Clustering

https://shala2020.github.io/

Sayan Chatterjee (Student, IIT Bombay)
&
Ravinder Singh (Team leader DS, Xebia)

Introduction to clustering:

- Clustering involves the grouping of similar objects into a set known as cluster.
- Objects in one cluster are likely to be different when compared to objects grouped under another cluster.
- It is an unsupervised learning (labels are unknown)
- It is essentially a grouping problem.

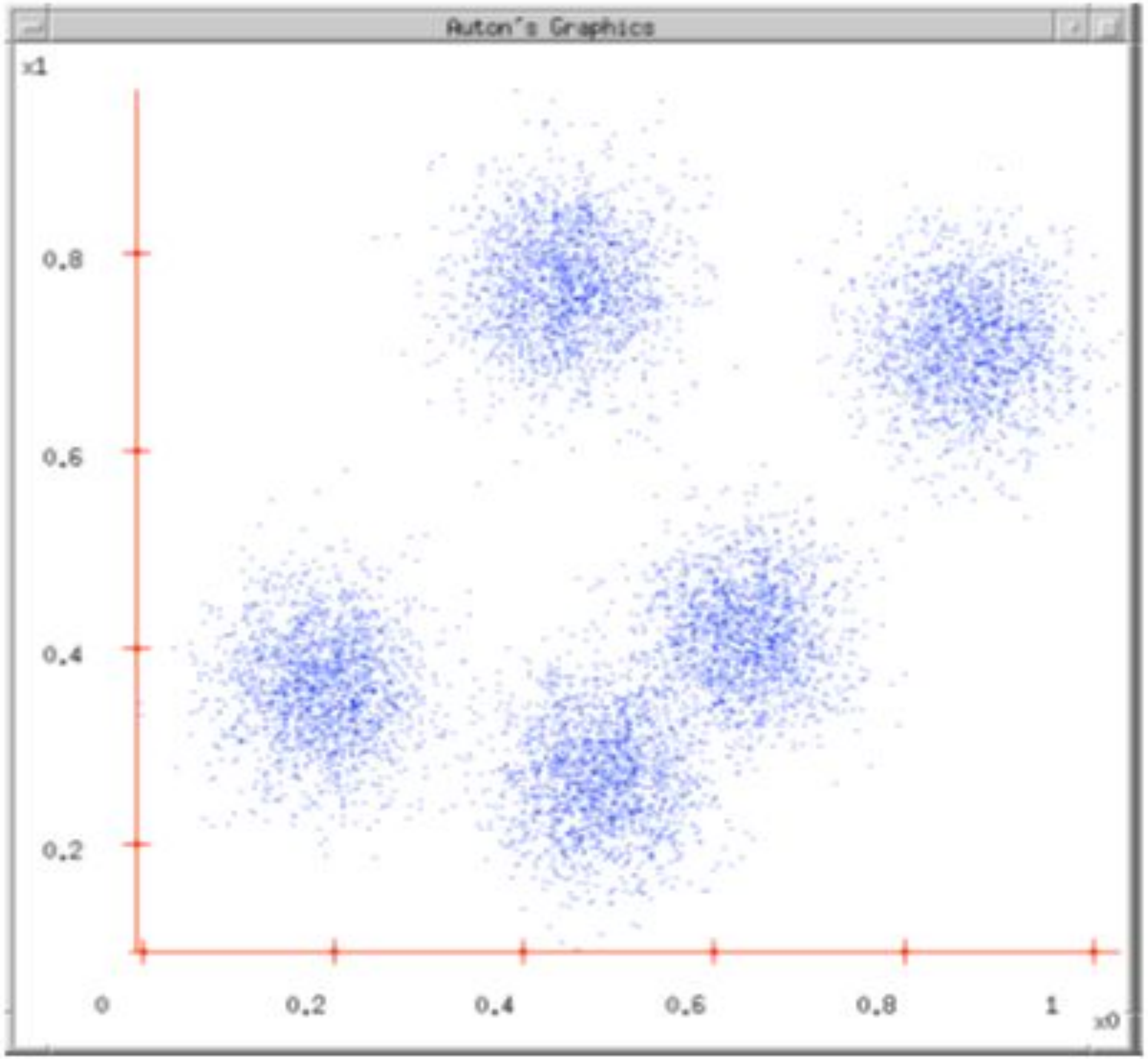
Unsupervised Learning (*Slide adapted from Andrew Moore, CMU)*

- Supervised learning used labeled data pairs (x, y) to learn a function f : X→Y.
- But, what if we don't have labels?

- No labels = **unsupervised learning**
- Only some points are labeled = **semi-supervised learning**
  - Labels may be expensive to obtain, so we only get a few.

- **Clustering** is the unsupervised grouping of data points. It can be used for **knowledge discovery**.

Can you spot the clusters here?

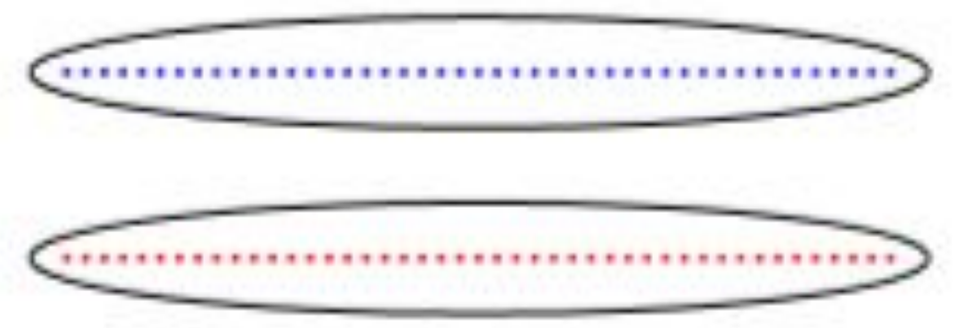


Slide adapted from Andrew Moore, CMU

Example:

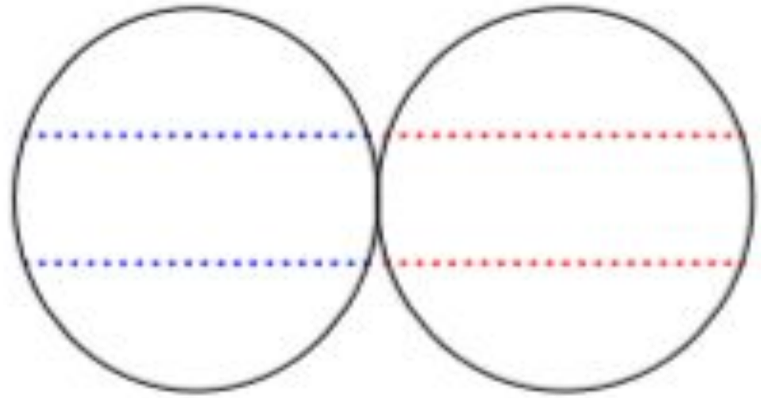
Option1

image source: Understanding machine learning



Option2

image source: Understanding machine learning



Various aspects of Clustering:

(**source :**http://www.mit.edu/~9.54/fall14/slides/Class13.pdf)

1. Proximity measure, *either*
   - similarity measure $s(\mathbf{x}_i,\mathbf{x}_k)$: large if $\mathbf{x}_i,\mathbf{x}_k$ are similar
   - dissimilarity(or distance) measure $d(\mathbf{x}_i,\mathbf{x}_k)$: small if $\mathbf{x}_i,\mathbf{x}_k$ are similar

2. Criterion function to evaluate a clustering

3. Algorithm to compute clustering
   - For example, by optimizing the criterion function

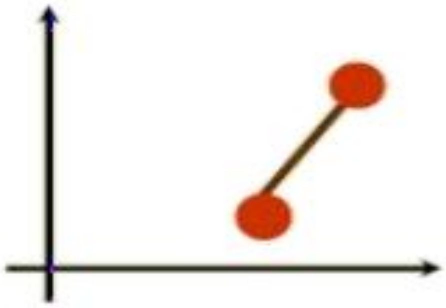
Similarity (dissimilarity) measures:

(**source :**http://www.mit.edu/~9.54/fall14/slides/Class13.pdf)

- Euclidean distance

$$d(\mathbf{x}_i,\mathbf{x}_j)=\sqrt{\sum_{k=1}^{d}\left(x_i^{(k)}-x_j^{(k)}\right)^2}$$

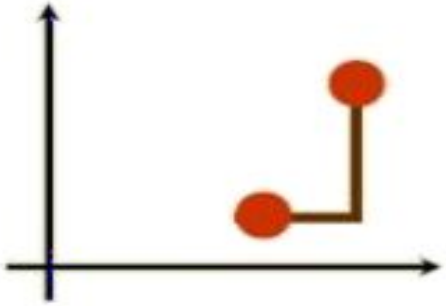
  - translation invariant



- Manhattan (city block) distance

$$d(\mathbf{x}_i,\mathbf{x}_j)=\sum_{k=1}^{d}\left|x_i^{(k)}-x_j^{(k)}\right|$$

  - approximation to Euclidean distance, cheaper to compute



- They are special cases of **Minkowski distance**:

$$d_p(\mathbf{x}_i,\mathbf{x}_j)=\left(\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right|^p\right)^{\frac{1}{p}}$$

Cluster evaluation

(**source :**http://www.mit.edu/~9.54/fall14/slides/Class13.pdf)

- **Intra-cluster cohesion** (compactness):
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- **Inter-cluster separation** (isolation):
  - Separation means that different cluster centroids should be far away from one another.
- In most applications, expert judgments are still the key

K means Algorithm (source ;http://www.mit.edu/~9.54/fall14/slides/Class13.pdf)

- K means:

1. K-means is a partitioning clustering algorithm

2. Let the set of data points D be $\{x_1, x_2, \ldots, x_n\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector and r is the number of dimensions.

3. The k-means algorithm partitions the given data into k clusters: – Each cluster has a cluster center, called centroid.

4. k is specified by the user beforehand.

K-means objective function

The centroid of C_i is defined to be

$$\mu_i(C_i) = \operatorname*{argmin}_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu)^2.$$

Then, the k-means objective is

$$G_{\mathrm{k-means}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i))^2.$$

This can also be rewritten as

$$G_{\mathrm{k-means}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2. \qquad (22.1)$$

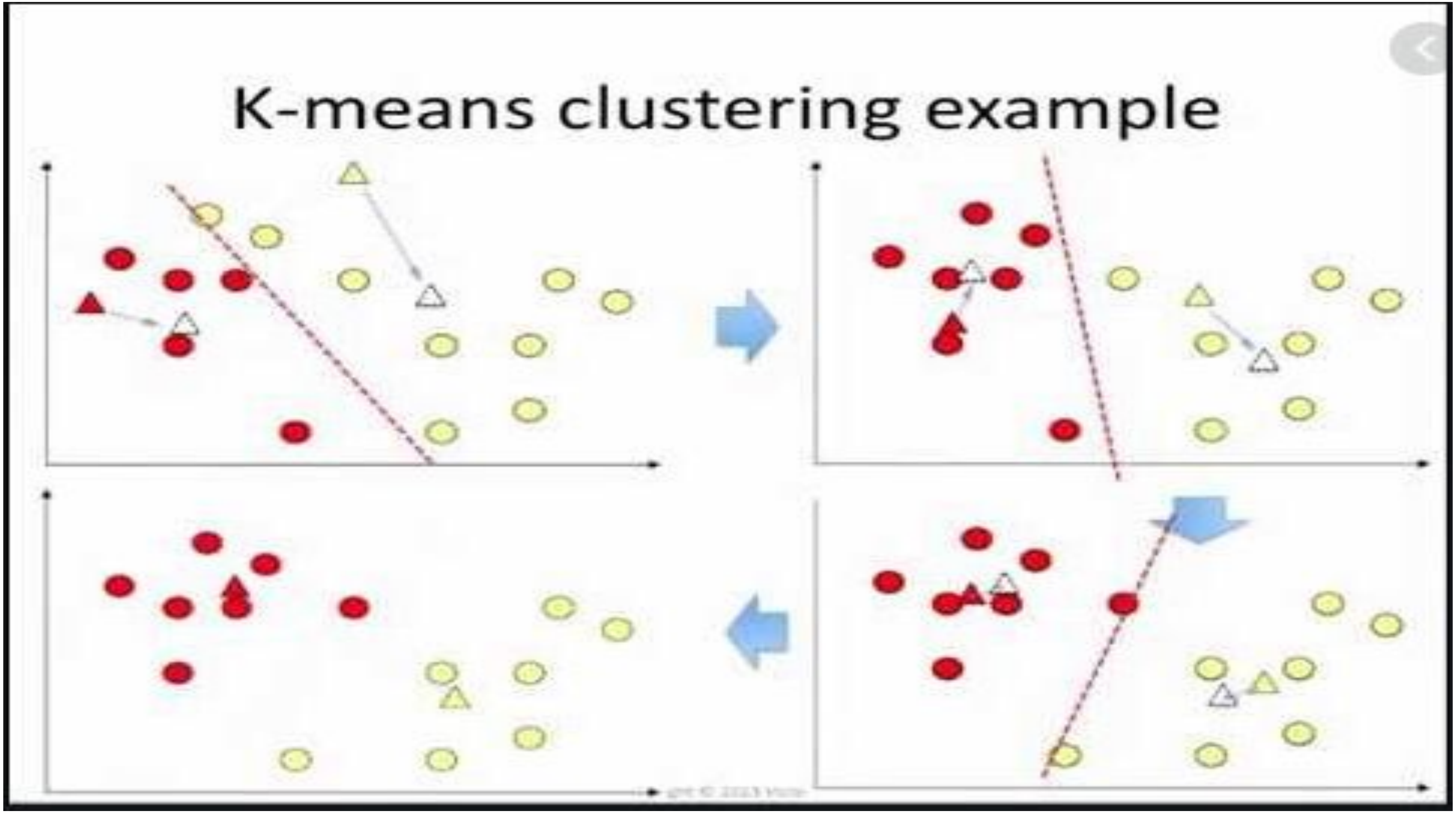
K means Algorithm

(source:http://www.mit.edu/~9.54/fall14/slides/Class13.pdf)

- Given k, the k-means algorithm works as follows:

1. Choose k (random) points (seeds) to be the initial centroids, cluster centers.
2. Assign each data point to the closest centroid.
3. Re-compute the centroids using the current cluster memberships.
4. If a convergence criterion is not met, repeat steps 2 and 3.

(Source: https://images.app.goo.gl/1QfbDT9aqu2nkjc2A)



Strengths and weakness of K means algorithm:

(*Note: Adapted from R. Palaniappan)*

- Strengths
  - Relatively efficient: where *N* is no. objects, *K* is no. clusters, and *T* is no. iterations. Normally, *K, T* << *N.*
  - Procedure always terminates successfully

- Weaknesses
  - Does not necessarily find the most optimal configuration
  - Significantly sensitive to the initial randomly selected cluster centres
  - Applicable only when mean is defined (i.e. can be computed)
  - Need to specify *K*, the number of clusters, in advance

Variants of k-means:

- **The k-medoids objective function** is similar to the k-means objective, except that it requires the cluster centroids to be members of the input set. The objective function is defined by

$$G_{\mathrm{K-medoid}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2.$$

- **The k-median objective function** is quite similar to the k-medoids objective, except that the "distortion" between a data point and the centroid of its cluster is measured by distance, rather than by the square of the distance:

$$G_{\mathrm{K-median}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i).$$

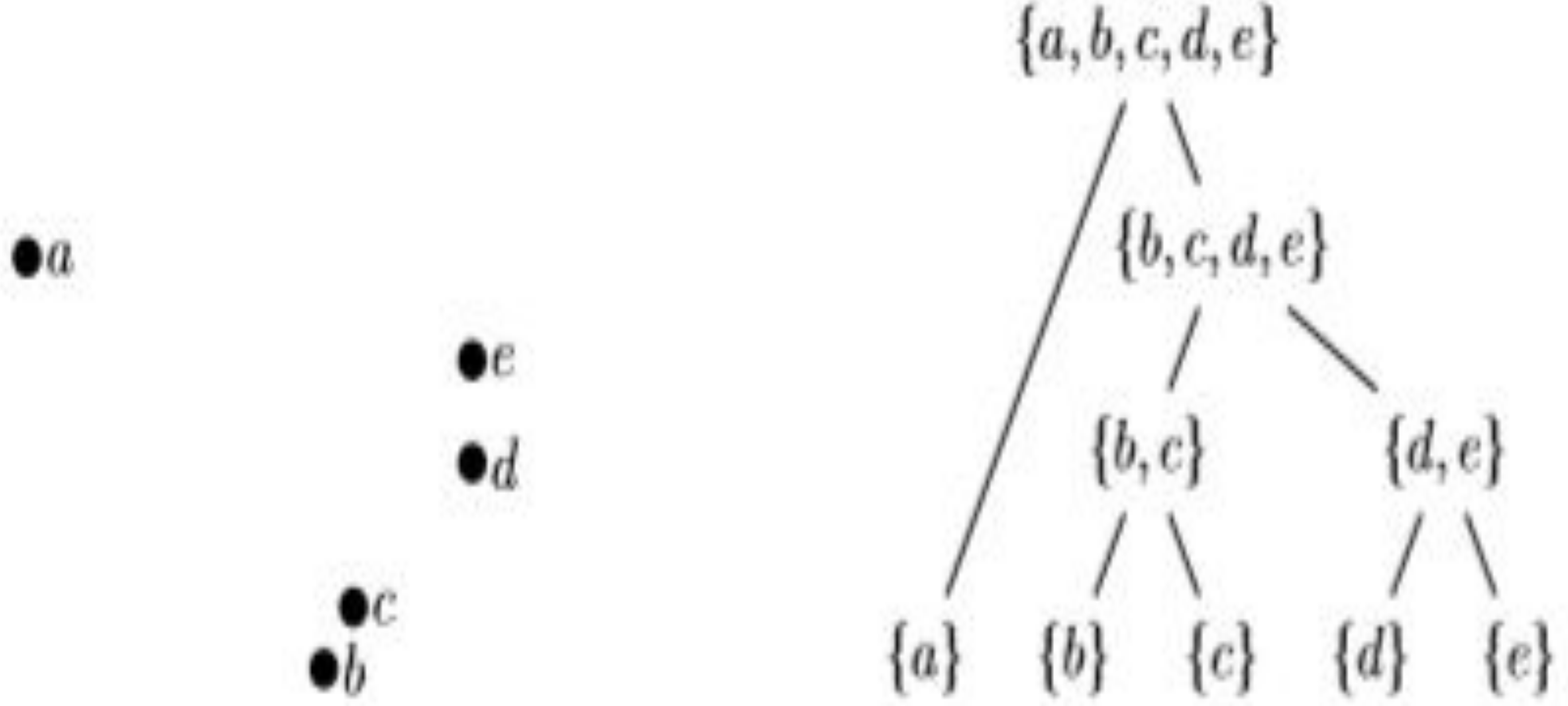
DBSCAN:

- Consider a N dimensional dataset $x_1, x_2, \ldots, x_N \ldots$
- Select a value of minpts>0 and rad>0
- Choose one point from the dataset randomly and consider a disk around it of radius = rad. Now if the no of points exceed minpts, make it a cluster(say A_i), otherwise mark it as noise and leave it.
- Repeat the last step for all the N points.
- Combine A_i and A_j (take the union) if their intersection set is non empty.
- Repeat last step until you find no such further unions.

Advantages: (source:https://en.wikipedia.org/wiki/DBSCAN#Algorithm)

- DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k means.
- DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster.
- DBSCAN has a notion of noise, and is robust to outliers.

Agglomerative clustering:
(image taken from: understanding machine learning)



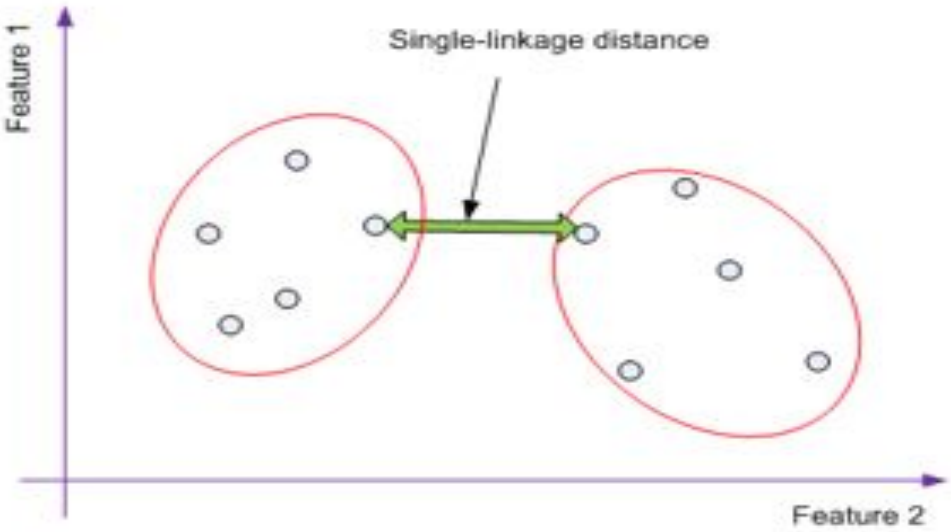
Agglomerative clustering criteria

- Single linkage

$$D(A,B) \stackrel{\text{def}}{=} \min\{d(x,y) : x \in A,\ y \in B\}$$

- Average linkage

$$D(A,B) \stackrel{\text{def}}{=} \frac{1}{|A||B|} \sum_{x \in A,\ y \in B} d(x,y)$$



- Max linkage

$$D(A,B) \stackrel{\text{def}}{=} \max\{d(x,y) : x \in A,\ y \in B\}.$$

Agglomerative clustering

- K-means approach starts out with a fixed number of clusters and allocates all data into the exactly number of clusters
- But agglomeration does not require the number of clusters *K* as an input
- Agglomeration starts out by forming each data as one cluster
  - So, data of N object will have N clusters
- Next by using some distance (or similarity) measure, reduces the number so clusters (one in each iteration) by merging process
- Finally, we have one big cluster than contains all the objects

Hierarchical clustering:

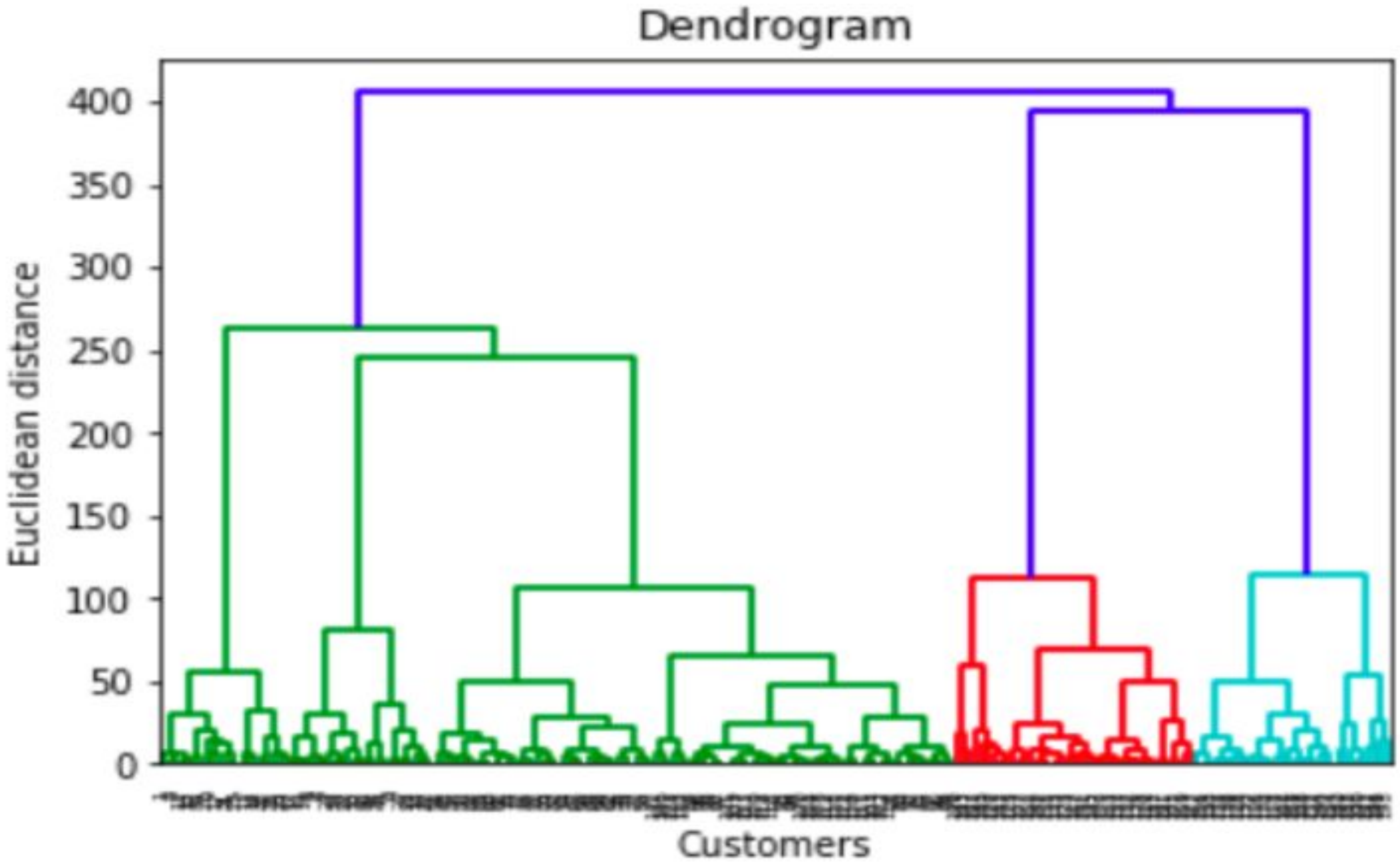
It is one of the Agglomerative clustering.

Approaches:

- Top down v/s bottom up approach
- Agglomerative v/s Divisive algorithm

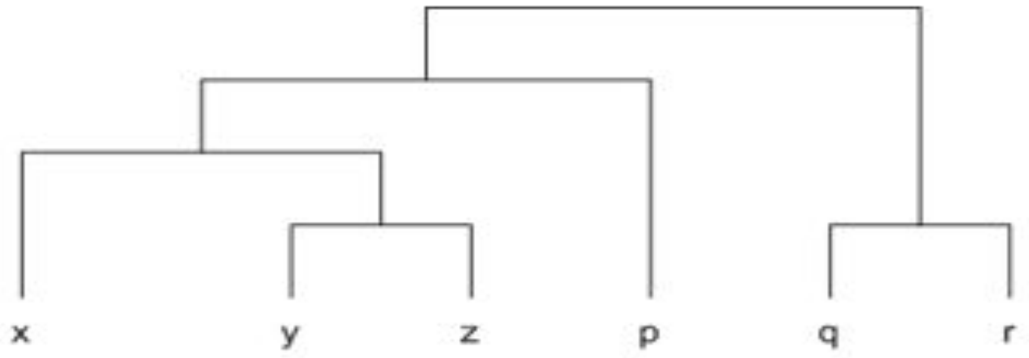
Hierarchical clustering: (continued)

(source: https://www.kdnuggets.com/2019/09/hierarchical-clustering.html)



Dendrogram

- While merging cluster one by one, we can draw a tree diagram known as dendrogram
- **Dendrograms** are used to represent agglomerative clustering
- From dendrograms, we can get any number of clusters
- E.g.: say we wish to have 2 clusters, then *cut the top one link*
  - Cluster 1: q, r
  - Cluster 2: x, y, z, p
- Similarly for 3 clusters, cut 2 top links
  - Cluster 1: q, r
  - Cluster 2: x, y, z
  - Cluster 3: p



A dendrogram example

Hierarchical clustering – algorithm

(source: *Adapted from R. Palaniappan)*

- **Hierarchical clustering algorithm is a type of agglomerative clustering**

- Given a set of *N* items to be clustered, hierarchical clustering algorithm:

  1. Start by assigning each item to its own cluster, so that if you have N items, you now have *N* clusters, each containing just one item

  2. Find the closest distance (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster

  3. Compute pairwise distances between the new cluster and each of the old clusters

  4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size *N*

  5. Draw the dendogram, and with the complete hierarchical tree, cut accordingly.

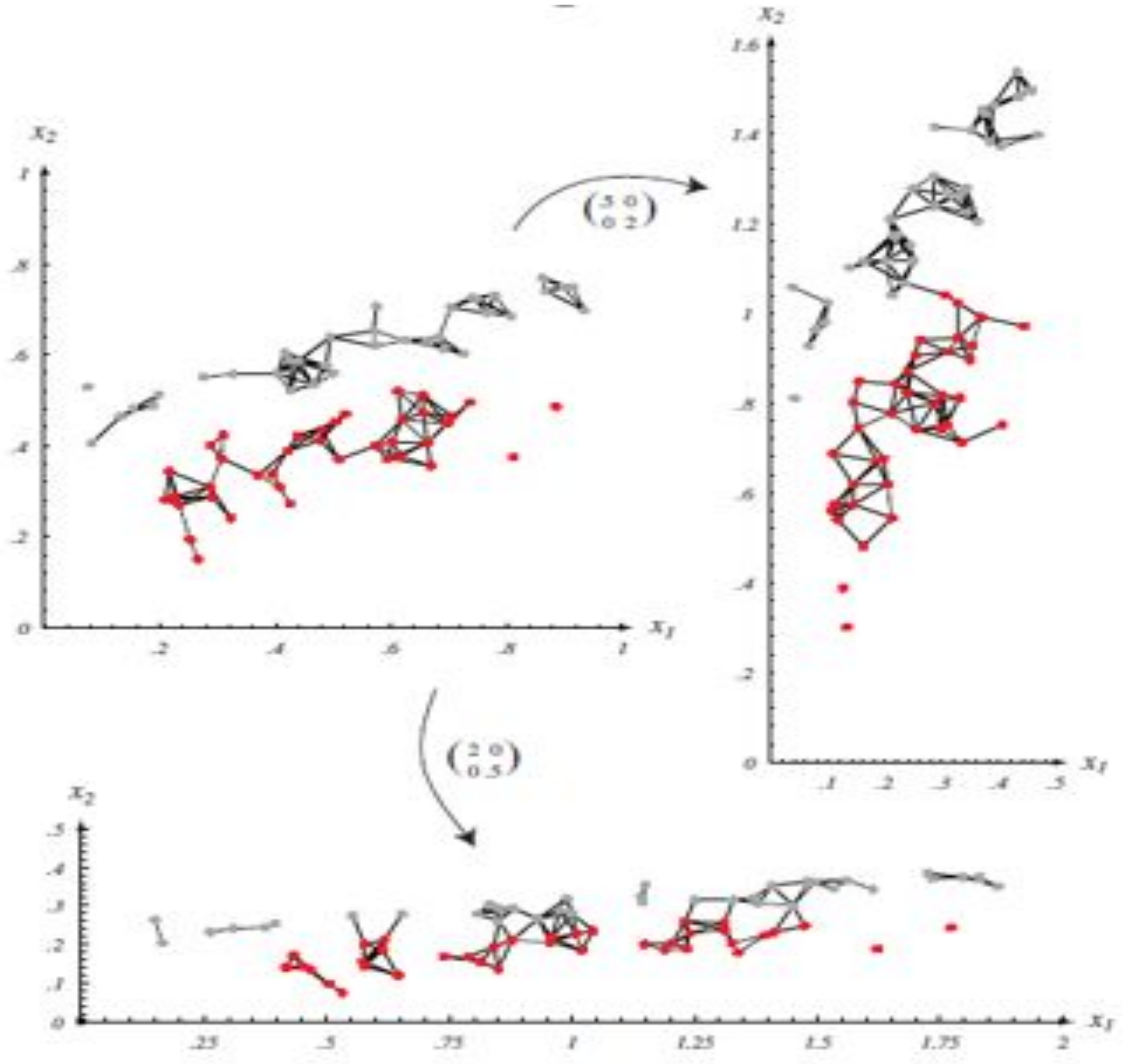
Note any distance measure can be used: Euclidean, Manhattan, etc.

Where will you cut a dendogram?

- Fixed number of clusters

- Distance-based upper bound

Impact of scaling a dimension



Clusterability:

- Clusterability is a measure of clustered structure in a data set
- Clusterability aims to determine whether a data set can be meaningfully clustered.
- Notions of clusterability tell us how much inherent cluster structure data possesses. Notions of clusterability quantify the degree of clustered structure in a data set.
- Further readings and References:

https://maya-ackerman.com/wp-content/uploads/2018/09/Clusterability.pdf

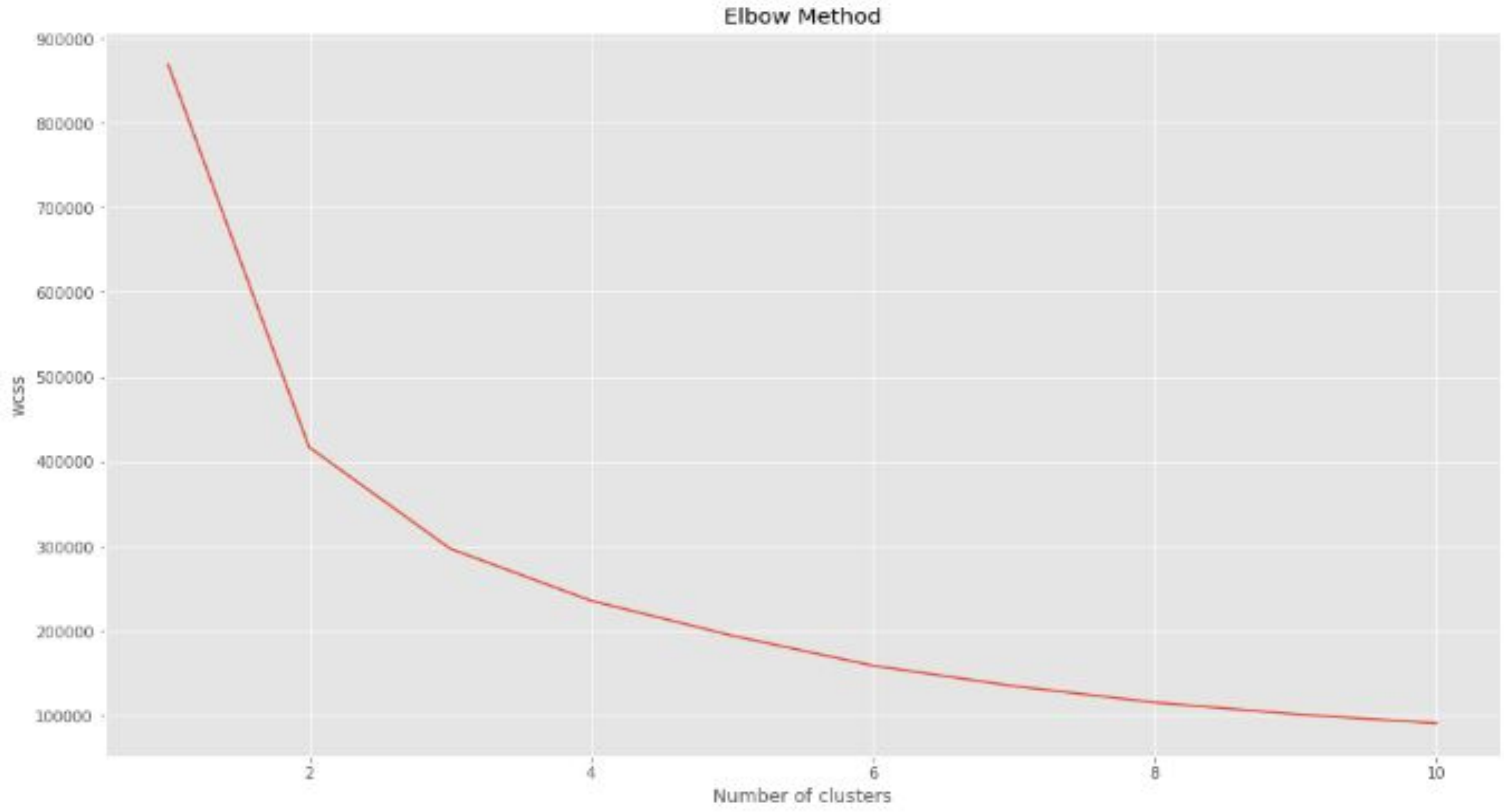
One Example:

- Generate 50 data points from a normal distribution with parameter u1(here u1 is a vector with 2 elements mean and variance)
- Generate 50 data points from a normal distribution with parameter u2(here u2 is a vector with 2 elements mean and variance) such that u1 is not equals to u2.
- Apply k means with k=4
- You will get four clusters.
- Is this correct clustering?

Clustering Quality Criteria

- Good clusters will have low inter–cluster similarity, i.e. high variance among inter-cluster members in addition to high intra–cluster similarity, i.e. low variance among intra-cluster members
- One good measure of clustering quality is Davies-Bouldin index
- The others are:
  - Dunn’s Validity Index
  - Silhouette method
  - C–index
  - Goodman–Kruskal index
- So, we compute DB index for different number of clusters, *K* and the best value of DB index tells us on the appropriate *K* value or on how good is the clustering method.
- This is how you should evaluate no of clusters.

One way to find no of clusters(taken from a blog from towardsdatascience.com)



Scree plot of given datatset on customer Income & Spend

Thank You