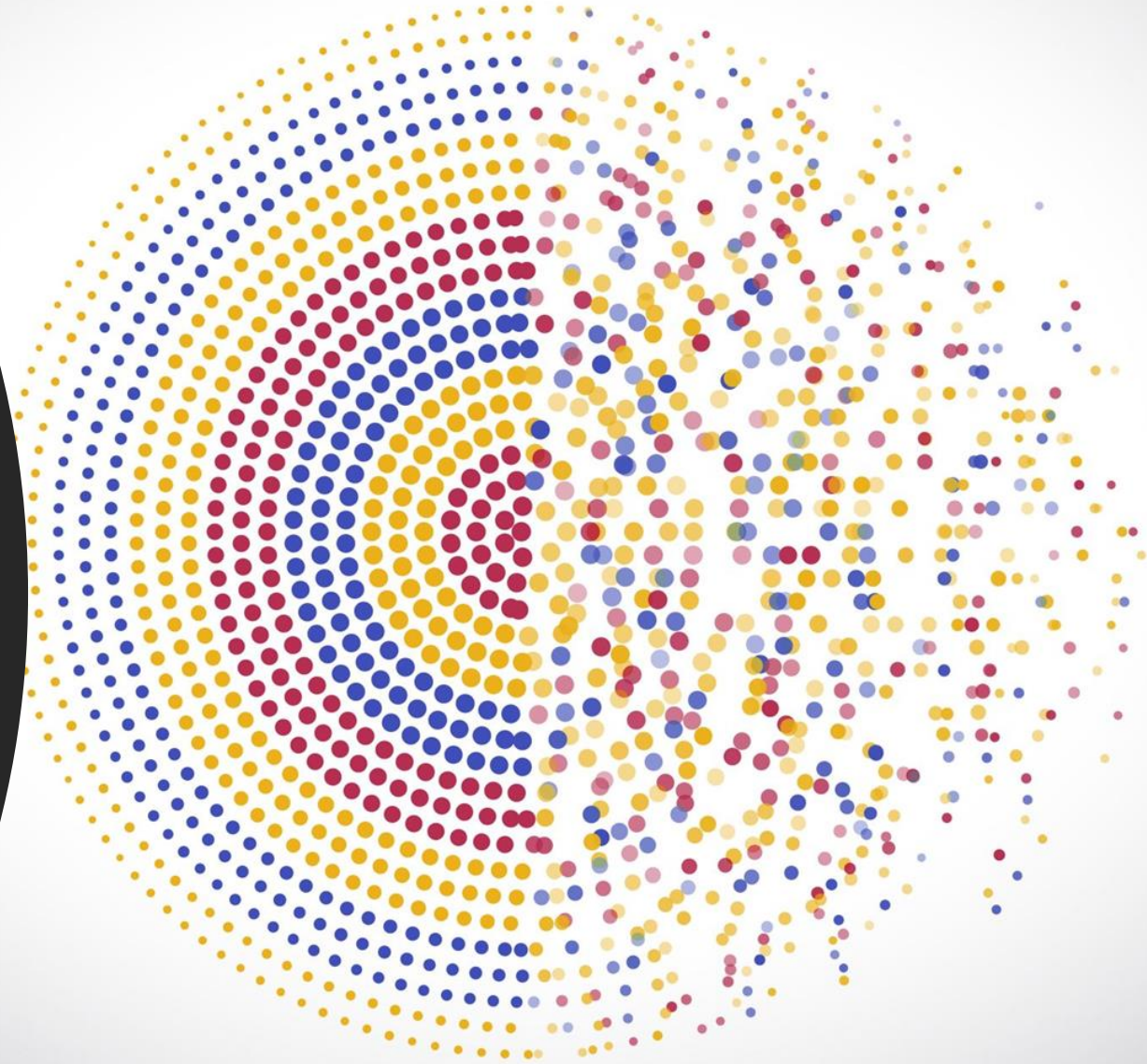
Unsupervised Learning

Dr. Mohamed AlHajri



Content

- Unsupervised Learning
- K-means clustering
- K-medoids clustering
- Ball K-means clustering (Further reading)

Unsupervised Learning

- Unsupervised learning is a branch of machine learning, including problems such as clustering, that seek to identify structure among unlabeled data. This is different than supervised learning, where labelled data is used. Unsupervised learning has many different applications such as news clustering (google news), customer segmentation, anomaly detection, image quantization, etc.

Clustering

- Clustering is the process of grouping similar data together. Therefore, a **measure of similarity** is needed to group datapoints together. To do that:
- We will assume we have k different clusters $(C_1, \cdots, C_k)$ and for each cluster there is a representative $(z_1, \cdots, z_k)$. Then the datapoints will be compared to these representative points to determine which cluster they belong to.

$$\min_{j=1,\cdots,k} dist(x_i, z_j)$$

where x_i is the ith datapoint, and z_j is the representative point of cluster j

- There are many different distacnes that can be used here but first we will concentrate on the Euclidean squared distance and then generalize from there.

$$dist(x_i, z_j) = \left|\left|x_i - z_j\right|\right|_2^2$$

- The cost function over all datapoints will be

$$cost(C_1, \cdots, C_k, z_1, \cdots, z_k) = \sum_{j=1}^{k} \sum_{i \in C_j} \left|\left|x_i - z_j\right|\right|_2^2$$

Clustering

- There are two types of clustering
- **Hard Clustering:** In hard clustering, each data point belongs exclusively to a single cluster. This approach classifies each data point definitively, assigning it to only one cluster based on its characteristics.

$$C_i \cap C_j = \emptyset \ (i \neq j)$$

- **Soft Clustering:** In contrast, soft clustering allows each data point to belong to multiple clusters with varying degrees of membership. Instead of a strict assignment, soft clustering provides a probability or likelihood that a data point belongs to each cluster. This approach is particularly useful when data points share features across different clusters, making it difficult to assign them to just one.

k-means clustering

- **K-Means Clustering** is one of the most popular and straightforward clustering algorithms used to partition a dataset into distinct groups, or "clusters." The goal of k-means is to divide the data points into k clusters, where each point belongs to the cluster with the nearest mean, serving as the cluster's centroid.

k-means clustering

1. Specify the number of clusters k.
2. Randomly select the representative $(z_1, \cdots, z_k)$.
3. Iterate until there is no change in the cost

3.1 Given $(z_1, \cdots, z_k)$, go over all x_i and for each x_i assign to it the closest z_j, more precisely:

$$\min_{j=1,\cdots,k} \left|\left|x_i - z_j\right|\right|_2^2$$

and the total cost function is:

$$cost(z_1, \cdots, z_k) = \sum_{i=1}^{n} \min_{j=1,\cdots,k} \left|\left|x_i - z_j\right|\right|_2^2$$

$$C_j = \{i | z_j is\ the\ closest\ to\ x_i\}$$

3.2 Given partitions (clusters) $(C_1, \cdots, C_k)$ find the best representative

$$cost(C_1, \cdots, C_k) = \min_{z_1,\cdots,z_k} \sum_{j=1}^{k} \sum_{i \in C_j} \left|\left|x_i - z_j\right|\right|_2^2$$

k-means clustering

3.2 Given partitions (clusters) $(C_1, \cdots, C_k)$ find the best representative

$$cost(C_1, \cdots, C_k) = \min_{z_1, \cdots, z_k} \sum_{j=1}^{k} \sum_{i \in C_j} \left|\left| x_i - z_j \right|\right|_2^2$$

- The step 3.2 is the general way to find the minimum, but in this case and since we are assuming we have a squared euclidean distance we can find it much easier and with a lower computational complexity.

$$\frac{\partial}{\partial z_j} \sum_{i \in C_j} ||x_i - z_j||_2^2 = 2 \sum_{i \in C_j} x_i - z_j = 0$$

$$\Longrightarrow \sum_{i \in C_j} x_i = \sum_{i \in C_j} z_j$$

$$\Longrightarrow z_j = \frac{\sum_{i \in C_j} x_i}{|C_j|}$$

- An important thing to keep in mind here is that what we derived is only applicable when we have a euclidean squared distance.
- In addition, the way we solve this problem is in an alternating manner and so the convergence of the k-means algorithm is local and this means every time the initialization is changed we will have a different result.

k-means clustering – Numerical Example

- Consider four points in a 2D space, and we aim to group them into two clusters (k=2):

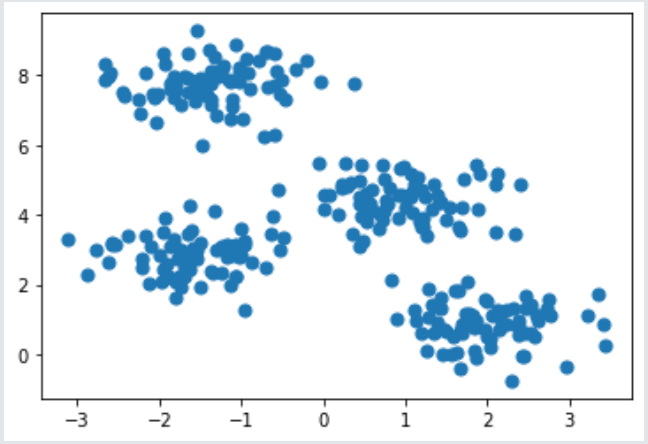
  Point A: (2,3)

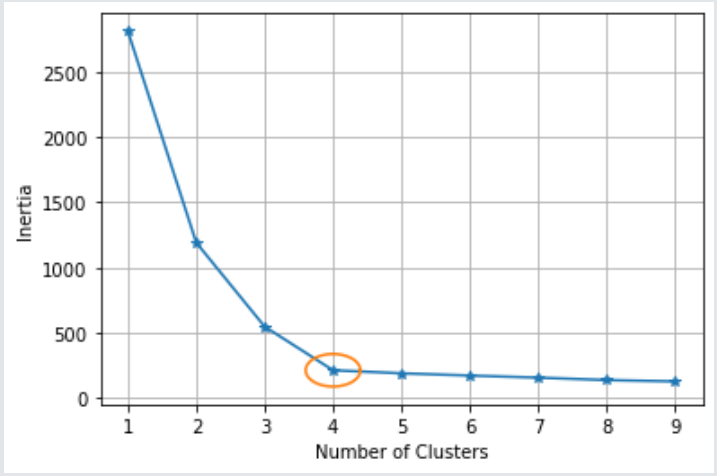
  Point B: (3,3)

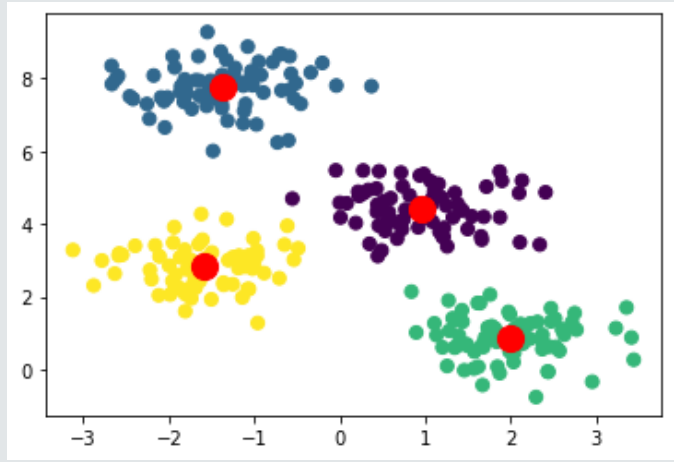
  Point C: (6,5)

  Point D: (8,8)

k-means clustering – Experiment 2



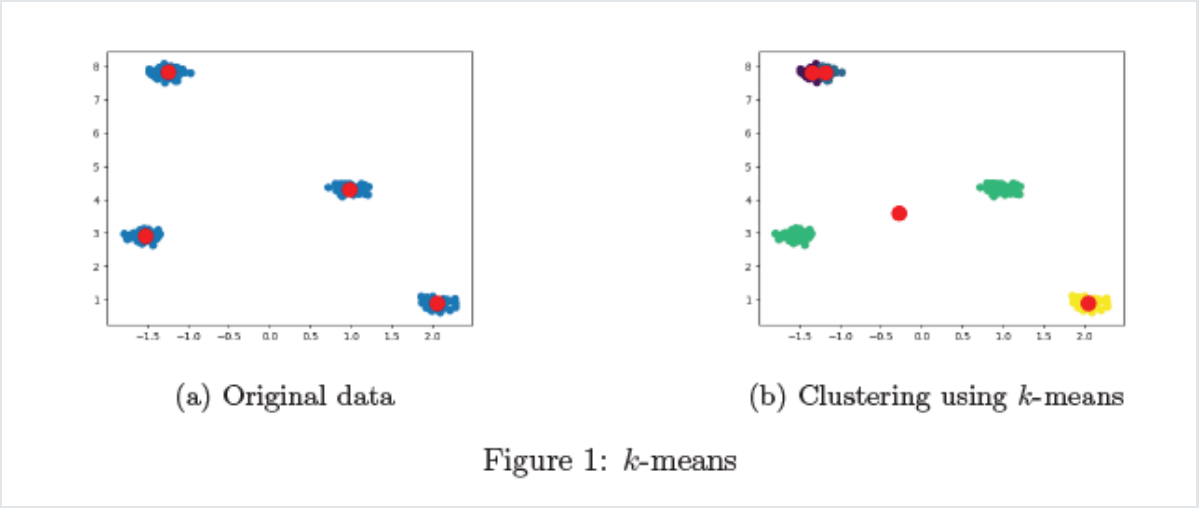




k-means clustering – Real dataset

Limitations of k-means clustering – Initialization

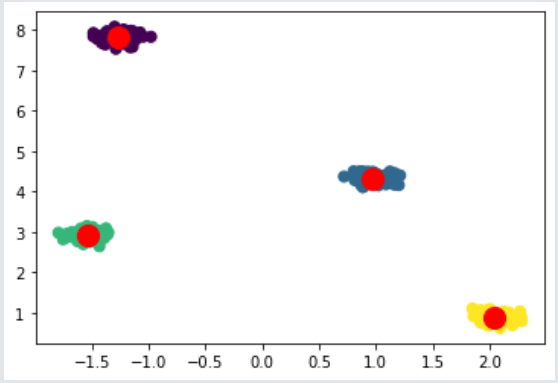
- Initialization have a major impact on the performance of k-means as shown below



(a) Original data (b) Clustering using k-means

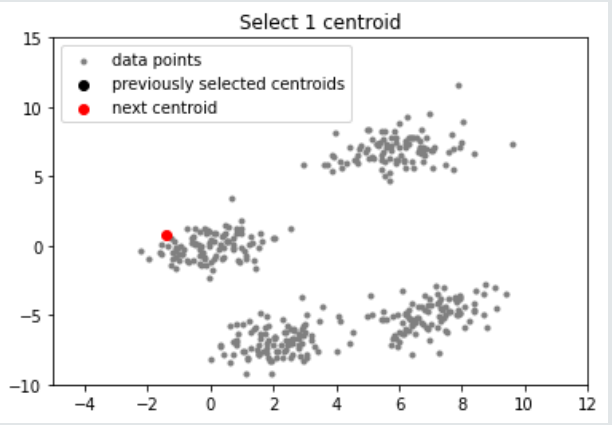
Figure 1: k-means

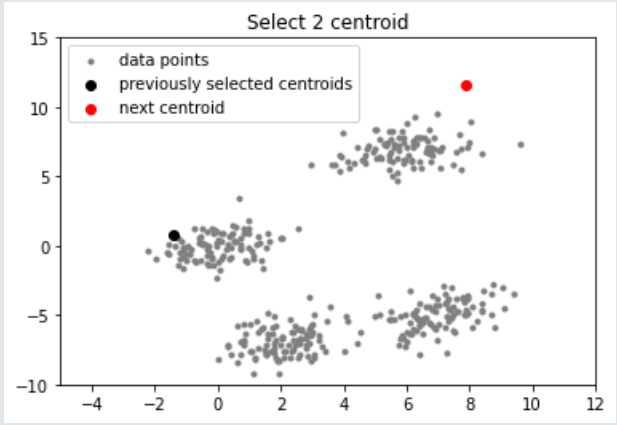
- To overcome this issue we will use the **k-means++ initialization** technique. This technique will choose representative points to be as far as possible to cover all of the clusters.

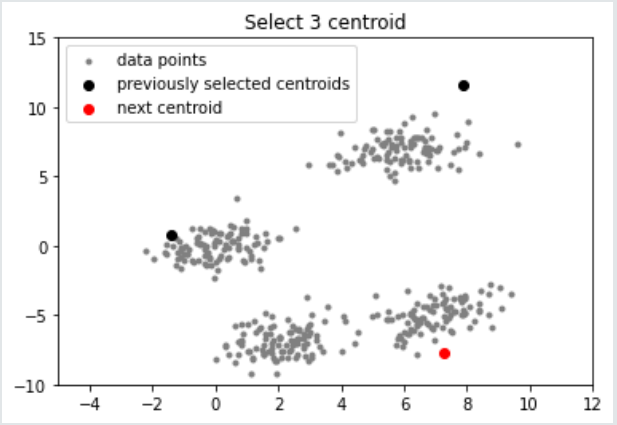


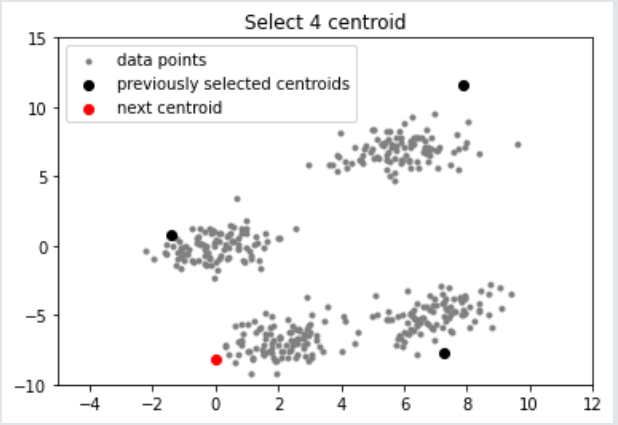
Limitations of k-means clustering – Initialization

- First point will be chosen randomly as a representative.
- Then the distance between the representative and all of the other points will be measured. Then the probability of selecting the next representative will be a function of the distance squared and this means further points will have a higher probability to be selected.
- Then this process is repeated until all points are chosen.

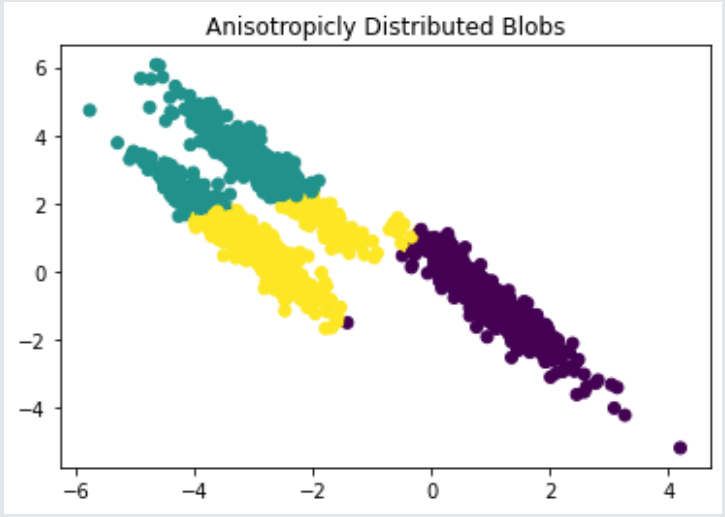


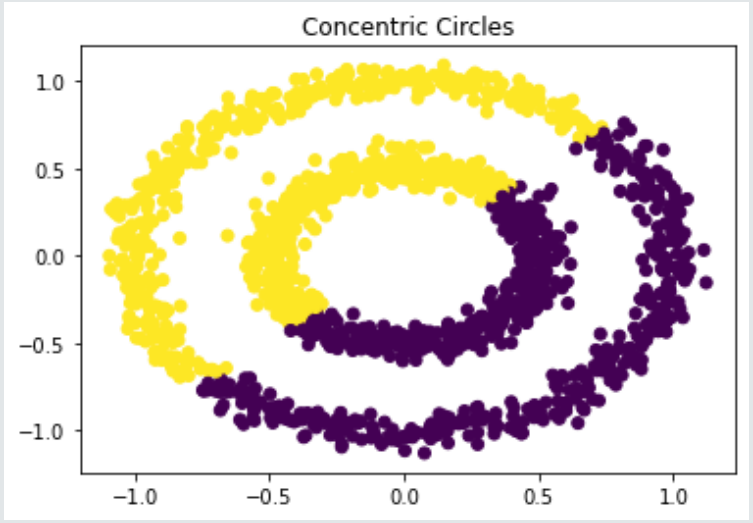


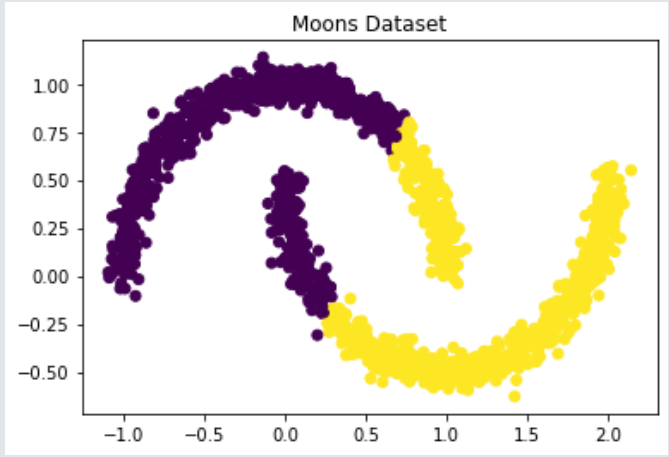




Limitations of k-means clustering – Spherical Cluster







Limitations of k-means clustering – Spherical Cluster

Kernel k-means clustering algorithm

1. Specify the number of clusters k
2. Apply a kernel function ϕ to the samples x_i and so we will have $\phi(x_i)$
3. Randomly select the representative $(z_1, \cdots z_k)$.
4. Iterate until there is no change in cost
   (a) Given $(z_1, \cdots, z_k)$, go over all $\phi(x_i)$ and for each $\phi(x_i)$ assign to it the closest z_j, more precisely:
   $$\min_{j=1,\cdots,k} ||\phi(x_i) - z_j||_2^2$$
   and the total cost function is:
   $$\text{cost}(z_1, \cdots, z_k) = \sum_{i=1}^{n} \min_{j=1,\cdots,k} ||\phi(x_i) - z_j||_2^2$$
   $$C_j = \{i | z_j \text{ is the closest to } x_i\}$$
   (b) Given partitions (clusters) $(C_1, \cdots, C_k)$, find the best representative:
   $$\text{cost}(C_1, \cdots, C_k) = \min_{z_1,\cdots,z_k} \sum_{j=1}^{k} \sum_{i \in C_j} ||\phi(x_i) - z_j||_2^2$$

Limitations of k-means clustering – Spherical Cluster

$$\frac{\partial}{\partial z_j}\sum_{i\in C_j}||\phi(x_i)-z_j||_2^2 = 2\sum_{i\in C_j}\phi(x_i)-z_j = 0$$

$$\implies \sum_{i\in C_j}\phi(x_i) = \sum_{i\in C_j} z_j$$

$$\implies z_j = \frac{\sum_{i\in C_j}\phi(x_i)}{|C_j|}$$

Therefore, when we measure the euclidean distance between the representative points z_j and the other points $\phi(x_i)$ we get something interesting.

$$||\phi(x_i)-z_j||_2^2 = \left|\left|\phi(x_i) - \frac{\sum_{t\in C_j}\phi(x_t)}{|C_j|}\right|\right|_2^2$$

$$= \phi(x_i)^T\phi(x_i) - 2\sum_{t\in C_j}\frac{\phi(x_i)^T\phi(x_t)}{|C_j|} + \sum_{t\in C_j}\frac{\phi(x_t)^T\phi(x_t)}{|C_j|^2}$$

kernel trick

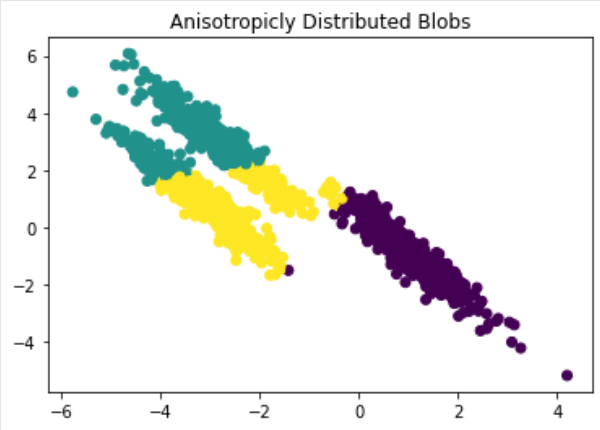
The dot product in this case will be very useful as it was very useful in the case of kernels in non-linear classification.

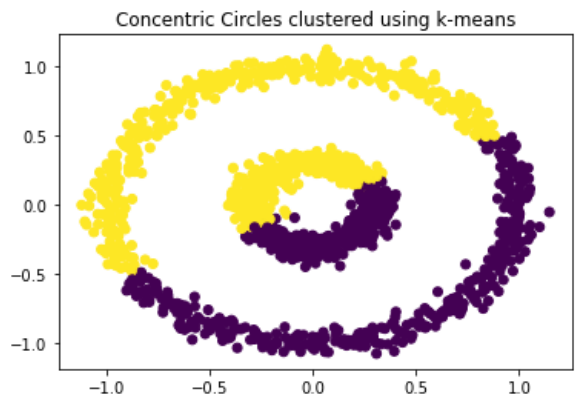
$$\phi(x_j)^T\phi(x_i) = (1+x_j^Tx_i)^d;\ \text{when the } \phi \text{ is polynomial}$$

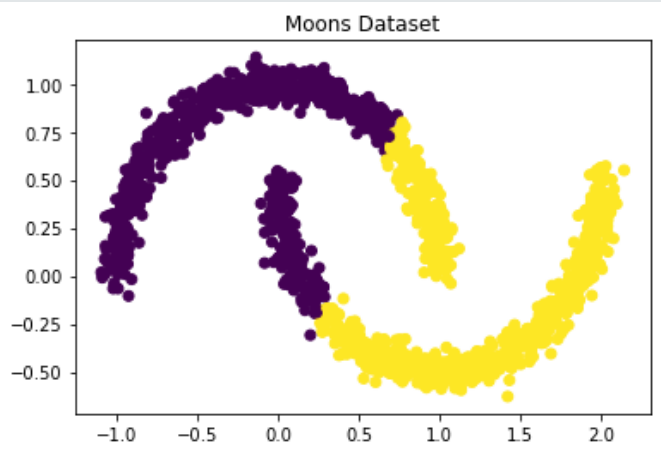
$$\phi(x_j)^T\phi(x_i) = \exp\left(-\gamma||x_j-x_i||_2^2\right);\ \text{when the } \phi \text{ is rbf}$$

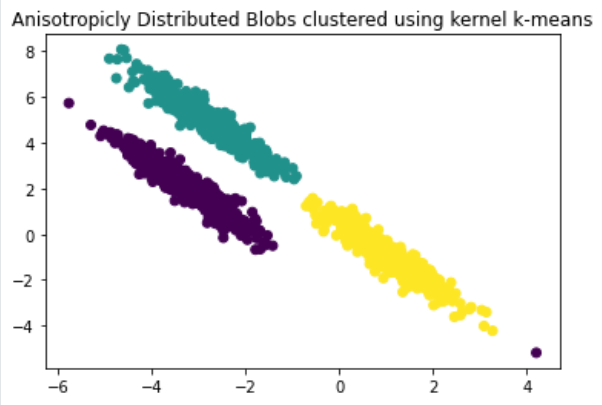
Dhillon, Inderjit S., Yuqiang Guan, and Brian Kulis. "Kernel k-means: spectral clustering and normalized cuts." *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004.

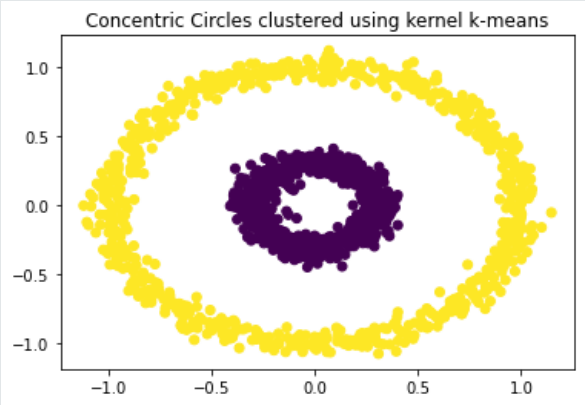
Limitations of k-means clustering – Spherical Cluster

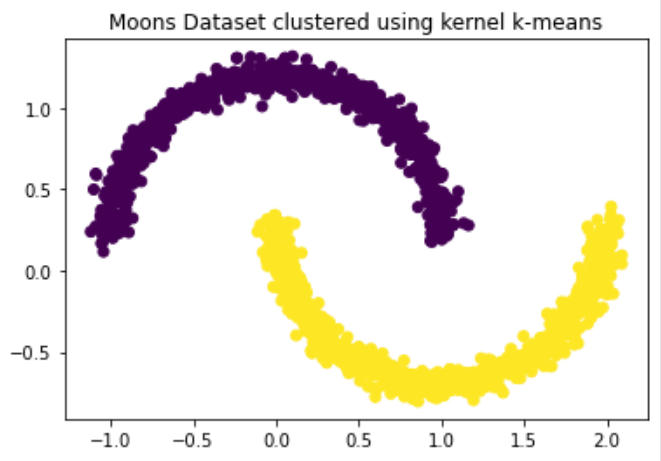












Limitations of k-means clustering – Squared Euclidean Distance

- As shown before, the k-means clustering algorithm uses only the squared Euclidean distance. What happens if we want to use another cost function? This will lead us to **k-medoids clustering.**

k-medoids clustering

The k-medoids is a generalization of the k-means to have a wider variety of distance functions.

1. Specify the number of clusters, k.
2. Randomly initialize $\{z_1, \cdots, z_k\} \subset \{x_1, \cdots, x_n\}$.
3. Iterate until there is no change in cost:
   (a) Given $z_1, \cdots, z_k$ go over all x_i, and for each x_i assign it to the closest z_j.

$$\min_{j=1,\cdots,k} \text{dist}(x_i, z_j)$$

and the total cost function

$$\text{cost}(z_1, \cdots, z_k) = \sum_{i=1}^{n} \min_{j=1,\cdots,k} \text{dist}(x_i, z_j)$$

and the clusters are

$$C_j = \{i | z_j \text{ is closest to } x_i\}$$

   (b) Given partitions (clusters) $C_1, \cdots, C_k$ find the representative which lead to the lowest cost:

$$\text{cost}(C_1, \cdots, C_k) = \sum_{j=1}^{k} \sum_{i \in C_j} \text{dist}(x_i, z_j)$$

To do this, for each z_j we will go over $\{x_1, \cdots, x_i\}$ to find the one that will result in the lowest error.

Limitations of k-means clustering – Squared Euclidean Distance