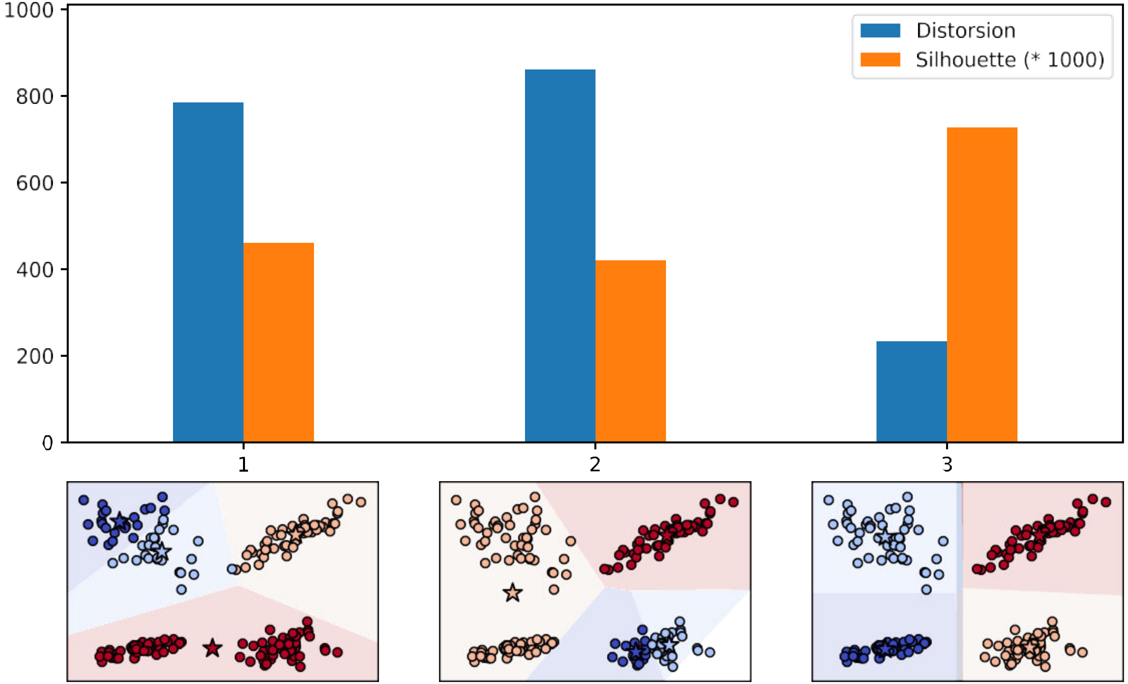
- K-means converges to local minima
- Both of the following states are stable (more iterations will not change the clustering)



- Possible solutions:
  - ➢ Run the algorithm multiple times and choose the result with lowest distortion (or highest silhouette)
  - ➢ Use a better initialization method

Local Minima

- Run the algorithm multiple times and choose the result with lowest distortion (or highest silhouette)



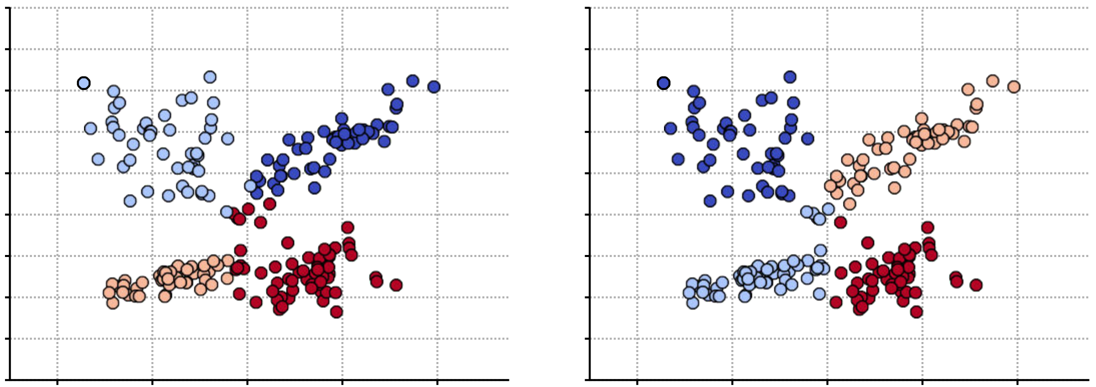
K-means++ initializaiton

- Use a better initialization method:
  - “k-means++: the advantages of careful seeding” [D. Arthur & S. Vassilvitskii, 2007]
  - Idea: choose initial centers such that they are spread out over entire data
  - Algorithm:

1. Randomly choose first center from the data points
2. Repeat until all k centers have been chosen:
   2.a. compute $D(x_i)$, the distance from x_i to the nearest chosen center
   2.b. randomly choose a new center with probability $\mathrm{P}(x_i) \sim D(x_i)^2$
3. Run the standard k-means algorithm

Comparing results

- How similar are these two clustering results?



The actual label assigned to each cluster is not important, only the grouping of points matters

- **Rand Index** measures how often two clustering results agree in terms of grouping:

$$R = \frac{a + b}{n(n-1)/2}$$

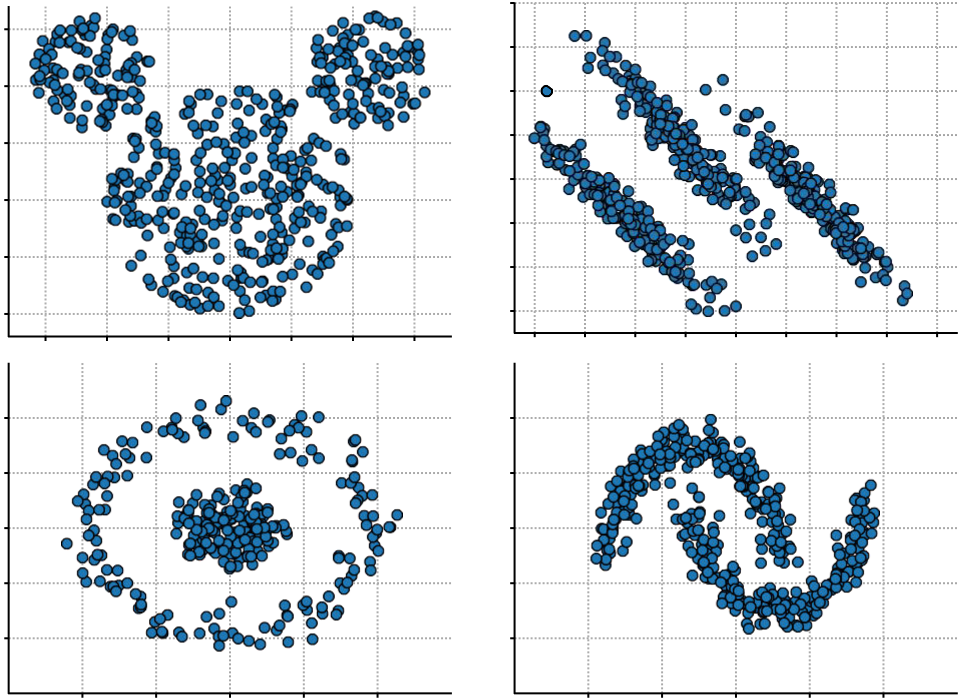
Adjusted Rand Index is another measure that takes into account that results might agree by chance

where:

- a is the number of pairs of points that in the same cluster in both assignments
- b is the number of pairs of points that are in different clusters in the two assignments

Limitations of K-means

- How will k-means handle these data sets?



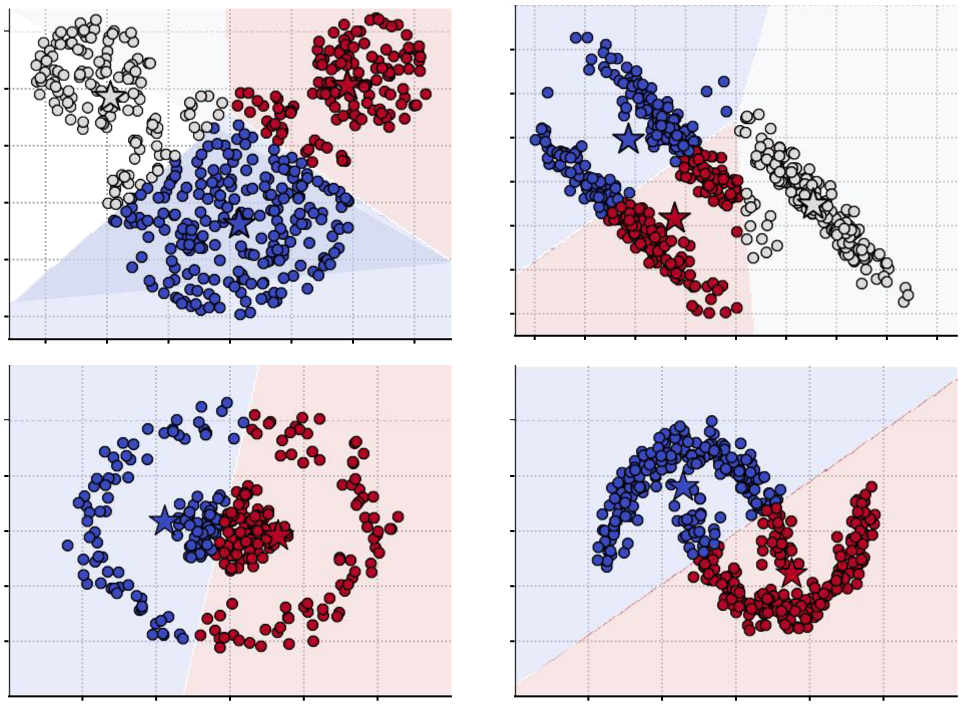
Limitations of K-means

- How will k-means handle these data sets?
- Not so good…

➢ K-means only produces convex clusters

➢ It does not handle non-spherical clusters too well

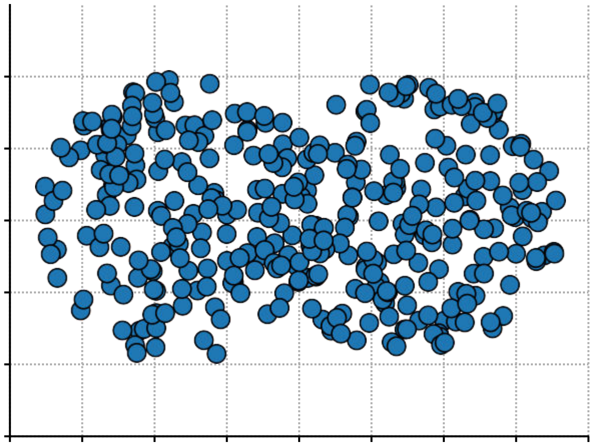
➢ It tends to produce clusters of equal size



K-means Variations

Soft K-means

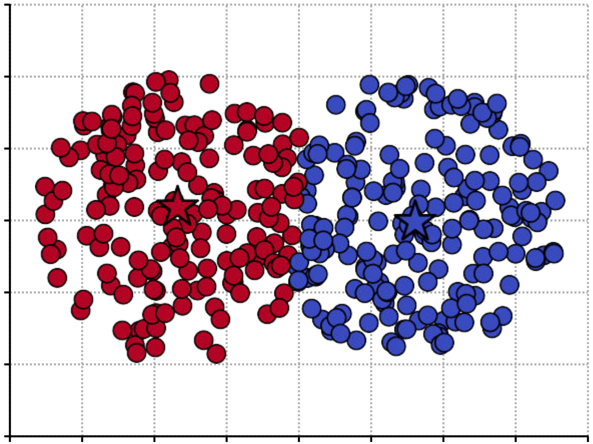
- Recall K-means Expectation step: $z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| \\ 0, otherwise \end{cases}$
- It will produce a partitioning (hard-clustering), i.e. a point belongs to one and only one cluster
- Sometimes, in practice, clusters might have overlapping regions, and there is no clear boundary between clusters



Soft K-means

- Recall K-means Expectation step: $z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| \\ 0, otherwise \end{cases}$
- It will produce a partitioning (hard-clustering), i.e. a point belongs to one and only one cluster
- Sometimes, in practice, clusters might have overlapping regions, and there is no clear boundary between clusters

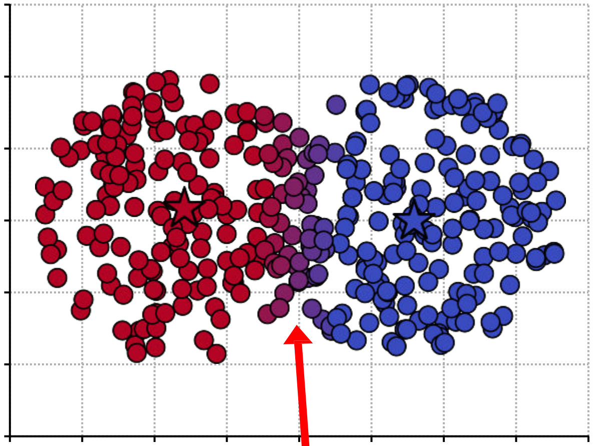
- With hard-clustering, the assignment in such regions will likely be caused by chance from random initialization



Soft K-means

- Recall K-means **Expectation** step: $z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| \\ 0, otherwise \end{cases}$
- It will produce a partitioning (hard-clustering), i.e. a point belongs to one and only one cluster
- Sometimes, in practice, clusters might have overlapping regions, and there is no clear boundary between clusters

- With hard-clustering, the assignment in such regions will likely be caused by chance from random initialization
- **Soft K-means** redefines the **Expectation** step such that $z_{ik} \in \mathbb{R}$ is the degree of membership of x_i to cluster C_k



Points in this area have ~0.5 membership in both clusters

$$z_{ik^*} = \frac{e^{-\beta\|x_i - \mu_{k^*}\|}}{\sum_k e^{-\beta\|x_i - \mu_k\|}}$$

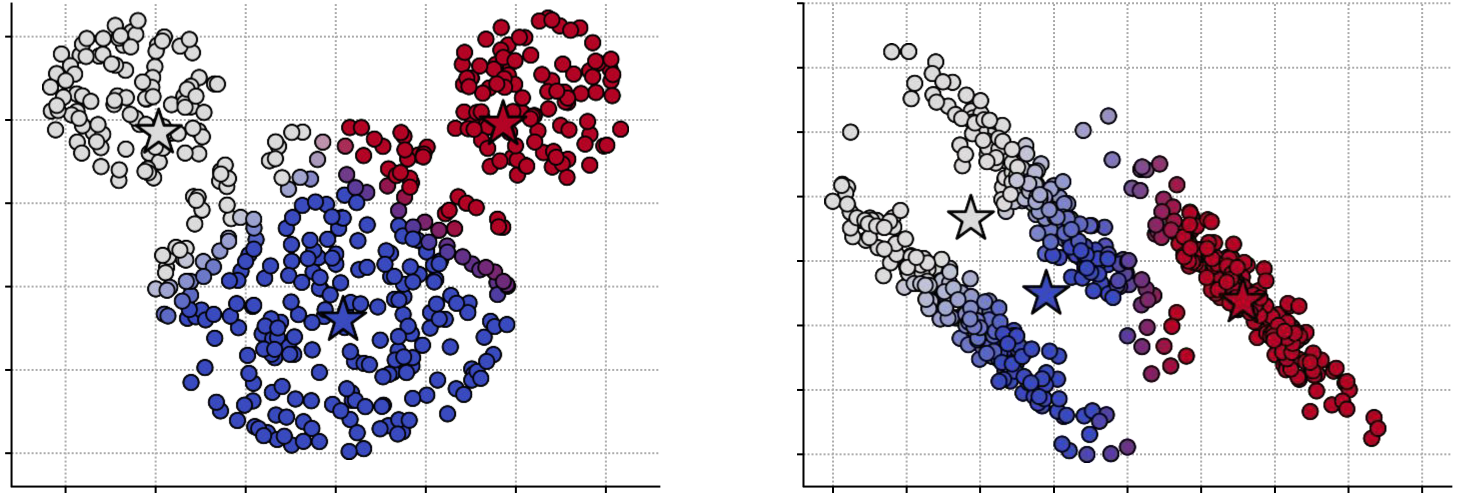
Expectation

$$\mu_k = \frac{\sum_i z_{ik} x_i}{\sum_i z_{ik}}$$

Maximization

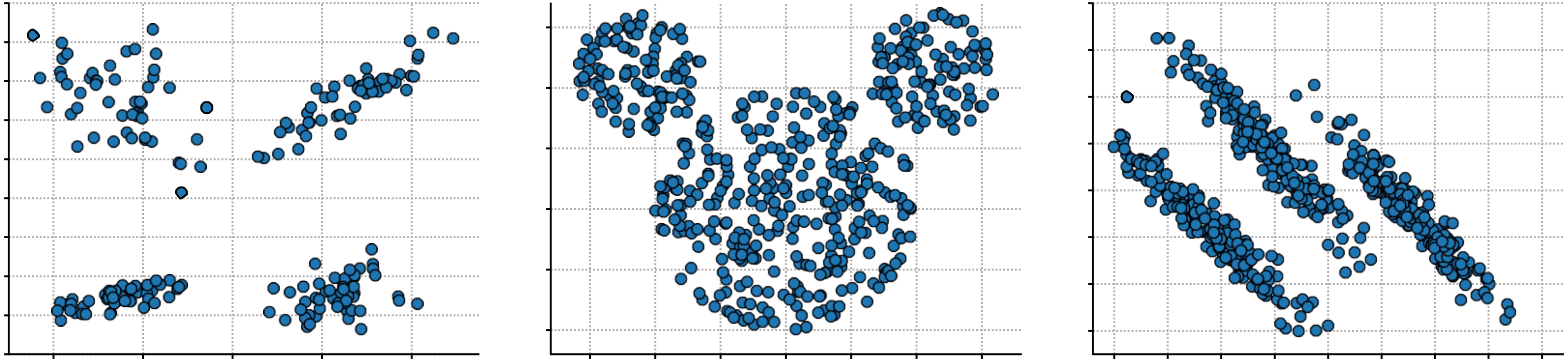
Soft K-means

- Soft K-means does not solve issues of unequally-sized and non-spherical clusters



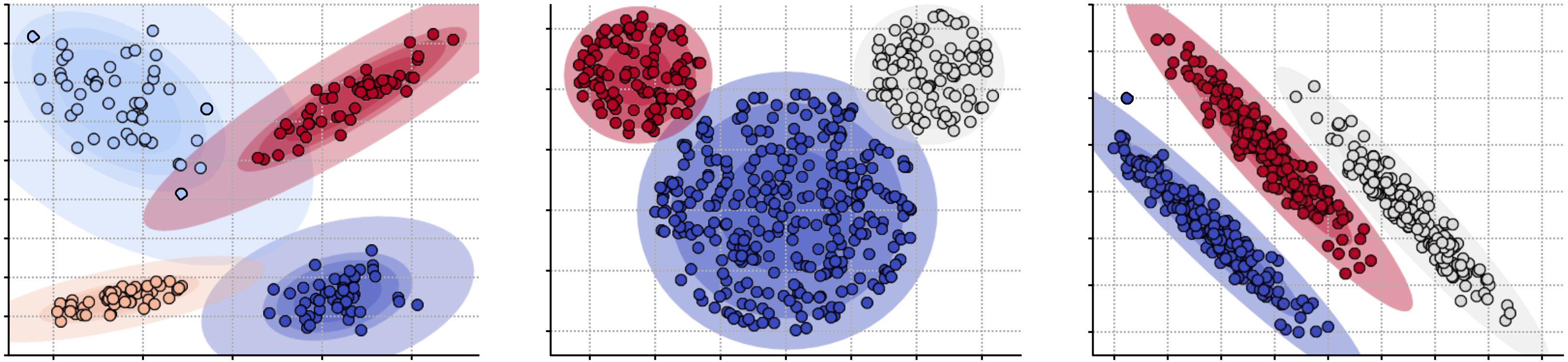
Gaussian Mixture Models

- GMMs are probabilistic models that assume that data points are generated by a mixture of normal distributions, i.e. Gaussians
- An EM algorithm can be used to fit the Gaussians by maximizing the likelihood of data



Gaussian Mixture Models

- GMMs are probabilistic models that assume that data points are generated by a mixture of normal distributions, i.e. Gaussians
- An EM algorithm can be used to fit the Gaussians by maximizing the likelihood of data



- GMMs can be viewed as an extension of soft K-means in which each cluster has both a mean and a covariance matrix (that gives the non-spherical shape)

Kernel K-means

Removing centroids from the equation

- **Expectation** step means setting: $z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| \\ 0, otherwise \end{cases}$

$$\|x_i - \mu_k\| = \sqrt{\langle x_i - \mu_k, x_i - \mu_k \rangle} = \sqrt{\langle x_i, x_i \rangle - 2\langle x_i, \mu_k \rangle + \langle \mu_k, \mu_k \rangle}$$

Norm of a vector is the square root of the dot product with itself

Dot product is distributive

Removing centroids from the equation

- **Expectation** step means setting: $z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| \\ 0, otherwise \end{cases}$

$$\|x_i - \mu_k\| = \sqrt{\langle x_i - \mu_k, x_i - \mu_k \rangle} = \sqrt{\langle x_i, x_i \rangle - 2\langle x_i, \mu_k \rangle + \langle \mu_k, \mu_k \rangle}$$

$$= \sqrt{\|x_i\|^2 - 2\left\langle x_i, \frac{1}{|C_k|} \sum_{s \in C_k} s \right\rangle + \left\langle \frac{1}{|C_k|} \sum_{s \in C_k} s\ , \frac{1}{|C_k|} \sum_{t \in C_k} t \right\rangle}$$

$$= \sqrt{\|x_i\|^2 - 2\,\frac{1}{|C_k|} \sum_{s \in C_k} \langle x_i, s \rangle + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} \langle s, t \rangle}$$

Dot product is distributive

$$\underset{k}{\operatorname{argmin}} \|x_i - \mu_k\| = \underset{k}{\operatorname{argmin}} \left(-2 \frac{1}{|C_k|} \sum_{s \in C_k} \langle x_i, s \rangle + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} \langle s, t \rangle \right)$$

$\|x_i\|^2$ and $\sqrt{}$ do not affect argmin

- We can do k-means clustering without computing the centroids
- The expression depends only on the dot product of pairs of training samples!
  - ➢ We can use a kernel function instead of the dot product!

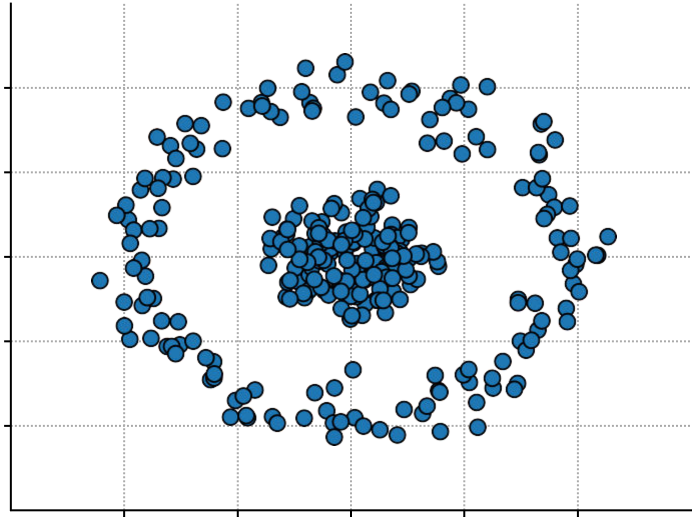
Kernel K-means

- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2 \frac{1}{|C_k|} \sum_{s \in C_k} K(x_i, s) + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} K(s, t) \right) \\ 0, otherwise \end{cases}$$

where:

- ➢ $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- ➢ $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



Kernel K-means

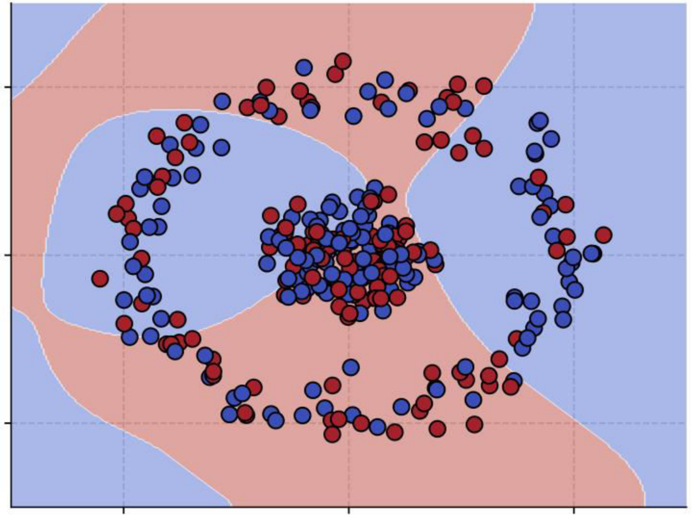
- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2 \frac{1}{|C_k|} \sum_{s \in C_k} K(x_i, s) + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function

- For example:
- $k = 2$
- $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$



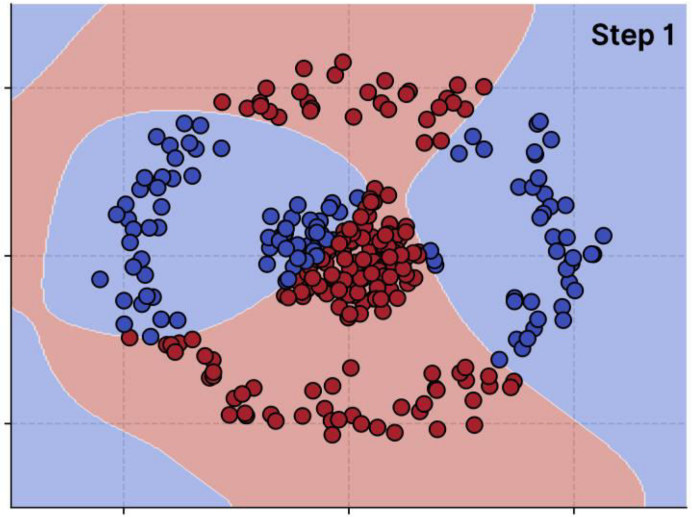
Kernel K-means

- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2\frac{1}{|C_k|}\sum_{s\in C_k} K(x_i,s) + \frac{1}{|C_k|^2}\sum_{s\in C_k}\sum_{t\in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



- For example:
- $k = 2$
- $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$

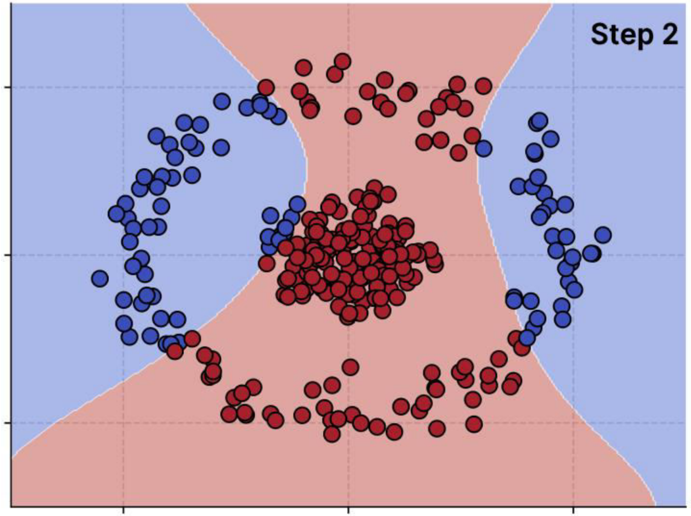
Kernel K-means

- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2\frac{1}{|C_k|} \sum_{s \in C_k} K(x_i, s) + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



- For example:
- $k = 2$
- $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$

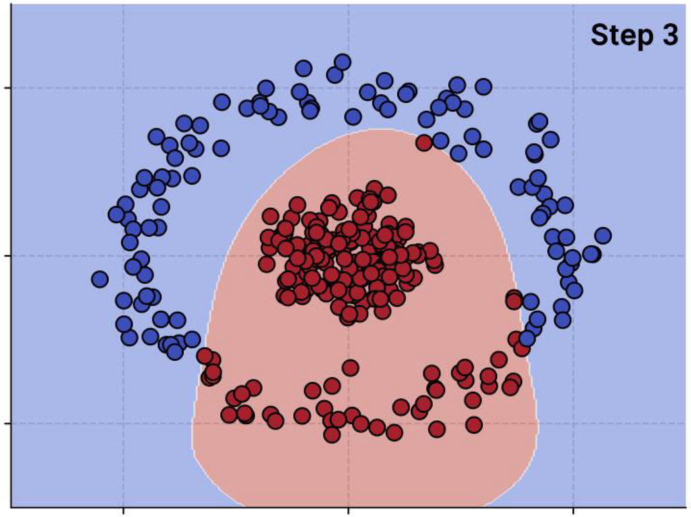
Kernel K-means

- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2 \frac{1}{|C_k|} \sum_{s \in C_k} K(x_i, s) + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



- For example:
- $k = 2$
- $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$

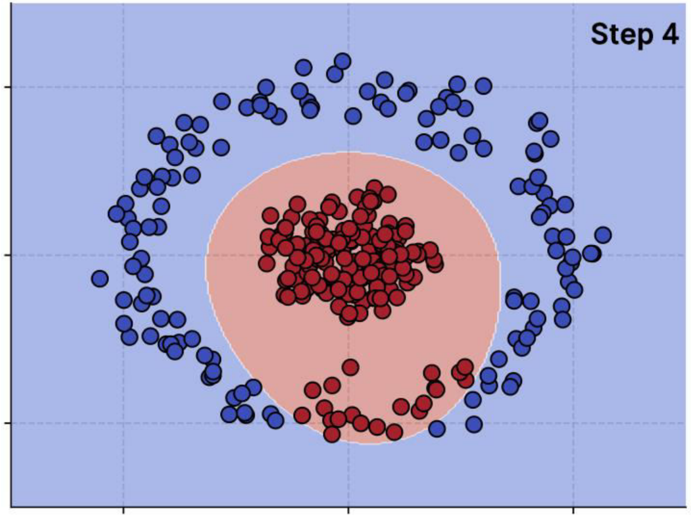
Kernel K-means

- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2 \frac{1}{|C_k|} \sum_{s \in C_k} K(x_i, s) + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



- For example:
- $k = 2$
- $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$

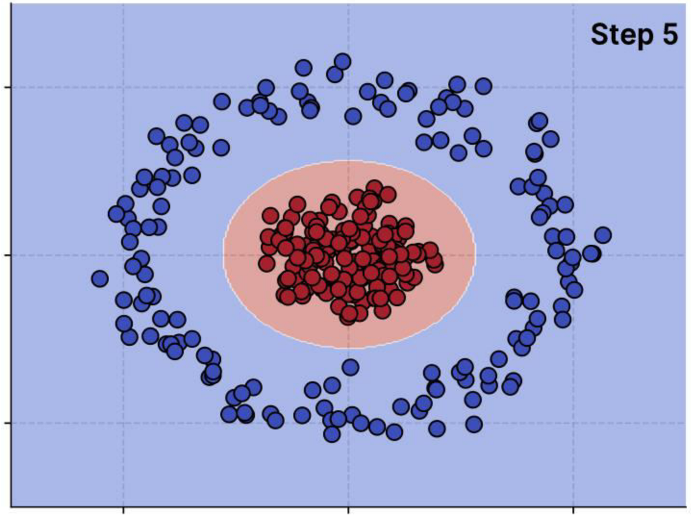
Kernel K-means

- Assign each point to a random cluster
- Repeat until no change occurs:

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2 \frac{1}{|C_k|} \sum_{s \in C_k} K(x_i, s) + \frac{1}{|C_k|^2} \sum_{s \in C_k} \sum_{t \in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- ➢ $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- ➢ $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



- For example:
- ➢ $k = 2$
- ➢ $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$

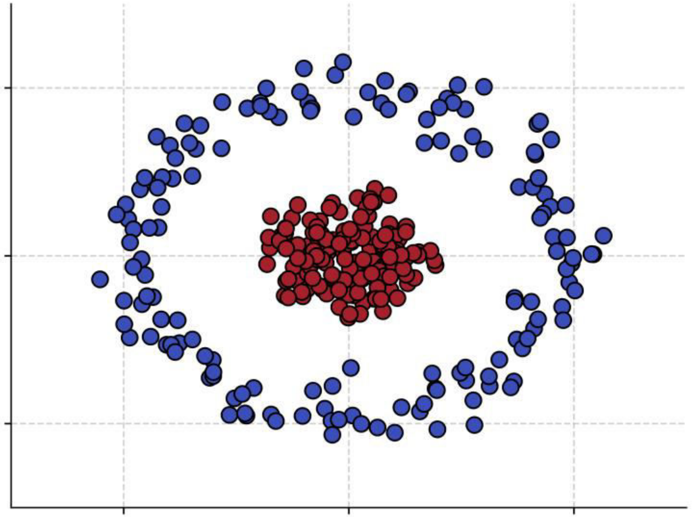
Kernel K-means

- Assign each point to a random cluster
- $\boxed{\text{Repeat until no change occurs:}}$

$$z_{ik^*} = \begin{cases} 1, if\ k^* = \underset{k}{\operatorname{argmin}} \left(-2\frac{1}{|C_k|}\sum_{s\in C_k} K(x_i, s) + \frac{1}{|C_k|^2}\sum_{s\in C_k}\sum_{t\in C_k} K(s,t) \right) \\ 0, otherwise \end{cases}$$

where:

- $z_{ik} = \begin{cases} 1, if\ x_i \in C_k \\ 0, otherwise \end{cases}$
- $K: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is a kernel function



- For example:
- $k = 2$
- $K(s,t) = e^{-\frac{\|s-t\|^2}{2\sigma^2}}$

K-means (Python)

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, adjusted_rand_score
from sklearn.mixture import GaussianMixture

km = KMeans(n_clusters = 4)
# k = 4, by default uses k-means++ initialization and does 10 runs

km.fit(X) # run the algorithm, compute the cluster centers

y = km.predict(X)
# cluster assignment for the points it was fitted on

km.cluster_centers_
km.inertia_ # final distortion value

silhouette_score(X, y) # mean silhouette score over all samples
```

Summary

- **K-means** is a clustering algorithm that partitions the data points into a fixed number of clusters k
- Each cluster is represented by a centroid and points are assigned to the cluster with the closest centroid
- The **EM algorithm** is used to optimize the objective function, but it can get stuck in local optima
- Number of clusters k is a hyperparameter that can be tuned using:
  - **Elbow** method
  - **Silhouette** coefficient
- K-means can only obtain convex spherical clusters and tends to produce equally-sized clusters
- **Soft K-means**, **Gaussian Mixture Models** and **Kernel K-means** are extensions that deal with some of the limitations of k-means