Unit-V

Clustering: k-means

Unsupervised learning

- Unsupervised learning:
  - Data with no target attribute. Describe hidden structure from unlabeled data.
  - Explore the data to find some intrinsic structures in them.
- Clustering: the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other clusters.
- Useful for
  - Automatically organizing data.
  - Understanding hidden structure in data.
  - Preprocessing for further analysis.

Clustering: k-means

What is K-Means Clustering?

Activate Windows
Go to Settings to activate Windo

Clustering: k-means

What is K-Means Clustering?

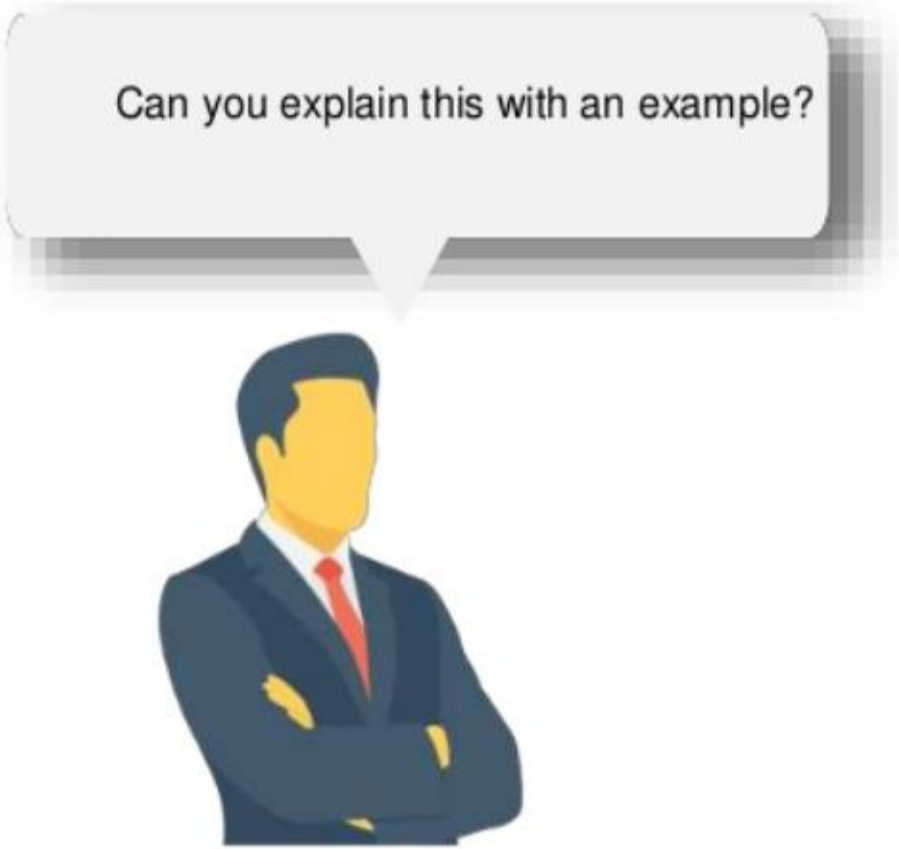
What is K-Means Clustering?

k-means performs division of objects into clusters which are “similar” between them and are “dissimilar” to the objects belonging to another cluster

Activate Windows

Clustering: k-means

What is K-Means Clustering?



Activate Windows
Go to Settings to activate Windows.

Clustering: k-means

What is K-Means Clustering?

Can you explain this with an example?

Sure. For understanding K-Means in a better way, let's take an example of **Cricket**

Activate Windows
Go to Settings to activate Win

Clustering: k-means

What is K-Means Clustering?

Task: Identify bowlers and batsmen

Activate Windows
Go to Settings to activ

Clustering: k-means

What is K-Means Clustering?

Task: Identify bowlers and batsmen

- The data contains runs and wickets gained in the last 10 matches
- So, the bowler will have more wickets and the batsmen will have higher runs

Scores

Activate Windows
Go to Settings to activate Windows.

Clustering: k-means

What is K-Means Clustering?

Assign data points

Here, we have our dataset with x and y coordinates

Now, we want to cluster this data using **K-Means**

Activate Windows
Go to Settings to activate

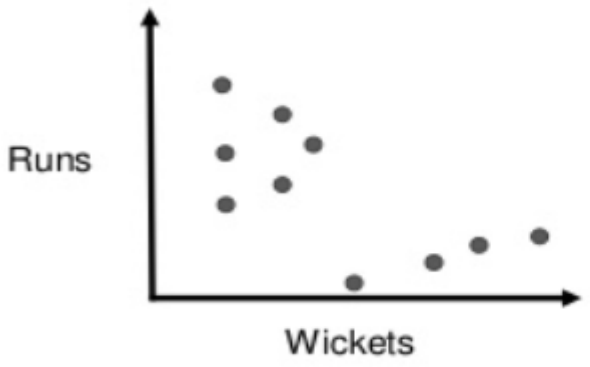
Clustering: k-means

What is K-Means Clustering?

Assign data points

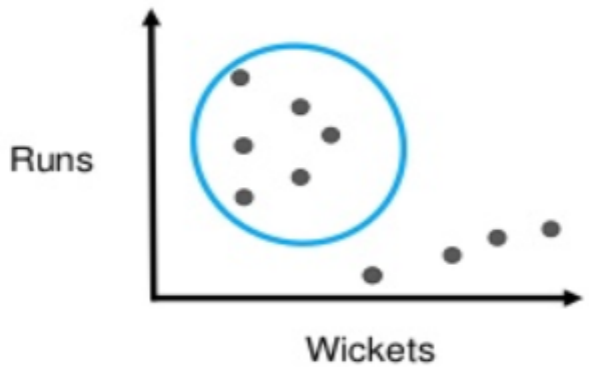
Here, we have our dataset with x and y coordinates

Now, we want to cluster this data using **K-Means**



Cluster 1

We can see that this cluster has players with high runs and low wickets



Activate Windows
Go to Settings to activate Windows

Clustering: k-means

What is K-Means Clustering?

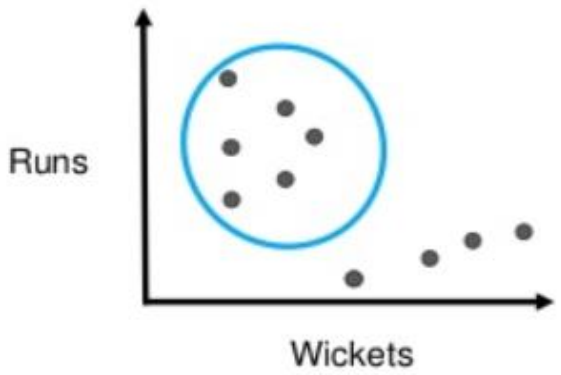
Assign data points

Here, we have our dataset with x and y coordinates

Now, we want to cluster this data using **K-Means**

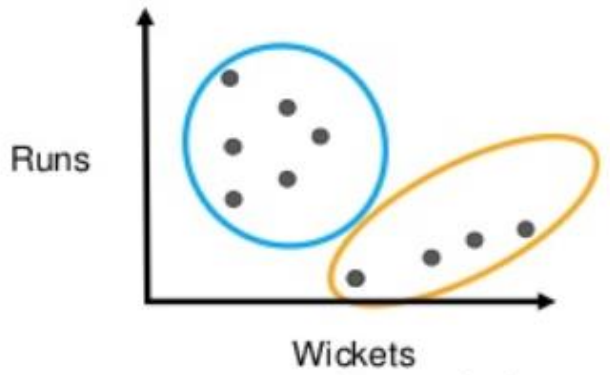
Cluster 1

We can see that this cluster has players with high runs and low wickets



Cluster 2

And here, we can see that this cluster has players with high wickets and low wickets



Activate Windows

Clustering: k-means

What is K-Means Clustering?

Consider the same data set of cricket

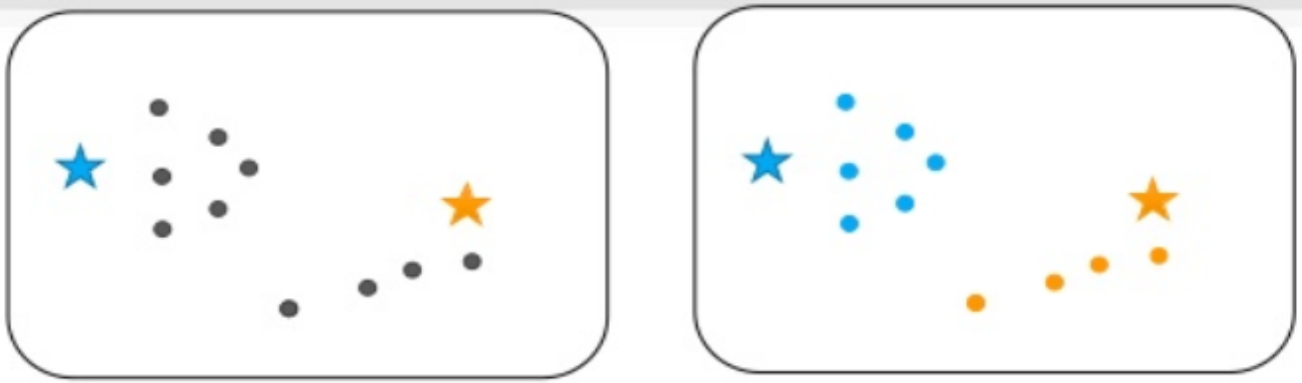
Solve the problem using K-Means

Clustering: k-means

What is K-Means Clustering?

Initially, two centroids are assigned randomly

Euclidean distance to find out which centroid is closest to each data point and the data points are assigned to the corresponding centroids

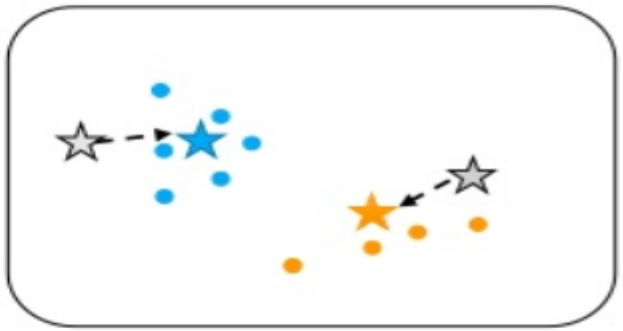


Activate Windows
Go to Settings to activate Windows.

Clustering: k-means

What is K-Means Clustering?

Reposition the two centroids for optimization.

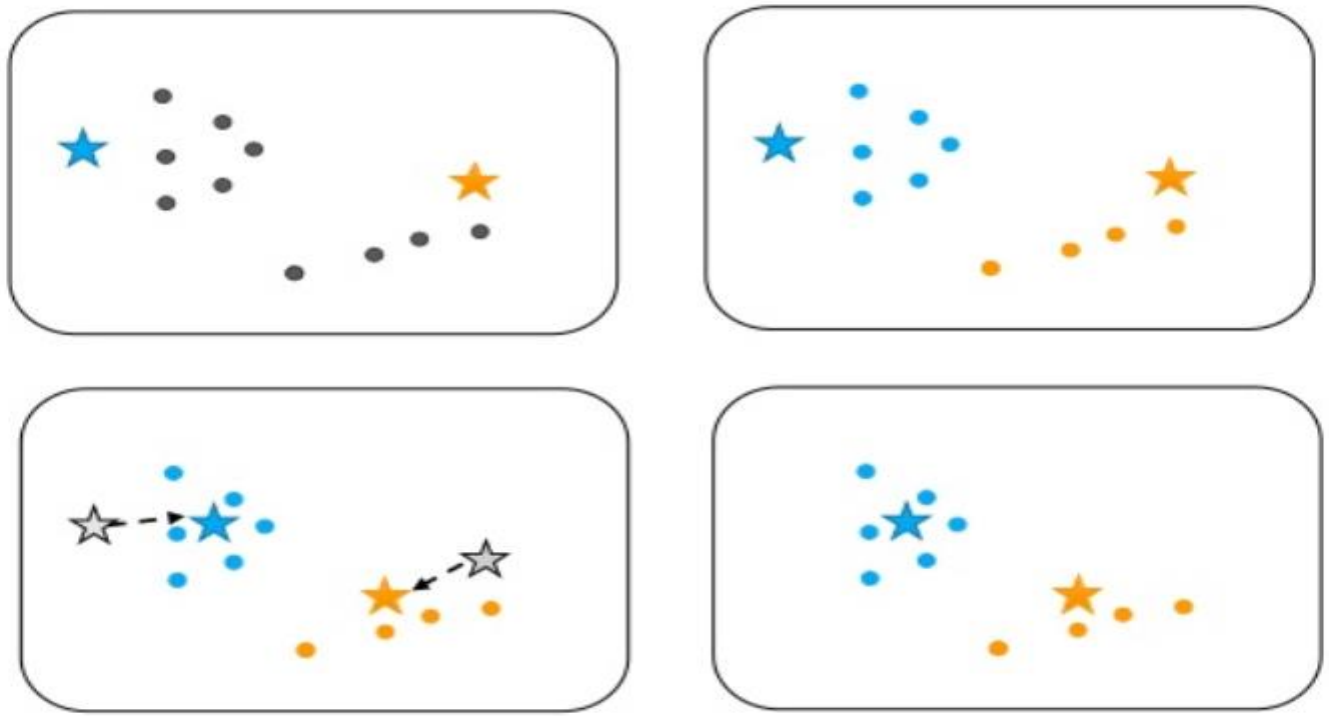


Activate Windows
Go to Settings to activate Windows

Clustering: k-means

What is K-Means Clustering?

The process is iteratively repeated until our centroids become static



Activate Windows
Go to Settings to activate Windows.

Clustering: k-means
Unsupervised Learning

What's in it for you?

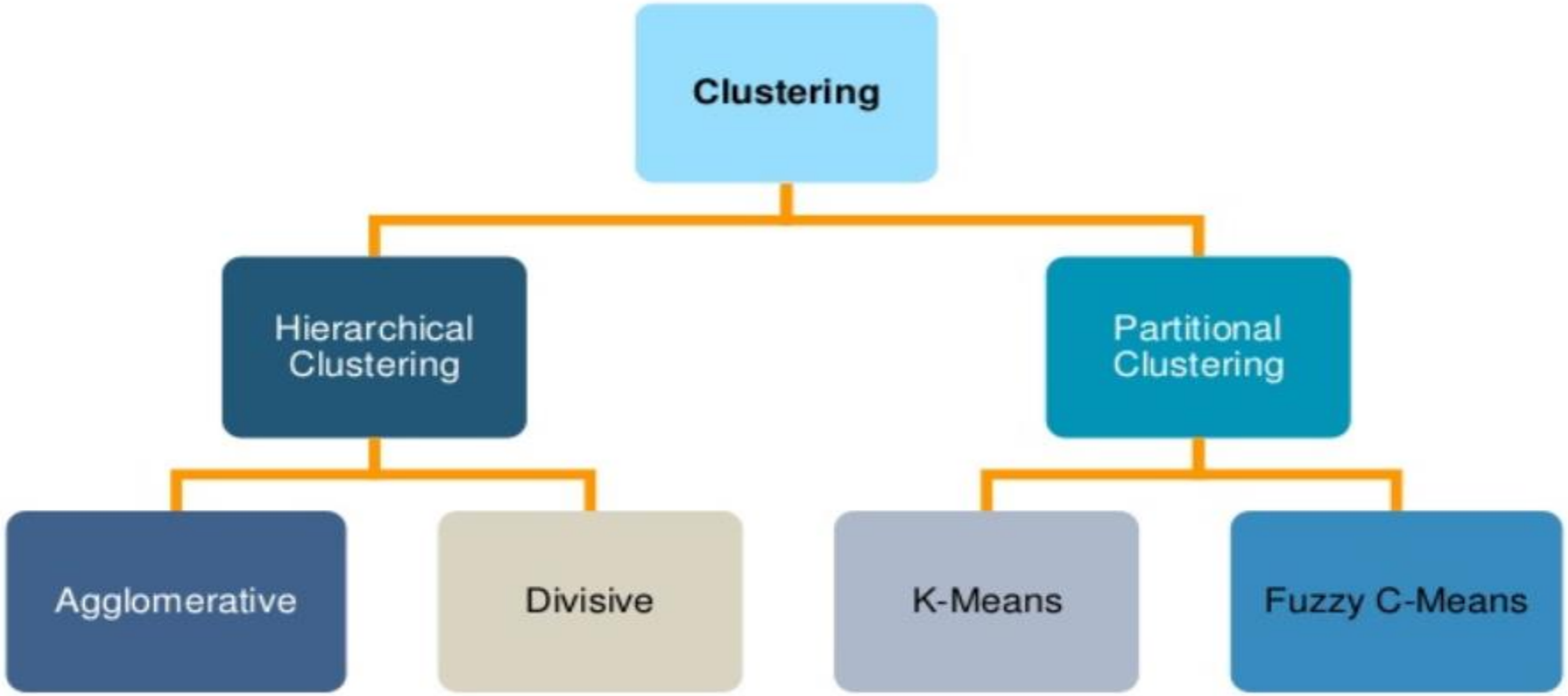
- Types of Clustering
- What is K-Means Clustering?
- Applications of K-Means clustering
- Common distance measure
- How does K-Means clustering work?
- K-Means Clustering Algorithm
- Demo: K-Means Clustering
- Use Case: Color Compression

Activate Windows

Clustering: k-means

Clustering: k-means

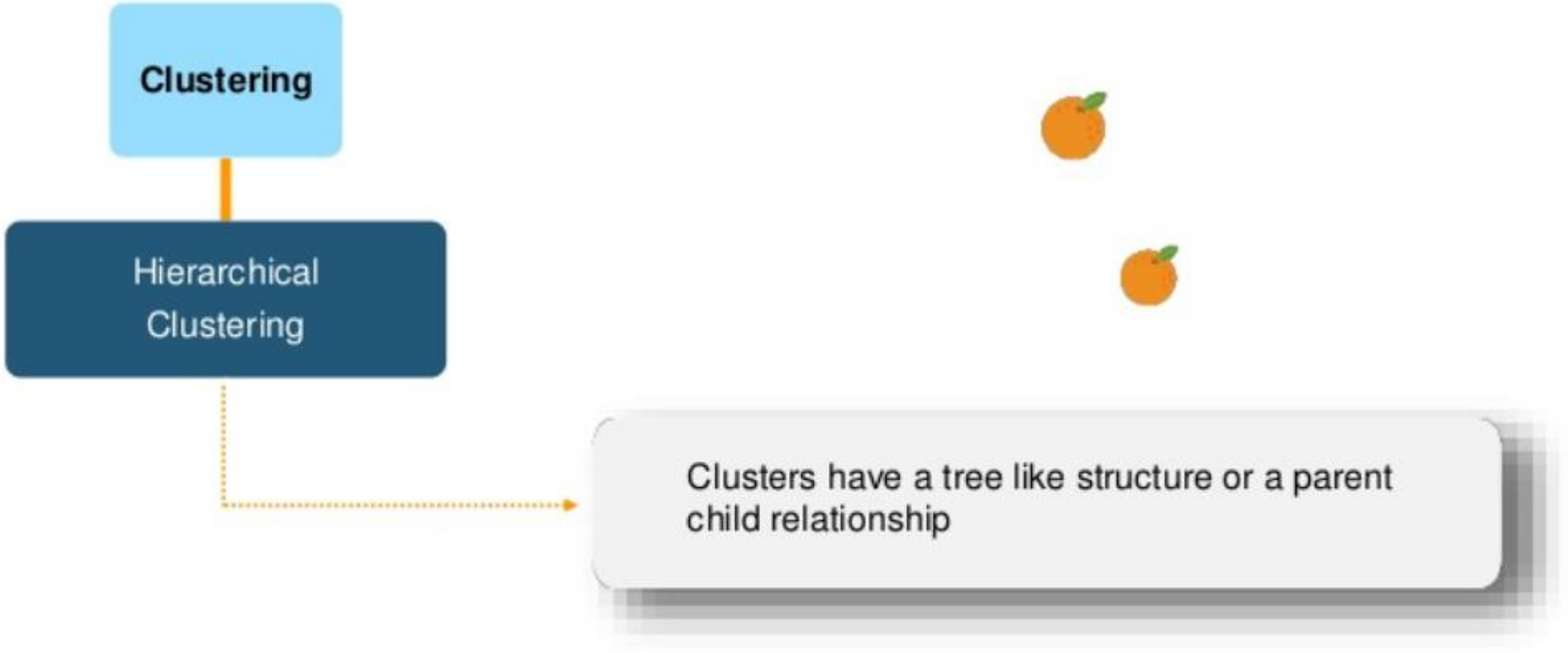
Types of Clustering



Activate Windows
Go to Settings to activate Windo

Clustering: k-means

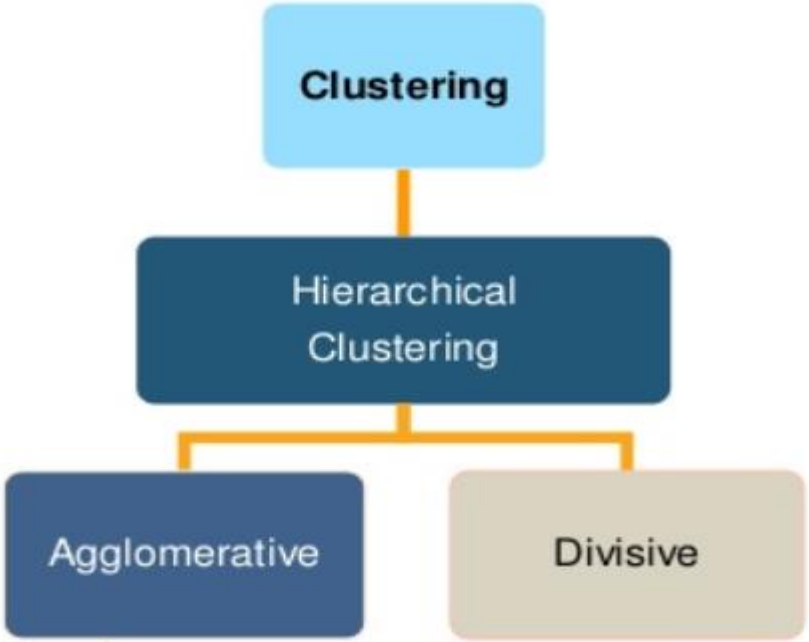
Types of Clustering

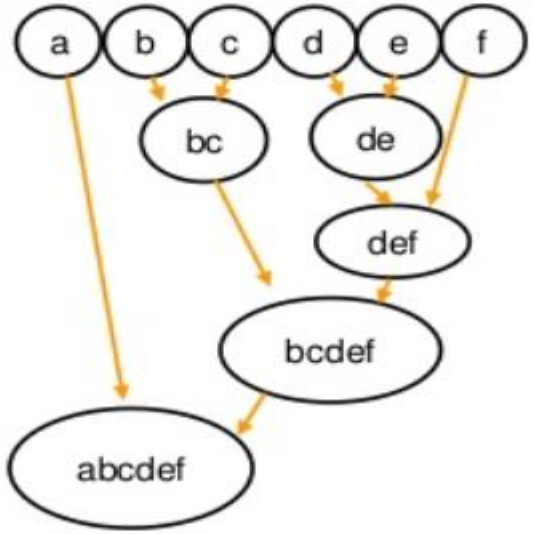


Activate Windows

Clustering: k-means

Types of Clustering



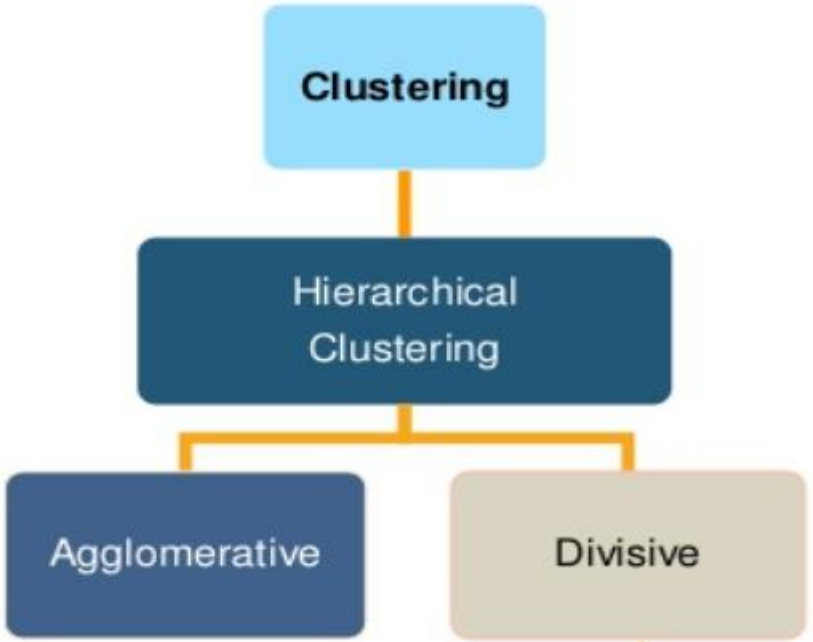


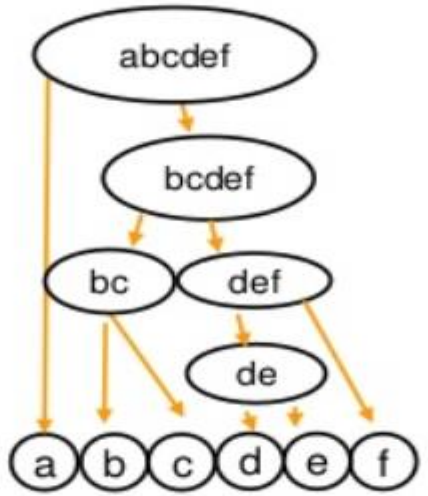
"Bottom up" approach: Begin with each element as a separate cluster and merge them into successively larger clusters

Activate Windo
Go to Settings to a

Clustering: k-means

Types of Clustering



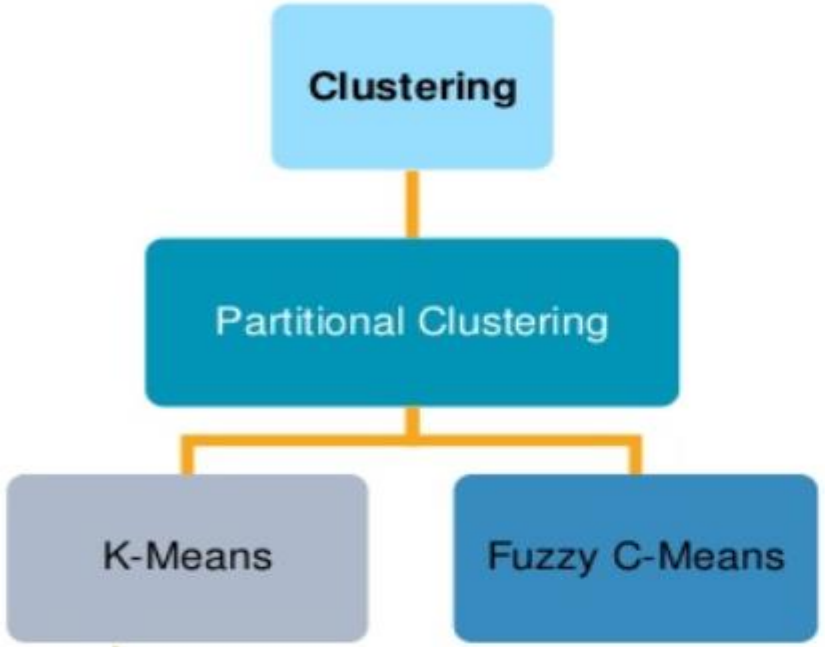


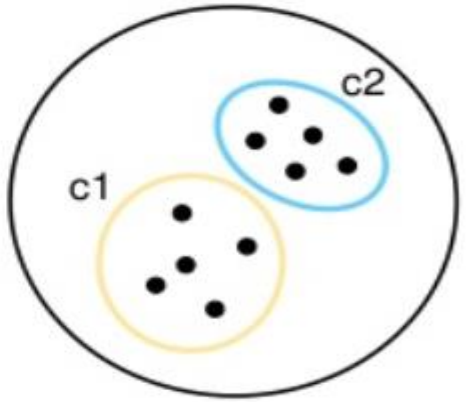
"Top down" approach begin with the whole set and proceed to divide it into successively smaller clusters.

Activate W
Go to Setting

Clustering: k-means

Types of Clustering



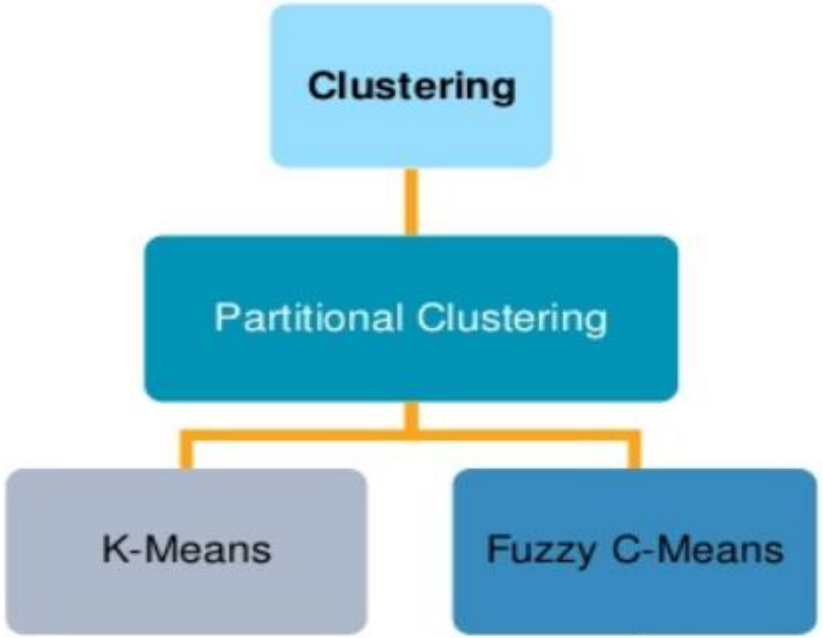


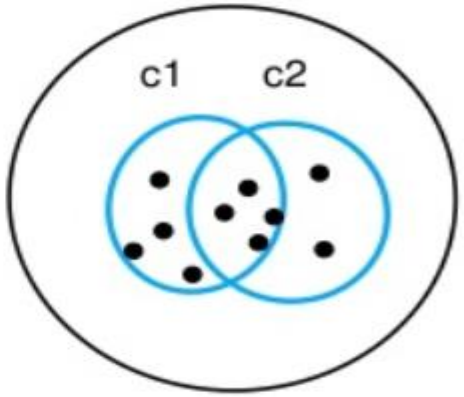
Division of objects into clusters such that each object is in exactly one cluster, not several

Activate W
Go to Settings

Clustering: k-means

Types of Clustering





Division of objects into clusters such that each object can belong to multiple clusters

Clustering: k-means

Types of Clustering

- Partitioning: Construct various partitions and then evaluate them by some criterion
- Hierarchical: Create a hierarchical decomposition of the set of objects using some criterion
- Model-based: Hypothesize a model for each cluster and find best fit of models to data
- Density-based: Guided by connectivity and density functions
- Graph-Theoretic Clustering

Clustering: k-means

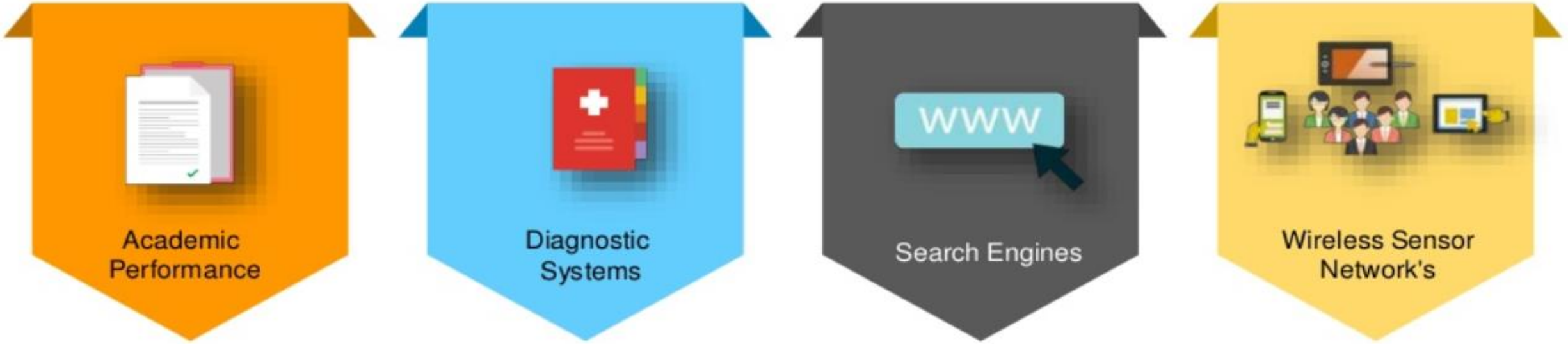
Partitioning Algorithms

- Partitioning method: Construct a partition of a database *D* of *m* objects into a set of *k* clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic method: *k-means* (MacQueen, 1967)

Clustering: k-means

Clustering: k-means

Applications of K-Means Clustering



Activate Windows

Clustering: k-means

Applications of K-Means Clustering

Other Applications

- Biology: classification of plants and animal kingdom given their features
- Marketing: Customer Segmentation based on a database of customer data containing their properties and past buying records
- Clustering weblog data to discover groups of similar access patterns.
- Recognize communities in social networks.

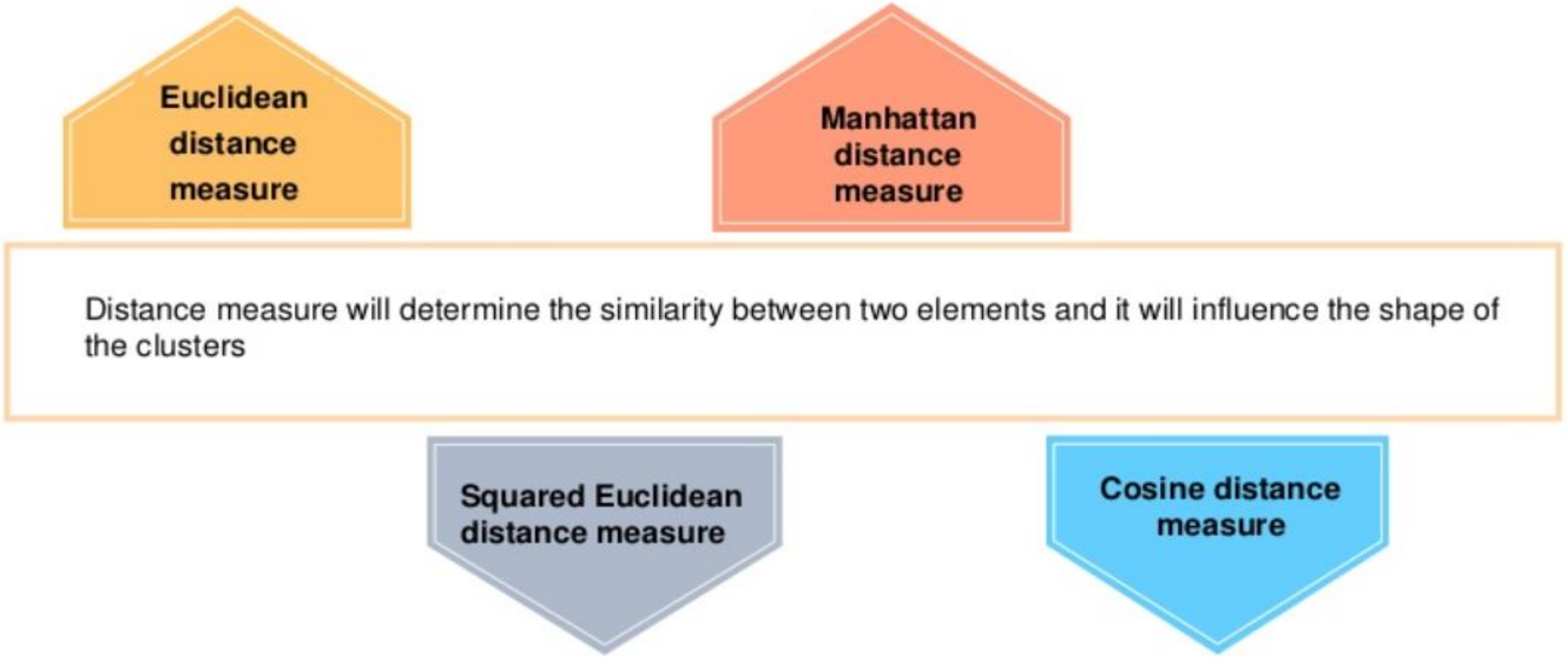
Activate Windows

Clustering: k-means

Activate Windows

Clustering: k-means

Distance Measure



Activate Windows
Go to Settings to activate Windows

Clustering: k-means

Euclidean Distance Measure

01 **Euclidean distance measure**

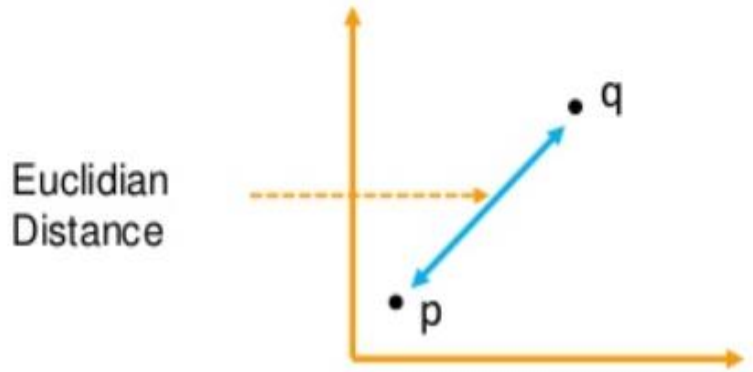
02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

- The Euclidean distance is the "ordinary" straight line
- It is the distance between two points in Euclidean space

$$d=\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$$



Activate Windows

Clustering: k-means

Squared Euclidean Distance Measure

01 **Euclidean distance measure**

02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The **Euclidean squared distance** metric uses the same equation as the **Euclidean distance** metric, but does not take the square root.

$$\mathbf{d} = \sum_{i=1}^{n} (q_i - p_i)^2$$

Activate Windows

Clustering: k-means

Manhattan Distance Measure

01 **Euclidean distance measure**

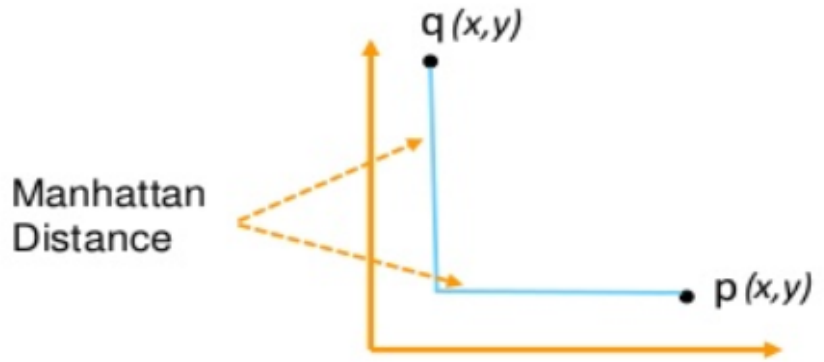
02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The Manhattan distance is the simple sum of the horizontal and vertical components or the distance between two points measured along axes at right angles

$$\mathbf{d}= \sum_{i=1}^{n} | q_x - p_x | + | q_x - p_y |$$



Activate Windows
Go to Settings to activate Windows.

Clustering: k-means

Cosine Distance Measure

01 **Euclidean distance measure**

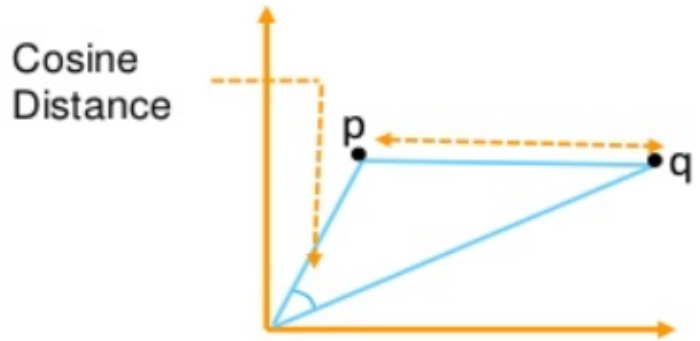
02 Squared euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The cosine distance similarity measures the angle between the two vectors

$$d = \frac{\sum_{i=0}^{n-1} q_{i-} p_x}{\sum_{i=0}^{n-1} (q_i)^2 \times \sum_{i=0}^{n-1} (p_i)^2}$$



Activate Windows
Go to Settings to activate Windows.

Clustering: k-means

Aspects of clustering

- A clustering algorithm such as
  - Partitional clustering eg, kmeans
  - Hierarchical clustering eg, AHC
  - Mixture of Gaussians
- A distance or similarity function
  - such as Euclidean, Minkowski, cosine
- Clustering quality
  - Inter-clusters distance ⇒ maximized
  - Intra-clusters distance ⇒ minimized

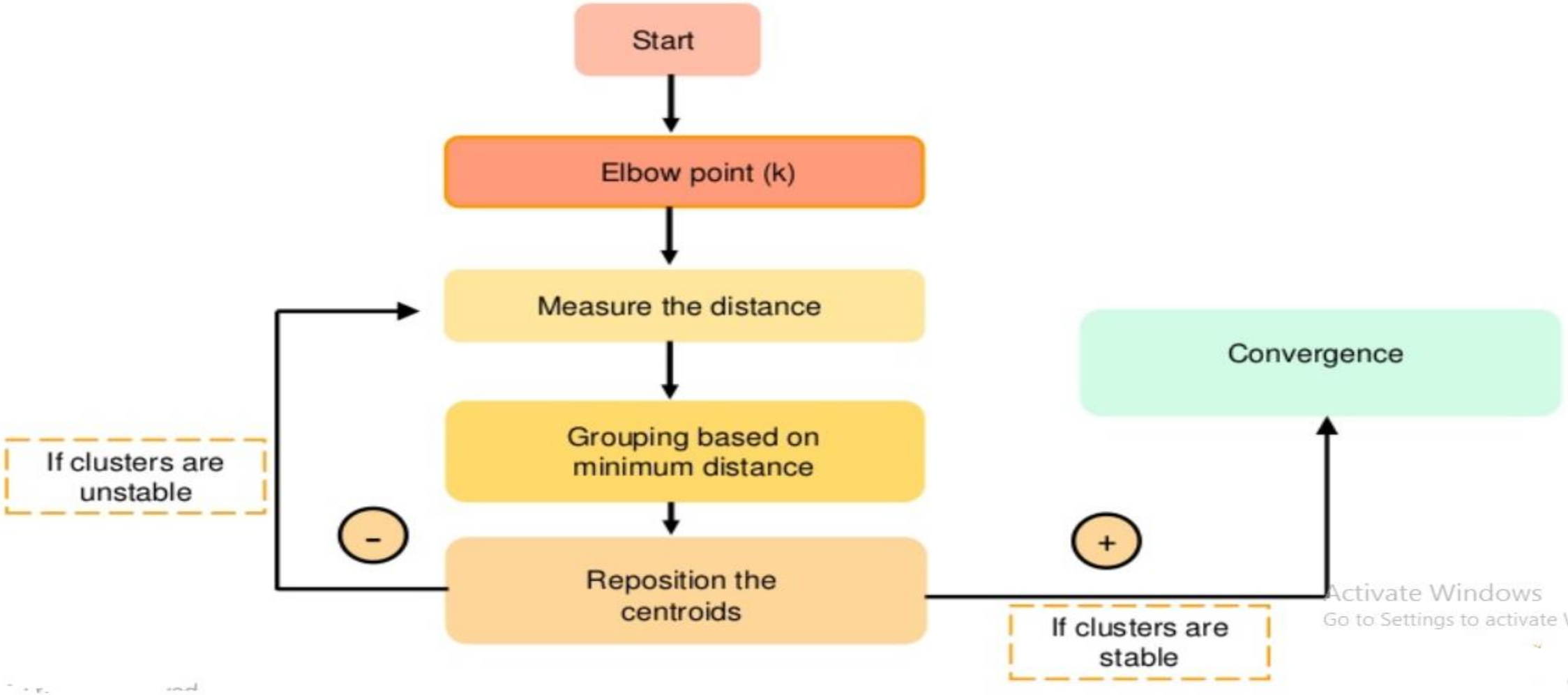
The quality of a clustering result depends on the algorithm, the distance function, and the application.

Clustering: k-means

Activate Windows
Go to Settings to activate Wi

Clustering: k-means

How does K-Means clustering work?



Activate Windows
Go to Settings to activate Wi

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

- Let's say, you have a dataset for a **Grocery shop**

- Now, the important question is, "***how would you choose the optimum number of clusters?***"

Activate Windo
Go to Settings to ac

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

- The best way to do this is by **Elbow method**
- The idea of the elbow method is to run K-Means clustering on the dataset where 'k' is referred as number of clusters
- Within sum of squares (WSS) is defined as the sum of the squared distance between each member of the cluster and its centroid

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

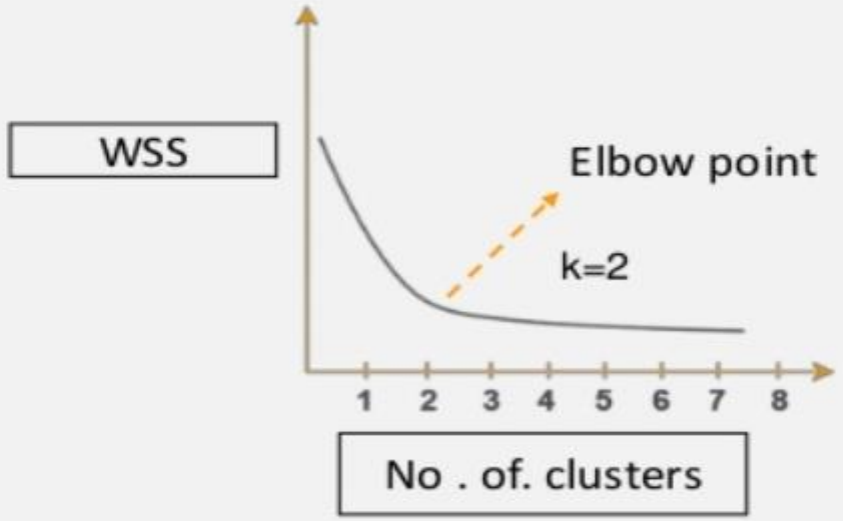
Where x_i = data point and c_i = closest point to centroid

Activate Windows
Go to Settings to activate Windows

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence



- Now, we draw a curve between ***WSS*** (within sum of squares) and the ***number of clusters***
- Here, we can see a very slow change in the value of WSS after k=2, so you should take that elbow point value as the final number of clusters

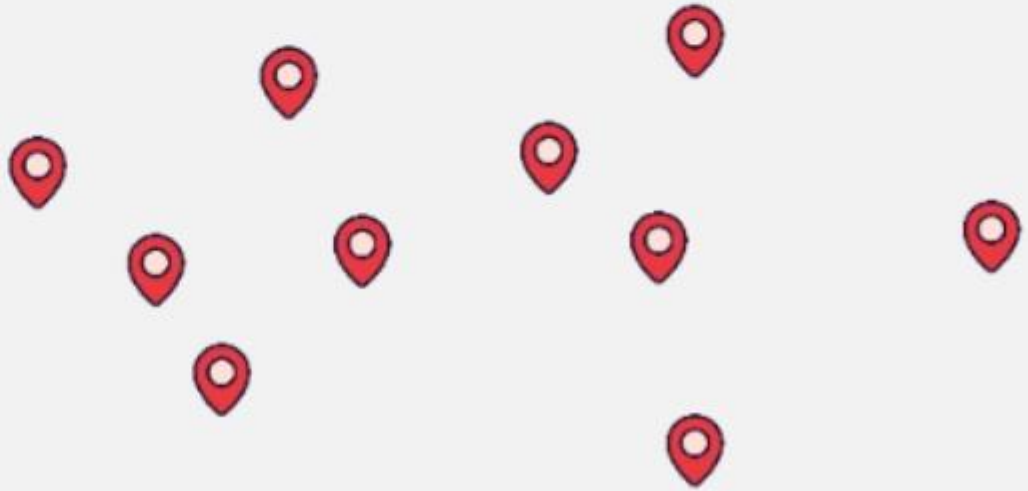
Activate Windows
Go to Settings to activate Windo

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 1: The given data points below are assumed as **delivery points**



Activate Windows
Go to Settings to activate Win

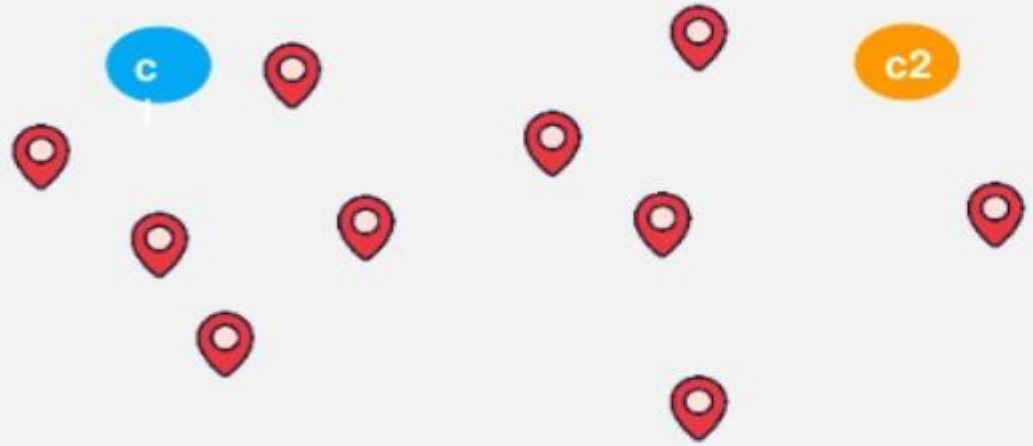
Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 2: We can randomly initialize two points called the cluster centroids,

Euclidean distance is a distance measure used to find out which data point is closest to our centroids



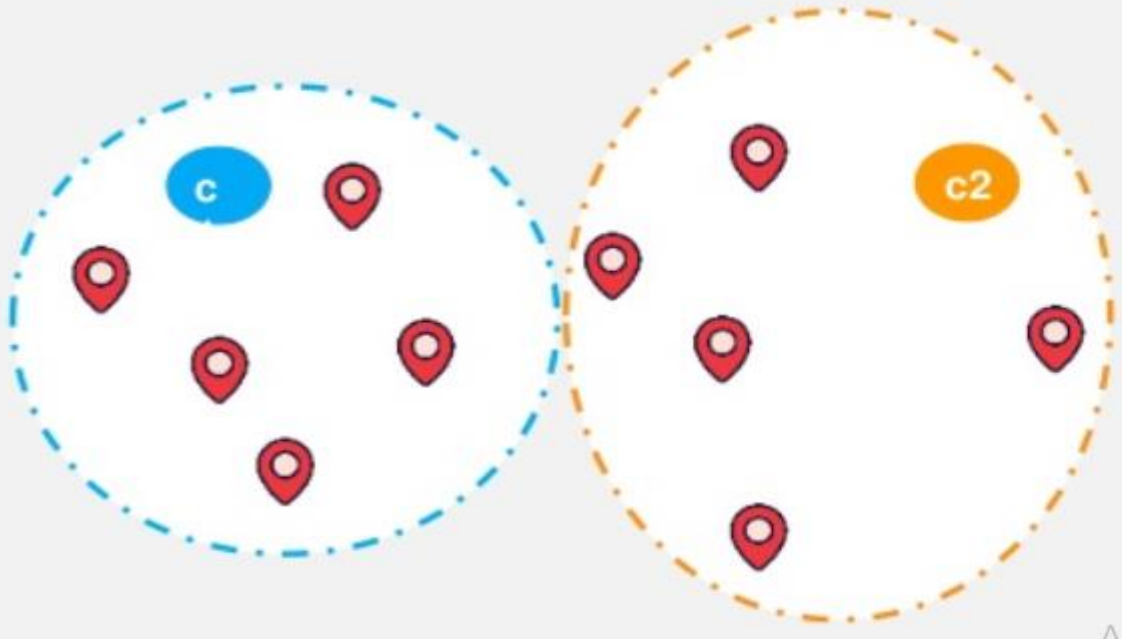
Activate Windows
Go to Settings to activate Windo

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 3: Based upon the distance from c1 and c2 centroids, the data points will group itself into clusters



Activate Windows
Go to Settings to activate Wi

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 4: Compute the centroid of data points inside blue cluster

Step 5: Reposition the centroid of the blue cluster to the new centroid

Activate Window
Go to Settings to activ

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

Step 6: Now, compute the centroid of data points inside orange cluster

Step 7: Reposition the centroid of the orange cluster to the new centroid

Activate Windows
Go to Settings to activate W

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence

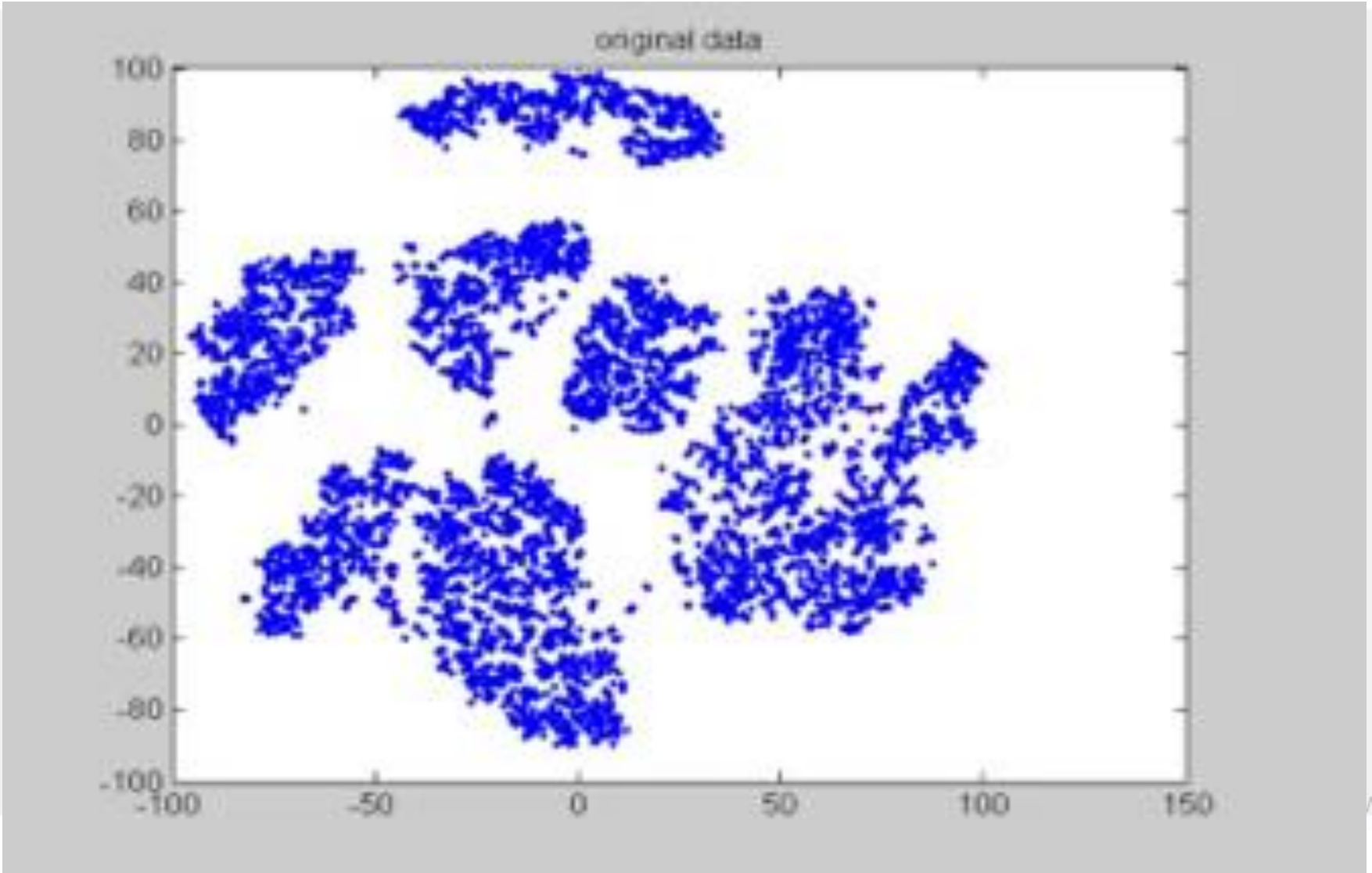
Step 8: Once the clusters become static, K-Means clustering algorithm is said to be converged

Activate Windows
Go to Settings to activate Wind

Clustering: k-means

How does K-Means clustering work?

- Elbow point
- Measure the distance
- Grouping
- Reposition the centroids
- Convergence



Clustering: k-means

Activate W
Go to Settings

Clustering: k-means

K-Means Clustering Algorithm

Assuming we have inputs x1,x2,x3,…, and value of **K,**

Step 1 : Pick K random points as cluster centers called centroids

Step 2 : Assign each xi to nearest cluster by calculating its distance to each centroid

Step 3 : Find new cluster center by taking the average of the assigned points

Step 4 : Repeat Step 2 and 3 until none of the cluster assignments change

Activate Windows

Clustering: k-means

K-Means Clustering Algorithm

Step 1 :

We randomly pick **K** cluster centers (centroids). Let's assume these are c1,c2,…,ckc1,c2,…,ck, and we can say that;

$$C = c_1, c_2, \ldots . c_k$$

C is the set of all centroids.

Step 2:

In this step, we assign each data point to closest center, this is done by calculating Euclidean distance

$$\arg\min_{c_i \in C} dist(\quad, x_i)^2$$

Where **dist()** is the Euclidean distance.

Activate Windows
Go to Settings to activate Wind

Clustering: k-means

K-Means Clustering Algorithm

Step 3:

In this step, we find the new centroid by taking the average of all the points assigned to that cluster.

$$c_i = \frac{1}{|Si|} \sum_{xi \in Si} x_i$$

S_i is the set of all points assigned to the i th cluster

Step 4:

In this step, we repeat step 2 and 3 until none of the cluster assignments change

That means until our clusters remain stable, we repeat the algorithm

Activate Windows

Use Case: K-Means for Color Compression

```
# output
```

Conclusion

Congratulations!

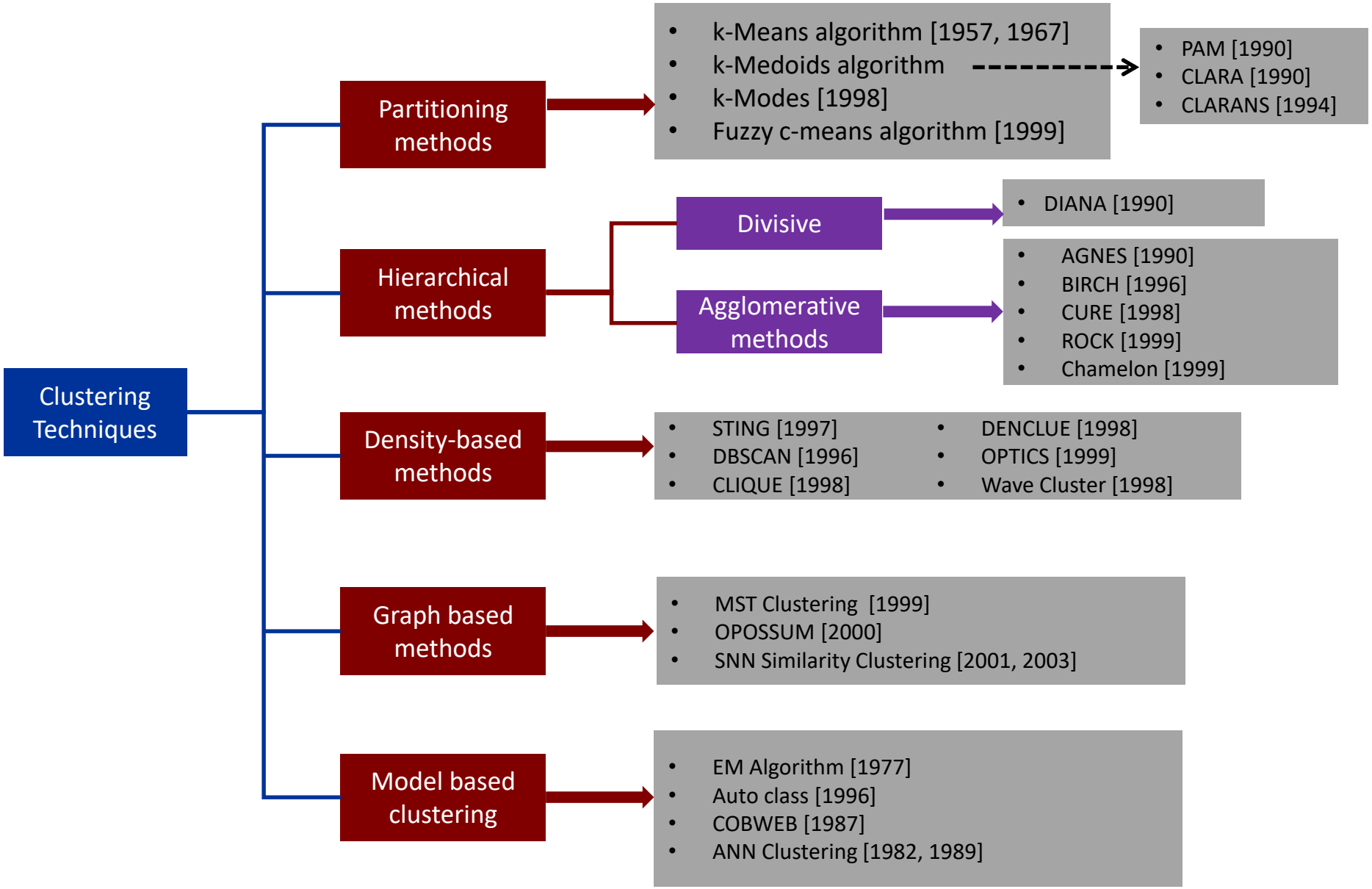
We have demonstrated K-Means in color compression.

The hands on example will help you to encounter any K-Means project in future.

Activate Windows
Go to Settings to activate Windows.

Clustering techniques

- Clustering has been studied extensively for more than 40 years and across many disciplines due to its broad applications.
- As a result, many clustering techniques have been reported in the literature.
- Let us categorize the clustering methods. In fact, it is difficult to provide a crisp categorization because many techniques overlap to each other in terms of clustering paradigms or features.
- It is not possible to cover all the techniques in this lecture series. We emphasize on major techniques belong to partitioning and hierarchical algorithms.


Clustering Techniques
Partitioning methods
k-Means algorithm [1957, 1967]
k-Medoids algorithm
k-Modes [1998]
Fuzzy c-means algorithm [1999]
PAM [1990]
CLARA [1990]
CLARANS [1994]
Hierarchical methods
Divisive
DIANA [1990]
Agglomerative methods
AGNES [1990]
BIRCH [1996]
CURE [1998]
ROCK [1999]
Chamelon [1999]
Density-based methods
STING [1997]
DBSCAN [1996]
CLIQUE [1998]
DENCLUE [1998]
OPTICS [1999]
Wave Cluster [1998]
Graph based methods
MST Clustering [1999]
OPOSSUM [2000]
SNN Similarity Clustering [2001, 2003]
Model based clustering
EM Algorithm [1977]
Auto class [1996]
COBWEB [1987]
ANN Clustering [1982, 1989]

k-Means Algorithm

- k-Means clustering algorithm proposed by J. Hartigan and M. A. Wong [1979].

- Given a set of *n* distinct objects, the k-Means clustering algorithm partitions the objects into *k* number of clusters such that intracluster similarity is high but the intercluster similarity is low.

- In this algorithm, user has to specify *k*, the number of clusters and consider the objects are defined with numeric attributes and thus using any one of the distance metric to demarcate the clusters.

k-Means Algorithm

The algorithm can be stated as follows.

- First it selects *k* number of objects at random from the set of n objects. These *k* objects are treated as the centroids or center of gravities of *k* clusters.

- For each of the remaining objects, it is assigned to one of the closest centroid. Thus, it forms a collection of objects assigned to each centroid and is called a cluster.

- Next, the centroid of each cluster is then updated (by calculating the mean values of attributes of each object).

- The assignment and update procedure is until it reaches some stopping criteria (such as, number of iteration, centroids remain unchanged or no assignment, etc.)

k-Means Algorithm

Algorithm 16.1: k-Means clustering

Input: D is a dataset containing *n* objects, *k* is the number of cluster

Output: A set of *k* clusters

Steps:

1. Randomly choose *k* objects from D as the initial cluster centroids.

2. **For** each of the objects in D **do**
   - Compute distance between the current objects and *k* cluster centroids
   - Assign the current object to that cluster to which it is closest.

3. Compute the "cluster centers" of each cluster. These become the new cluster centroids.

4. Repeat step 2-3 until the convergence criterion is satisfied

5. Stop

k-Means Algorithm

Note:

1) Objects are defined in terms of set of attributes. $A = \{A_1, A_2, \ldots\ldots, A_m\}$ where each A_i is continuous data type.

2) Distance computation: Any distance such as L_1, L_2, L_3 or cosine similarity.

3) Minimum distance is the measure of closeness between an object and centroid.

4) Mean Calculation: It is the mean value of each attribute values of all objects.

5) Convergence criteria: Any one of the following are termination condition of the algorithm.
   - Number of maximum iteration permissible.
   - No change of centroid values in any cluster.
   - Zero (or no significant) movement(s) of object from one cluster to another.
   - Cluster quality reaches to a certain level of acceptance.

Illustration of k-Means clustering algorithms

Table 16.1: 16 objects with two attributes A_1 and A_2.

A_1	A_2
6.8	12.6
0.8	9.8
1.2	11.6
2.8	9.6
3.8	9.9
4.4	6.5
4.8	1.1
6.0	19.9
6.2	18.5
7.6	17.4
7.8	12.2
6.6	7.7
8.2	4.5
8.4	6.9
9.0	3.4
9.6	11.1

Fig 16.1: Plotting data of Table 16.1

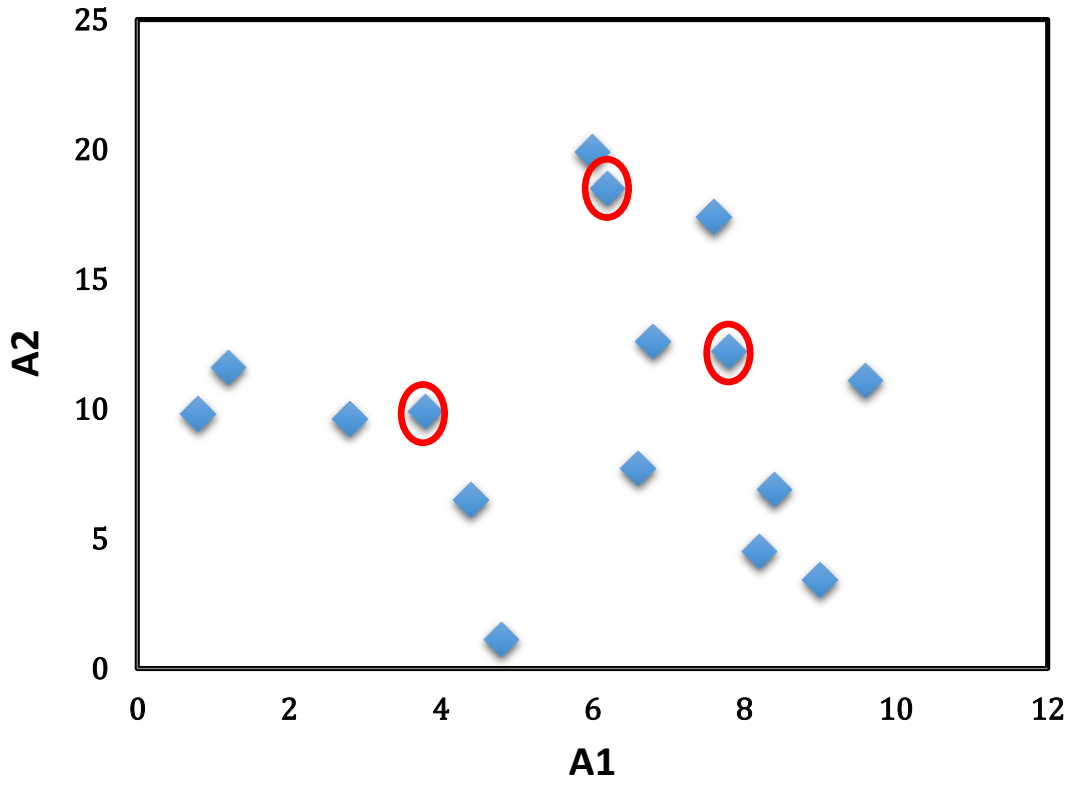


Illustration of k-Means clustering algorithms

- Suppose, k=3. Three objects are chosen at random shown as circled (see Fig 16.1). These three centroids are shown below.

Initial Centroids chosen randomly

Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

- Let us consider the Euclidean distance measure (L_2 Norm) as the distance measurement in our illustration.
- Let d_1, d_2 and d_3 denote the distance from an object to c_1, c_2 and c_3 respectively. The distance calculations are shown in Table 16.2.
- Assignment of each object to the respective centroid is shown in the right-most column and the clustering so obtained is shown in Fig 16.2.

Illustration of k-Means clustering algorithms

Table 16.2: Distance calculation

A_1	A_2	d_1	d_2	d_3	cluster
6.8	12.6	4.0	1.1	5.9	2
0.8	9.8	3.0	7.4	10.2	1
1.2	11.6	3.1	6.6	8.5	1
2.8	9.6	1.0	5.6	9.5	1
3.8	9.9	0.0	4.6	8.9	1
4.4	6.5	3.5	6.6	12.1	1
4.8	1.1	8.9	11.5	17.5	1
6.0	19.9	10.2	7.9	1.4	3
6.2	18.5	8.9	6.5	0.0	3
7.6	17.4	8.4	5.2	1.8	3
7.8	12.2	4.6	0.0	6.5	2
6.6	7.7	3.6	4.7	10.8	1
8.2	4.5	7.0	7.7	14.1	1
8.4	6.9	5.5	5.3	11.8	2
9.0	3.4	8.3	8.9	15.4	1
9.6	11.1	5.9	2.1	8.1	2

Fig 16.2: Initial cluster with respect to Table 16.2

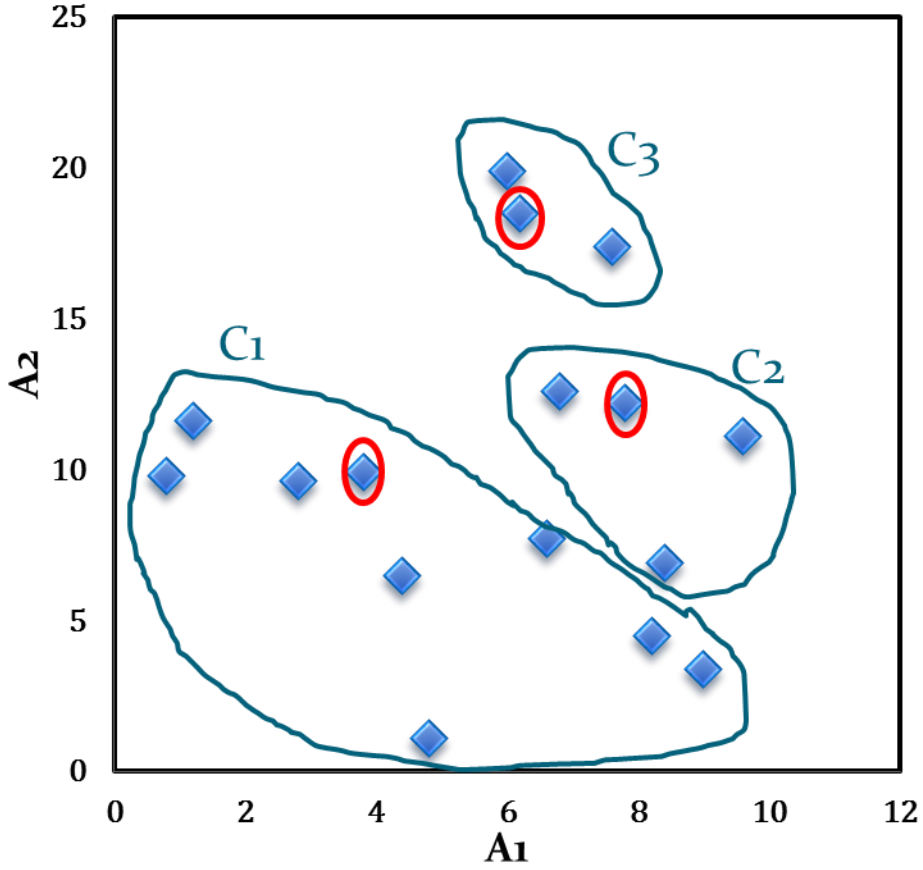


Illustration of k-Means clustering algorithms

The calculation new centroids of the three cluster using the mean of attribute values of A_1 and A_2 is shown in the Table below. The cluster with new centroids are shown in Fig 16.3.

Calculation of new centroids

New Centroid	Objects	
	A1	A2
c_1	4.6	7.1
c_2	8.2	10.7
c_3	6.6	18.6

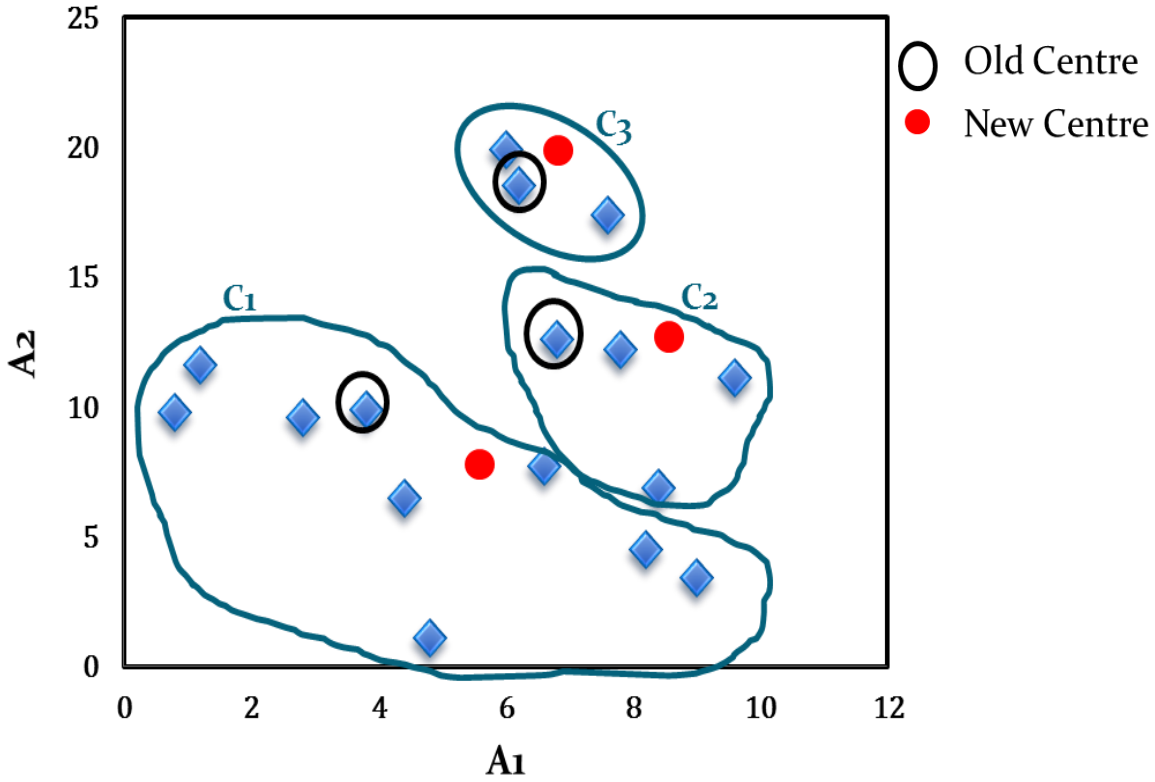


Fig 16.3: Initial cluster with new centroids

Illustration of k-Means clustering algorithms

We next reassign the 16 objects to three clusters by determining which centroid is closest to each one. This gives the revised set of clusters shown in Fig 16.4.

Note that point p moves from cluster C_2 to cluster C_1.

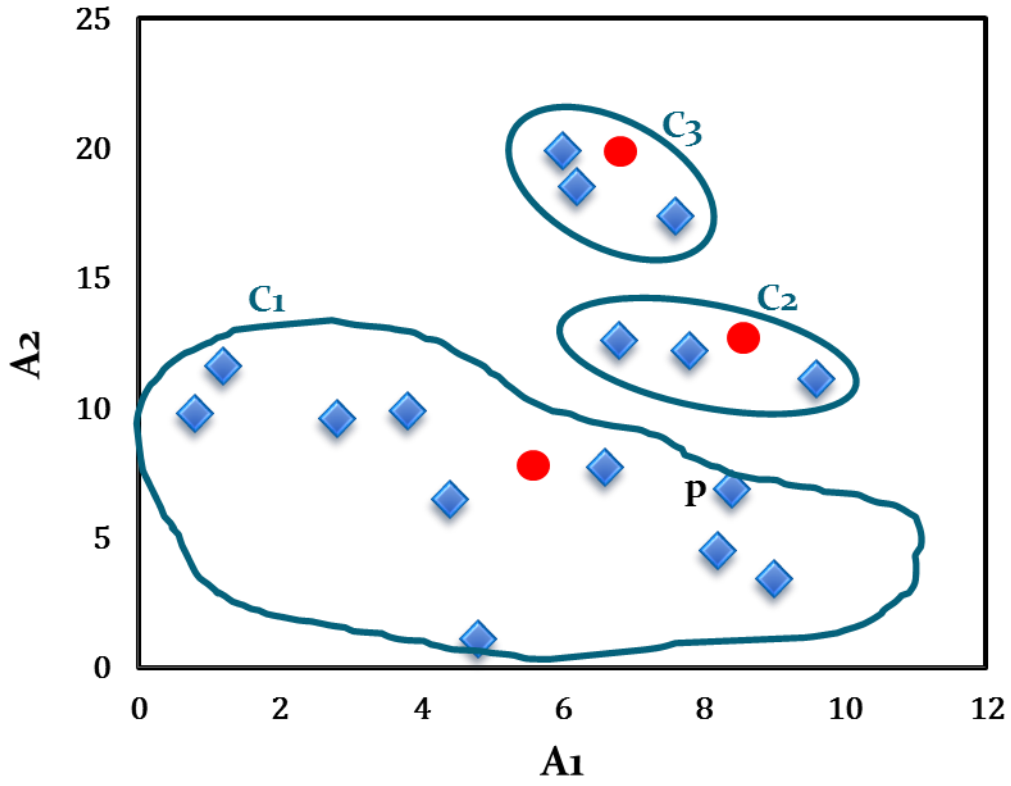


Fig 16.4: Cluster after first iteration

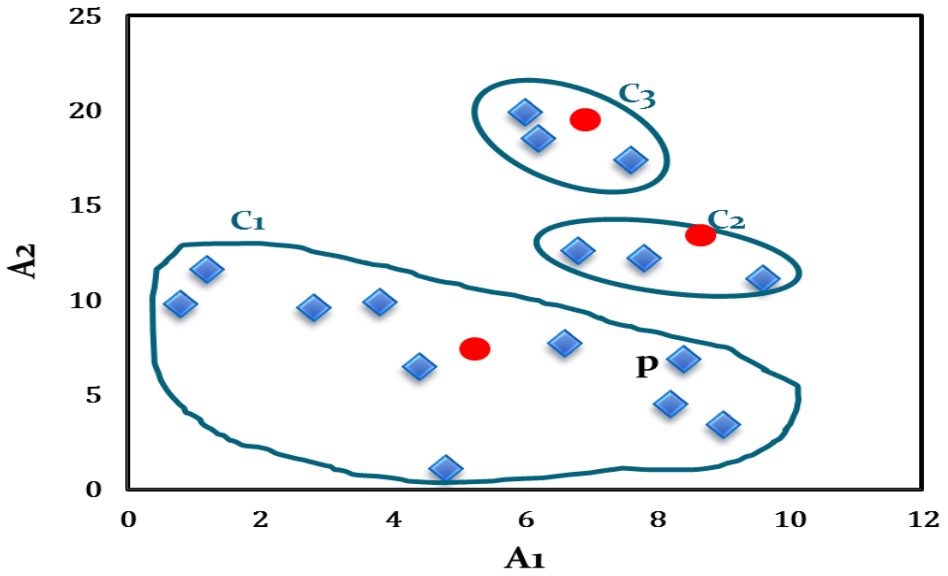
Illustration of k-Means clustering algorithms

- The newly obtained centroids after second iteration are given in the table below. Note that the centroid c_3 remains unchanged, where c_2 and c_1 changed a little.
- With respect to newly obtained cluster centres, 16 points are reassigned again. These are the same clusters as before. Hence, their centroids also remain unchanged.
- Considering this as the termination criteria, the k-means algorithm stops here. Hence, the final cluster in Fig 16.5 is same as Fig 16.4.

Cluster centres after second iteration

Centroid	Revised Centroids	
	A1	A2
c_1	5.0	7.1
c_2	8.1	12.0
c_3	6.6	18.6

Fig 16.5: Cluster after Second iteration



Comments on k-Means algorithm

Let us analyse the k-Means algorithm and discuss the pros and cons of the algorithm.

We shall refer to the following notations in our discussion.

- Notations:
  - x : an object under clustering
  - n : number of objects under clustering
  - $\boldsymbol{C}_i$: the *i-th* cluster
  - c_i : the centroid of cluster $\boldsymbol{C}_i$
  - n_i : number of objects in the cluster $\boldsymbol{C}_i$
  - c : denotes the centroid of all objects
  - k : number of clusters

Comments on k-Means algorithm

1. **Value of k:**
   - The k-means algorithm produces only one set of clusters, for which, user must specify the desired number, k of clusters.
   - In fact, k should be the best guess on the number of clusters present in the given data. Choosing the best value of k for a given dataset is, therefore, an issue.
   - We may not have an idea about the possible number of clusters for high dimensional data, and for data that are not scatter-plotted.
   - Further, possible number of clusters is hidden or ambiguous in im, audio, video and multimedia clustering applications etc.
   - There is no principled way to know what the value of k ought to be. We may try with successive value of k starting with 2.
   - The process is stopped when two consecutive k values produce more-or-less identical results (with respect to some cluster quality estimation).
   - Normally $k \ll n$ and there is heuristic to follow $k \approx \sqrt{n}$.

Comments on k-Means algorithm

Example 16.1: k versus cluster quality

- Usually, there is some objective function to be met as a goal of clustering. One such objective function is sum-square-error denoted by SSE and defined as

$$SSE = \sum_{i=1}^{k} \sum_{x \in \boldsymbol{C}_i} (x - c_i)^2$$

- Here, $x - c_i$ denotes the error, if x is in cluster $\boldsymbol{C}_i$ with cluster centroid c_i.

- Usually, this error is measured as distance norms like L_1, L_2, L_3 or Cosine similarity, etc.

Comments on k-Means algorithm

Example 16.1: k versus cluster quality

- With reference to an arbitrary experiment, suppose the following results are obtained.

k	SSE
1	62.8
2	12.3
3	9.4
4	9.3
5	9.2
6	9.1
7	9.05
8	9.0

- With respect to this observation, we can choose the value of $k \approx 3$, as with this smallest value of k it gives reasonably good result.
- Note: If $k = n$, then SSE=0; However, the cluster is useless! This is another example of overfitting.

Comments on k-Means algorithm

2. Choosing initial centroids:

- Another requirement in the k-Means algorithm to choose initial cluster centroid for each *k* would be clusters.
- It is observed that the k-Means algorithm terminate whatever be the initial choice of the cluster centroids.
- It is also observed that initial choice influences the ultimate cluster quality. In other words, the result may be trapped into local optima, if initial centroids are chosen properly.
- One technique that is usually followed to avoid the above problem is to choose initial centroids in multiple runs, each with a different set of randomly chosen initial centroids, and then select the best cluster (with respect to some quality measurement criterion, e.g. SSE).
- However, this strategy suffers from the combinational explosion problem due to the number of all possible solutions.

Comments on k-Means algorithm

2. Choosing initial centroids:

- A detail calculation reveals that there are $c(n,k)$ possible combinations to examine the search of global optima.

$$c(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-i}\binom{k}{i}(i)^n$$

- For example, there are $o(10^{10})$ different ways to cluster 20 items into 4 clusters!

- Thus, the strategy having its own limitation is practical only if
  1) The sample is negatively small (~100-1000), and
  2) *k* is relatively small compared to *n* (i.e.. $k \ll n$).

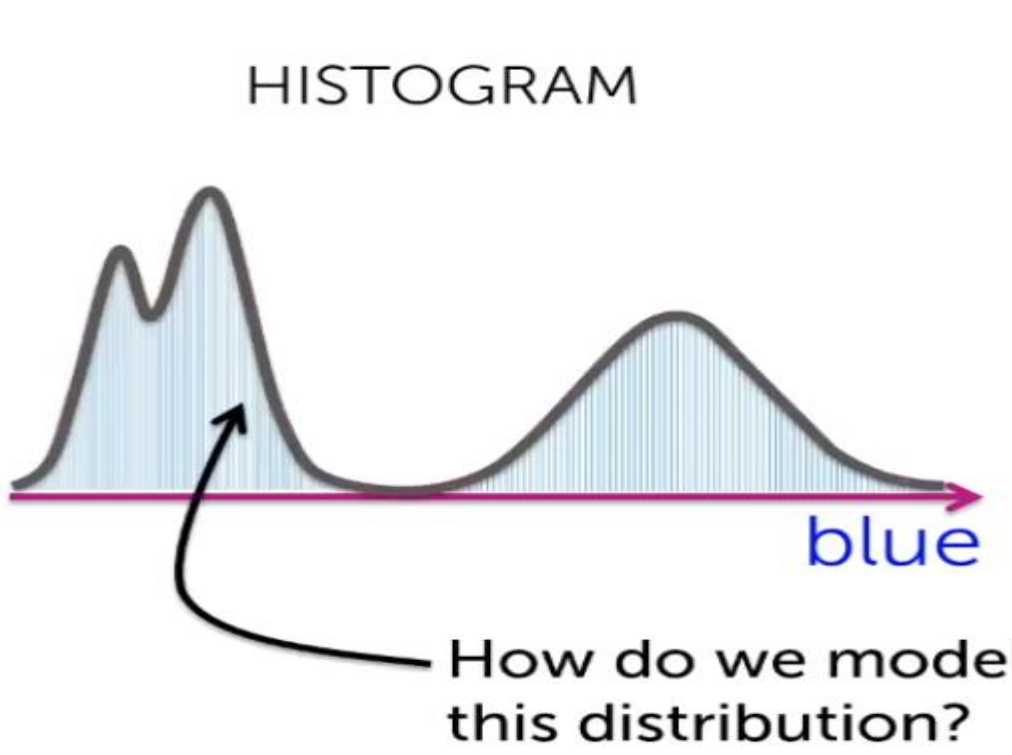
Introduction to Gaussian Mixture Models (GMMs)

Example 1:

Jumble of unlabeled images

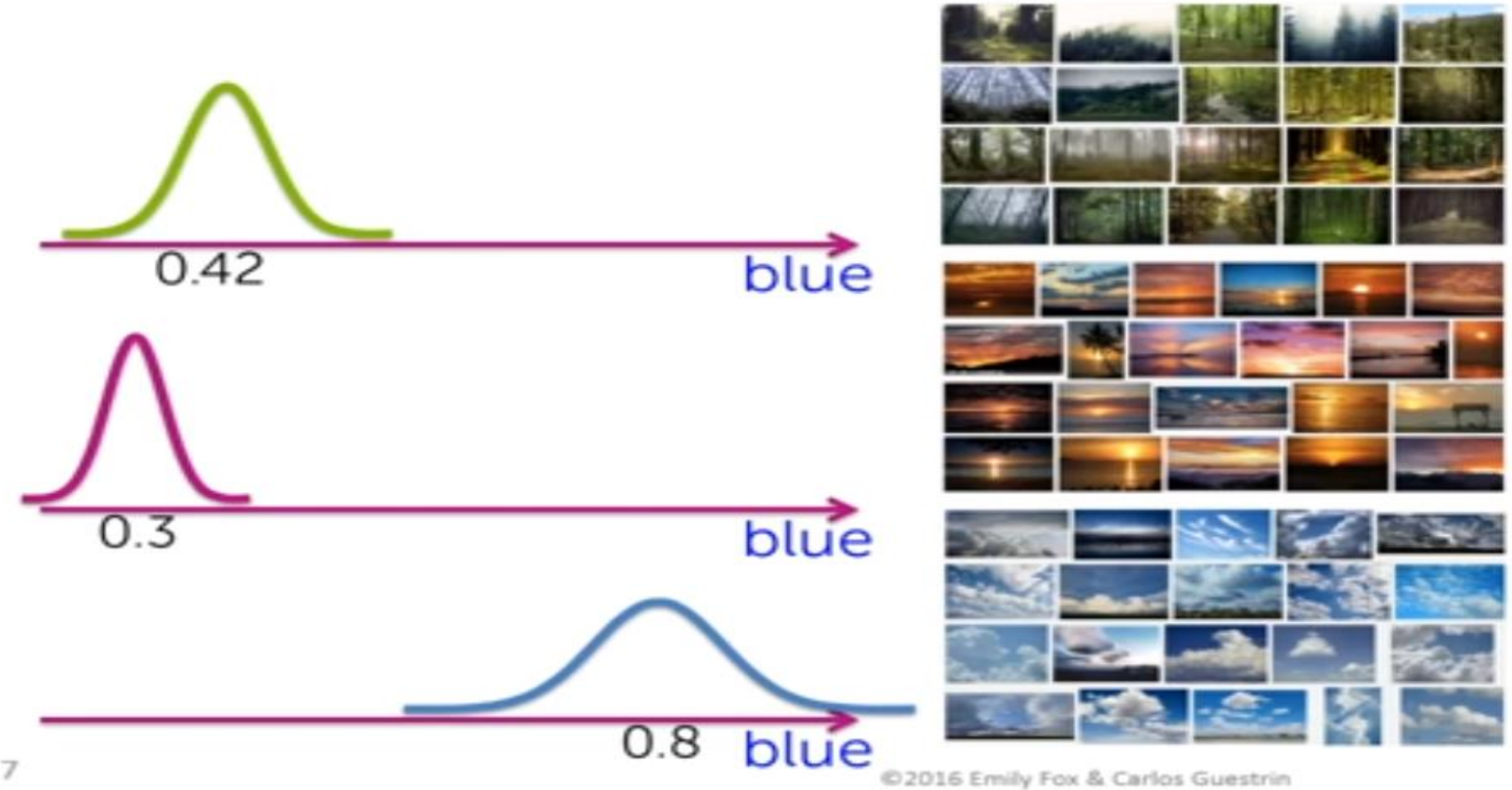
Introduction to Gaussian Mixture Models (GMMs)

Jumble of unlabeled images



Introduction to Gaussian Mixture Models (GMMs)

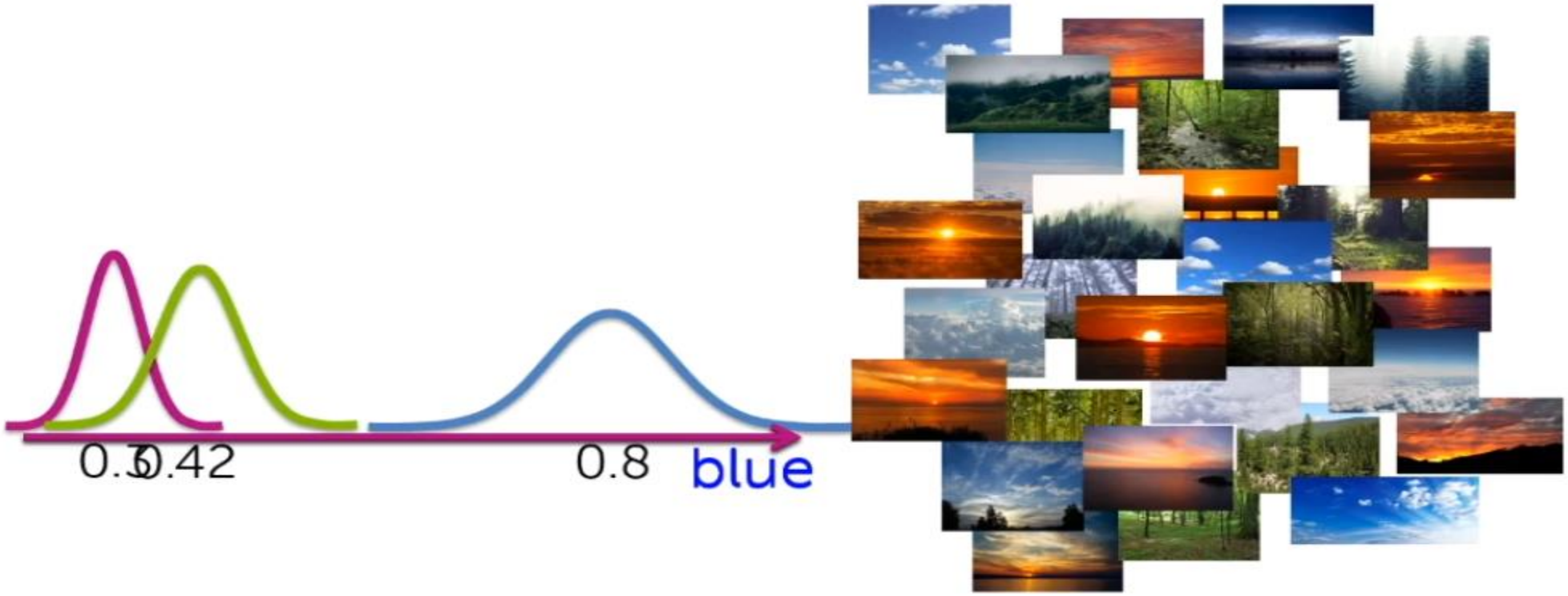
Model as Gaussian per category/cluster



©2016 Emily Fox & Carlos Guestrin

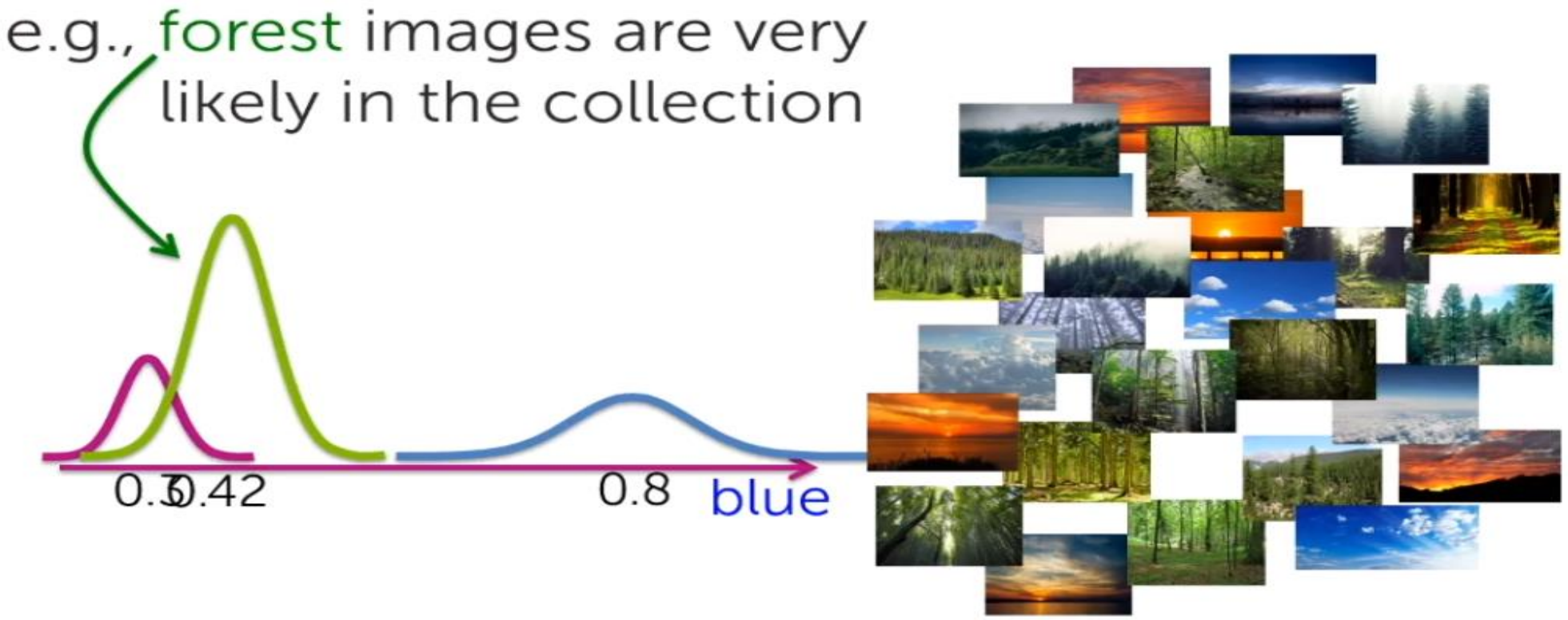
Introduction to Gaussian Mixture Models (GMMs)

Model of jumble of unlabeled images



Introduction to Gaussian Mixture Models (GMMs)

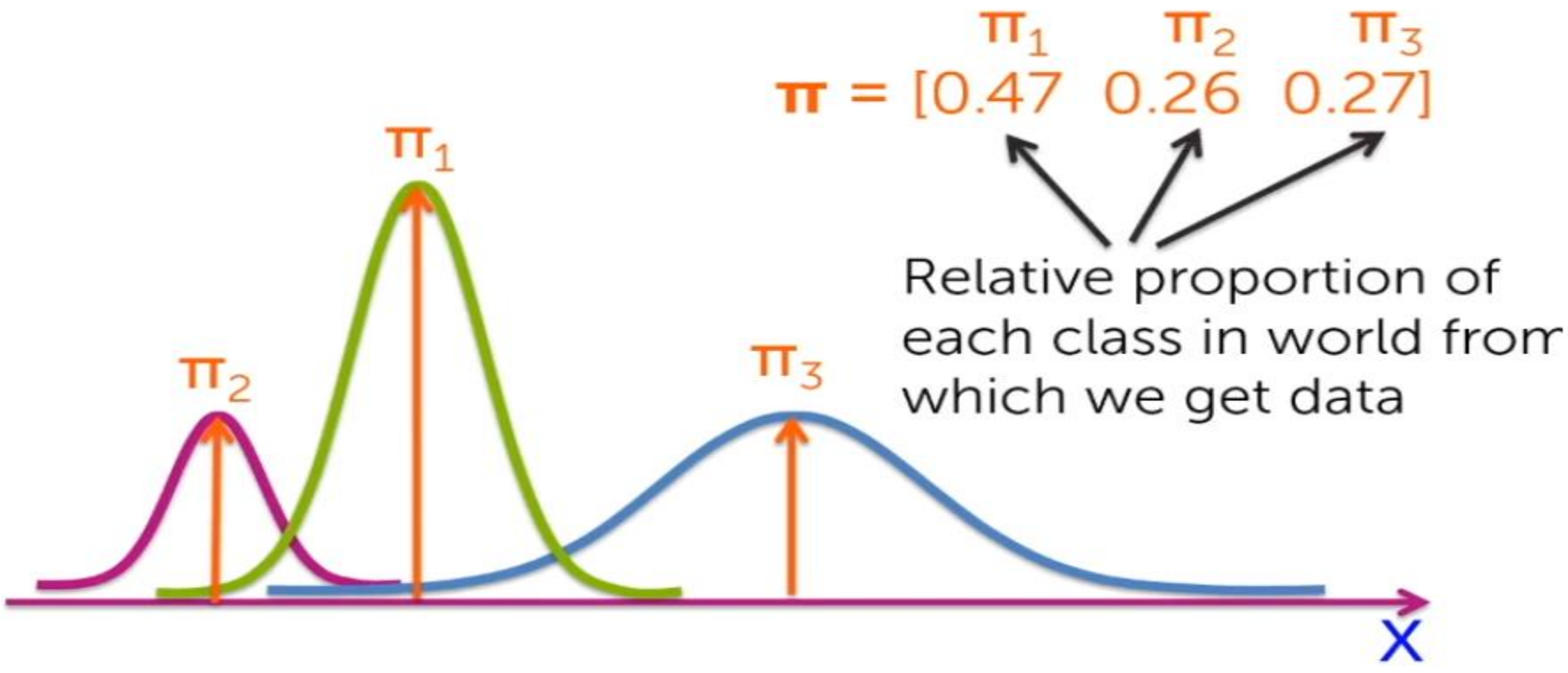
What if image types not equally represented?



Introduction to Gaussian Mixture Models (GMMs)

Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

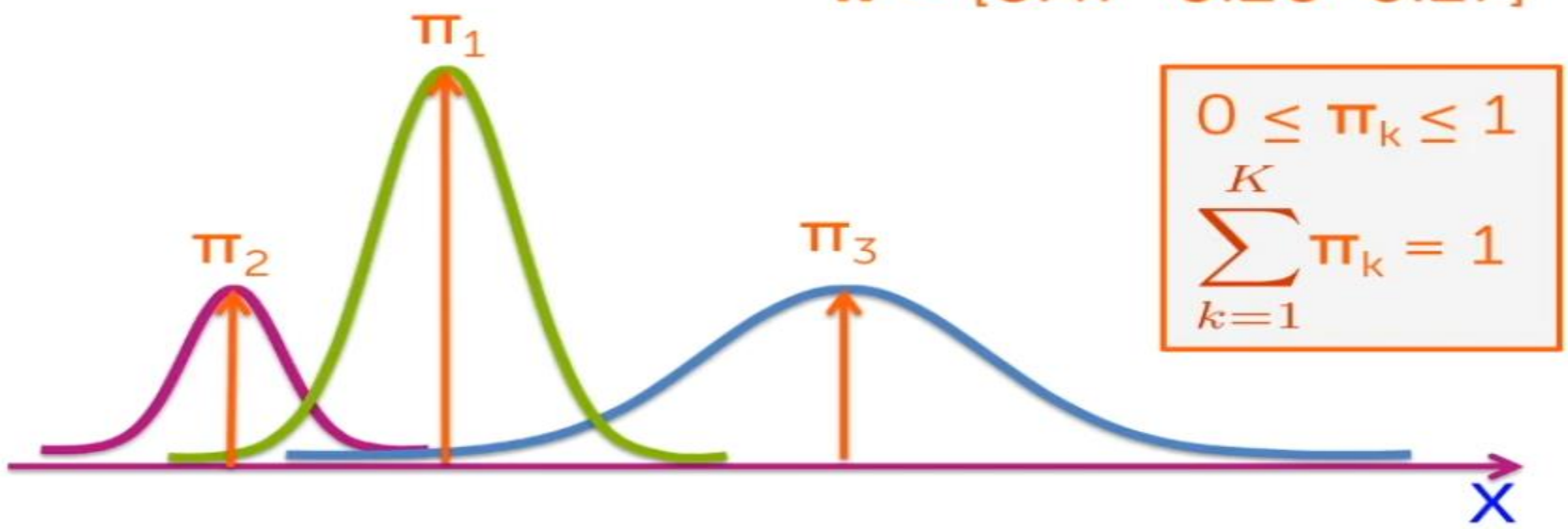


Introduction to Gaussian Mixture Models (GMMs)

Combination of weighted Gaussians

Associate a weight π_k with each Gaussian component

$$\boldsymbol{\pi} = [\pi_1 \quad \pi_2 \quad \pi_3] = [0.47 \quad 0.26 \quad 0.27]$$



$$0 \leq \pi_k \leq 1$$

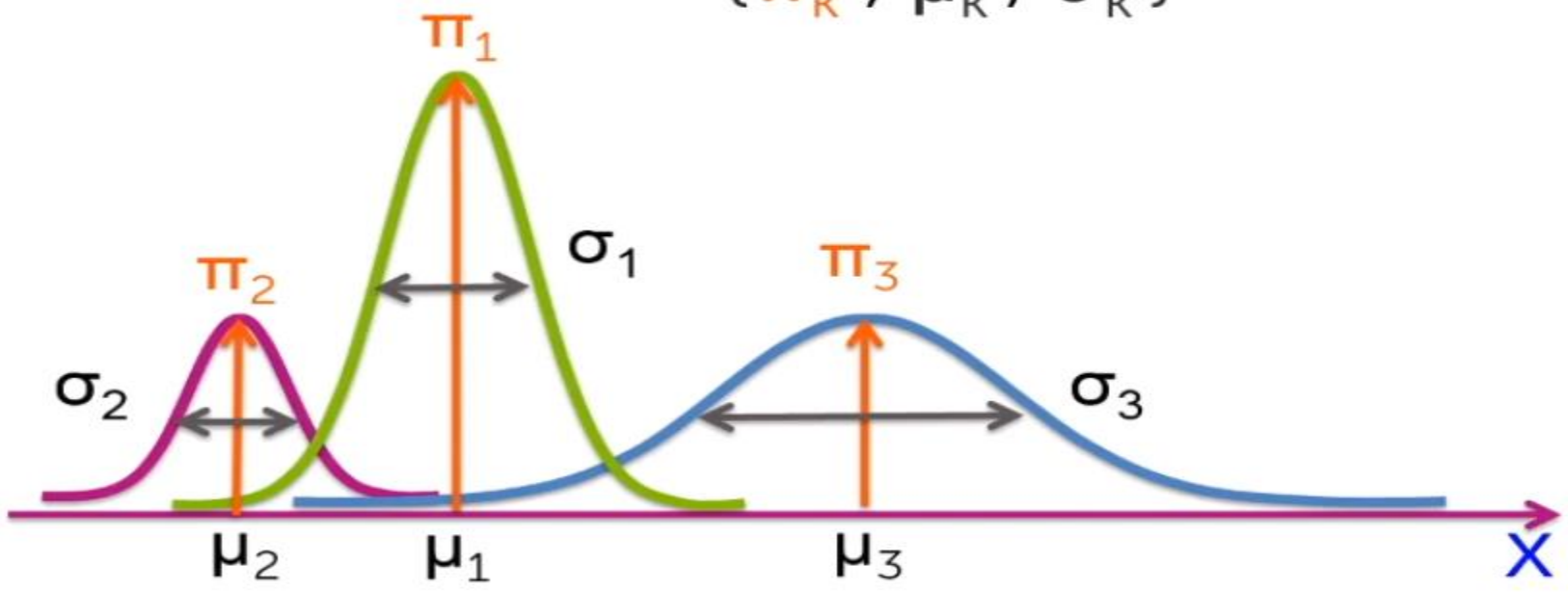
$$\sum_{k=1}^{K} \pi_k = 1$$

Introduction to Gaussian Mixture Models (GMMs)

Mixture of Gaussians (1D)

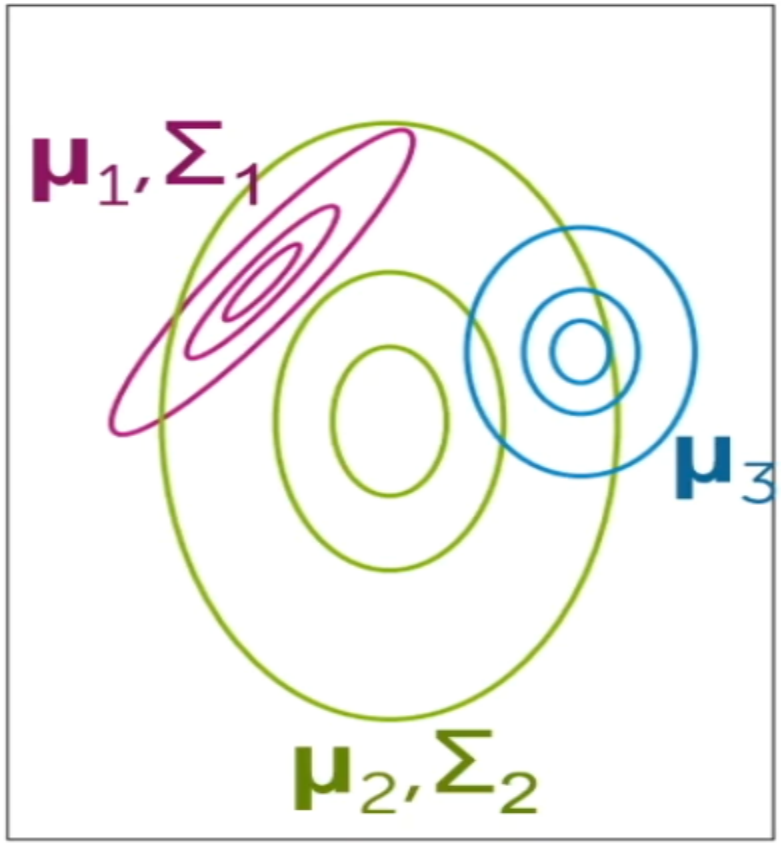
Each mixture component represents a unique cluster specified by:

$$\{\pi_k , \mu_k , \sigma_k^2\}$$



Introduction to Gaussian Mixture Models (GMMs)

Mixture of Gaussians (general)

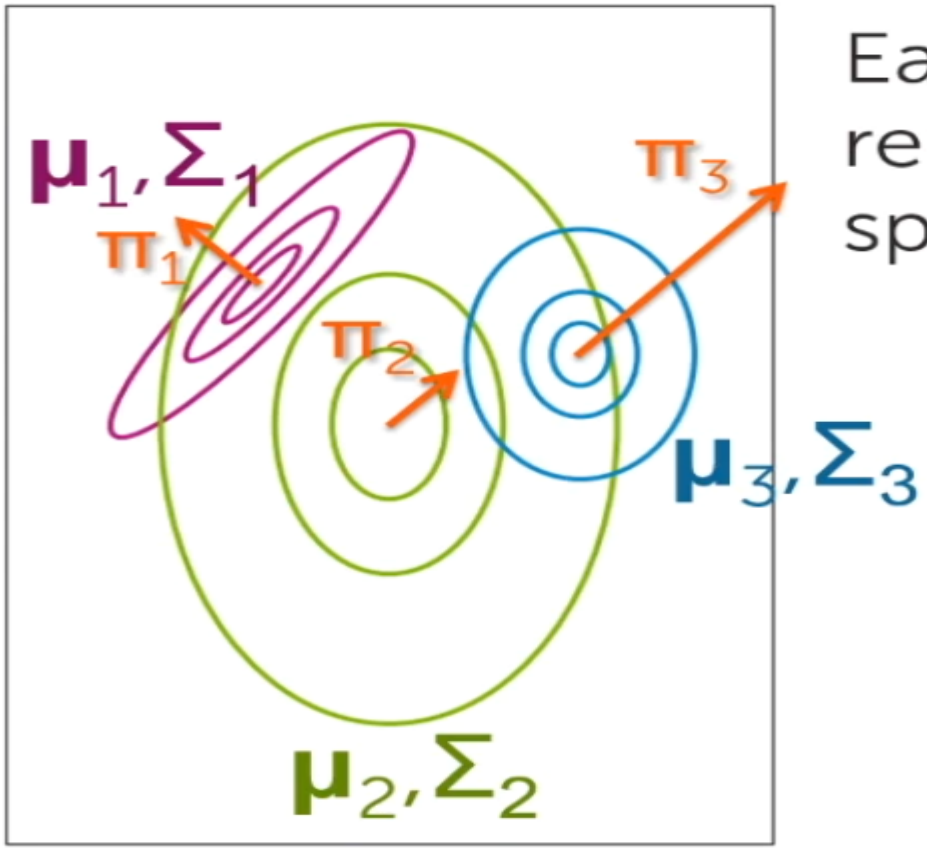


Each mixture component represents a unique cluster specified by:

$$\{\pi_k , \mu_k , \Sigma_k\}$$

Introduction to Gaussian Mixture Models (GMMs)

Mixture of Gaussians (general)

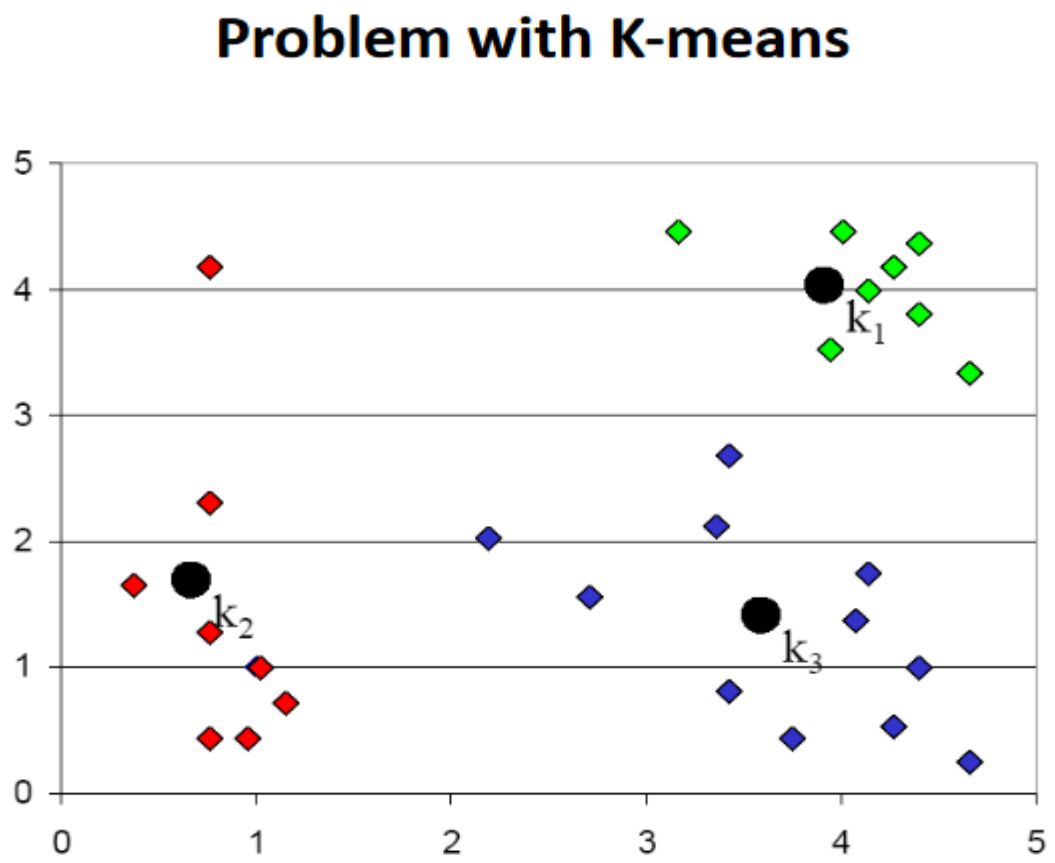


Each mixture component represents a unique cluster specified by:

$$\{\pi_k , \mu_k , \Sigma_k\}$$

Introduction to Gaussian Mixture Models (GMMs)

Example 2:

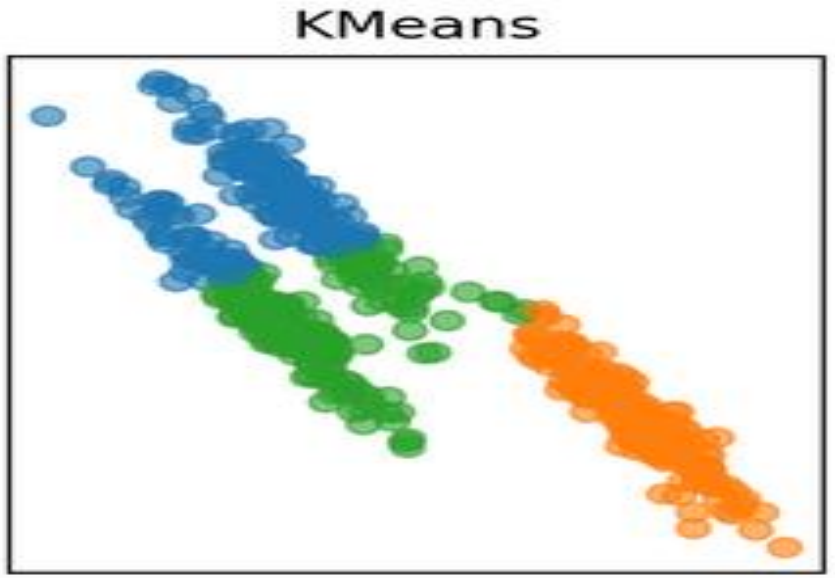


Introduction to Gaussian Mixture Models (GMMs)

- **Drawbacks of k-means Clustering**
- The k-means clustering concept is simple to understand, relatively easy to implement, and can be applied in quite a number of use cases.
- But there are certain drawbacks and limitations that we need to be aware of.
- Let's take the same income-expenditure example we saw above.
- The k-means algorithm seems to be working pretty well, right? Hold on – if you look closely, you will notice that all the clusters created have a circular shape.
- This is because the centroids of the clusters are updated iteratively using the mean value.

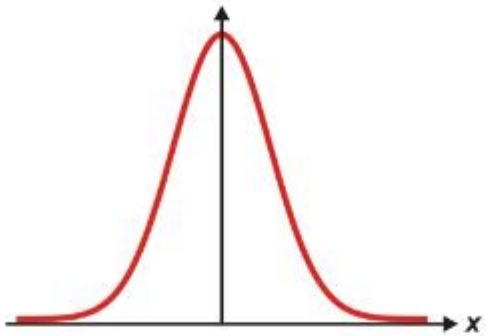
Introduction to Gaussian Mixture Models (GMMs)

- **Drawbacks of k-means Clustering**
- Now, consider the following example where the distribution of points is *not* in a circular form.
- What do you think will happen if we use k-means clustering on this data? It would still attempt to group the data points in a circular fashion.
- That's not great! k-means fails to identify the right clusters:



Introduction to Gaussian Mixture Models (GMMs)

Introduction



Carl Friedrich Gauss invented the normal distribution in 1809 as a way to rationalize the method of least squares.

Acti
Go t

Introduction to Gaussian Mixture Models (GMMs)

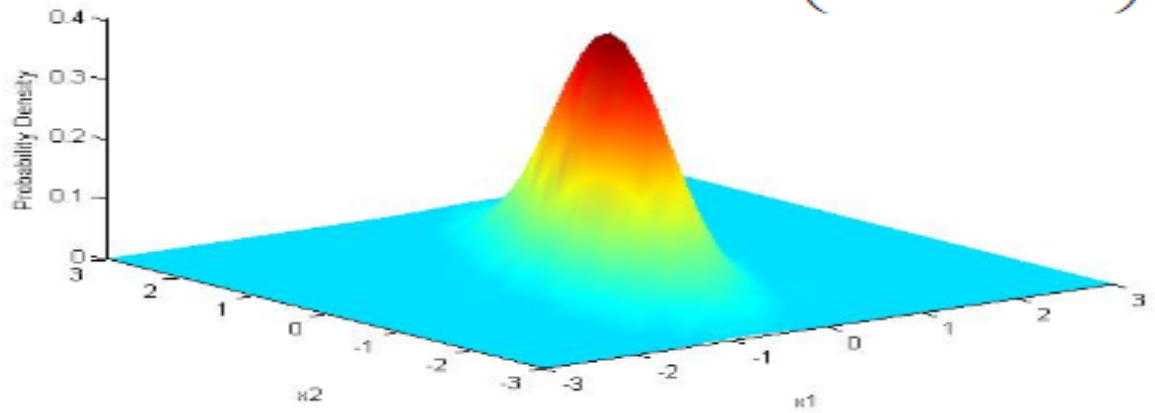
- **Gaussian Model**

For ***d* dimensions**, the Gaussian distribution of a vector $x = (x^1, x^2, \ldots, x^d)^T$ is defined by:

$$N(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

where μ is the mean and Σ is the covariance matrix of the Gaussian.

Example: $\mu = (0,0)^T$ $\Sigma = \begin{pmatrix} 0.25 & 0.30 \\ 0.30 & 1.00 \end{pmatrix}$



Introduction to Gaussian Mixture Models (GMMs)

- Gaussian Mixture Models (GMMs) assume that there are a certain number of Gaussian distributions, and each of these distributions represent a cluster.
- Hence, a Gaussian Mixture Model tends to group the data points belonging to a single distribution together.

Introduction to Gaussian Mixture Models (GMMs)

- Let’s say we have three Gaussian distributions– GD1, GD2, and GD3.
- These have a certain mean (μ1, μ2, μ3) and variance (σ1, σ2, σ3) value respectively.
- For a given set of data points, our GMM would identify the probability of each data point belonging to each of these distributions

Introduction to Gaussian Mixture Models (GMMs)

- Def: **Gaussian Mixture Models are probabilistic models and use the soft clustering approach for distributing the points in different clusters.**

The probability given in a mixture of K Gaussians is:

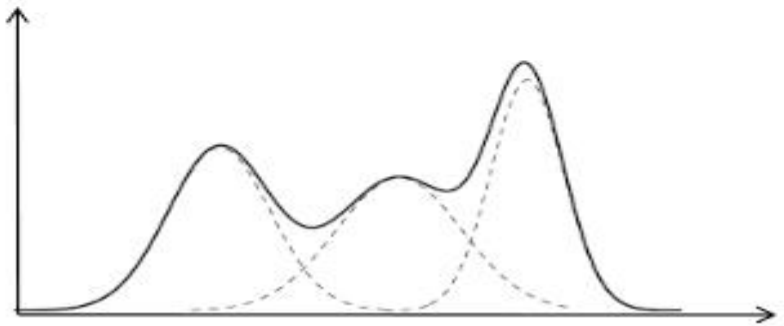
$$p(x) = \sum_{j=1}^{K} w_j \cdot N(x \mid \mu_j, \Sigma_j)$$

where w_j is the prior probability (weight) of the jth Gaussian.

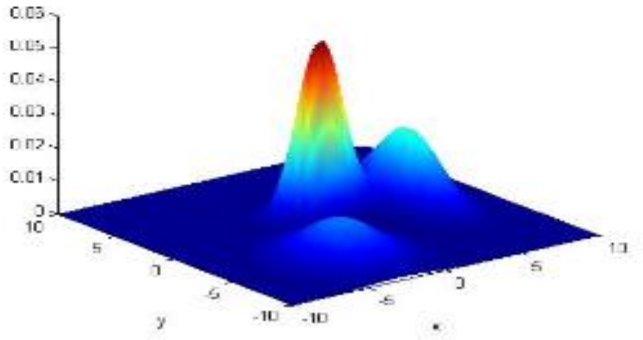
$$\sum_{j=1}^{K} w_j = 1 \qquad \text{and} \qquad 0 \le w_j \le 1$$

Examples:

d=1:

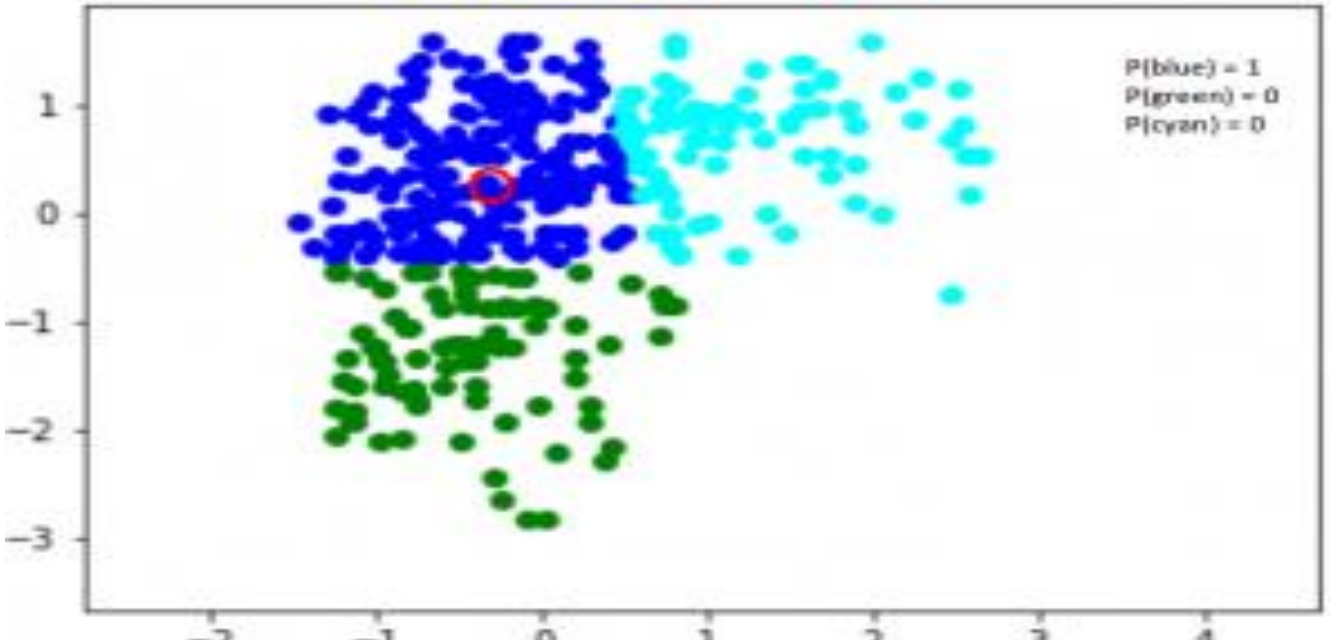


d=2:



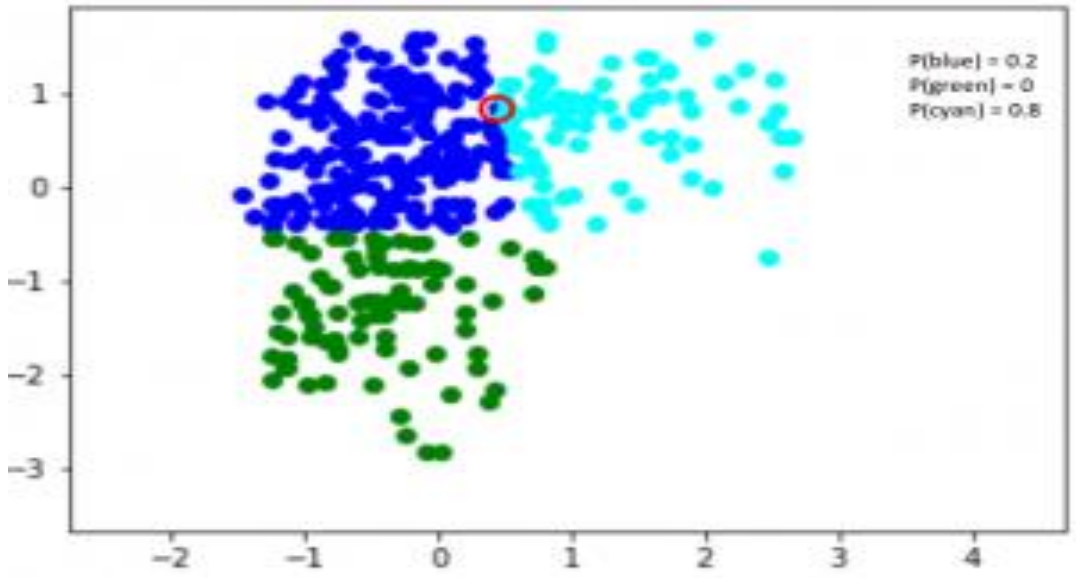
Introduction to Gaussian Mixture Models (GMMs)

- Example: Here, we have three clusters that are denoted by three colors – Blue, Green, and Cyan. Let's take the data point highlighted in red.
- The probability of this point being a part of the blue cluster is 1, while the probability of it being a part of the green or cyan clusters is 0.



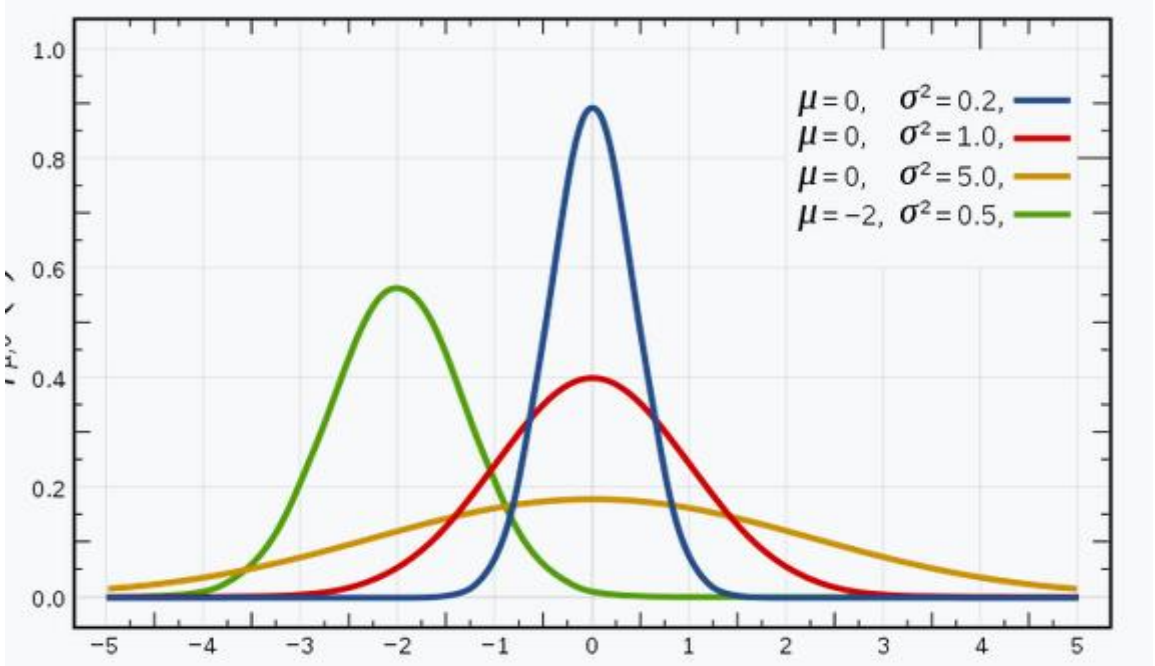
Introduction to Gaussian Mixture Models (GMMs)

- Now, consider another point – somewhere in between the blue and cyan (highlighted in the below figure).
- The probability that this point is a part of cluster green is 0, right? And the probability that this belongs to blue and cyan is 0.2 and 0.8 respectively
- Gaussian Mixture Models use the soft clustering technique for assigning data points to Gaussian distributions.



The Gaussian Distribution

- Gaussian Distributions (or the Normal Distribution): It has a bell-shaped curve, with the data points symmetrically distributed around the mean value.
- The below im has a few Gaussian distributions with a difference in mean (μ) and variance (σ^2). Remember that the higher the σ value more would be the spread:



The Gaussian Distribution

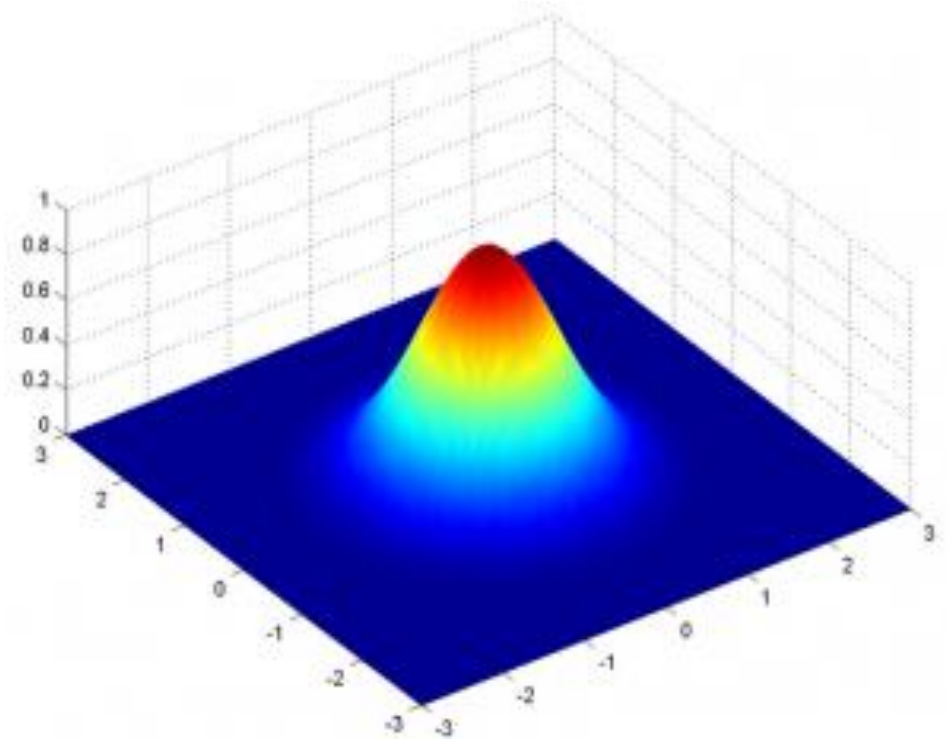
- In a one dimensional space, the probability density function of a Gaussian distribution is given by

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean and σ^2 is the variance

The Gaussian Distribution

- In the case of two variables, instead of a 2D bell-shaped curve, we will have a 3D bell curve as shown below:



The Gaussian Distribution

- The probability density function would be given by:

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right]$$

Where x is the input vector, μ is the 2D mean vector, and Σ is the 2×2 covariance matrix.
The covariance would now define the shape of this curve. We can generalize the same for d-dimensions.

- Thus, this multivariate Gaussian model would have x and μ as vectors of length d, and Σ would be a *d x d* covariance matrix.