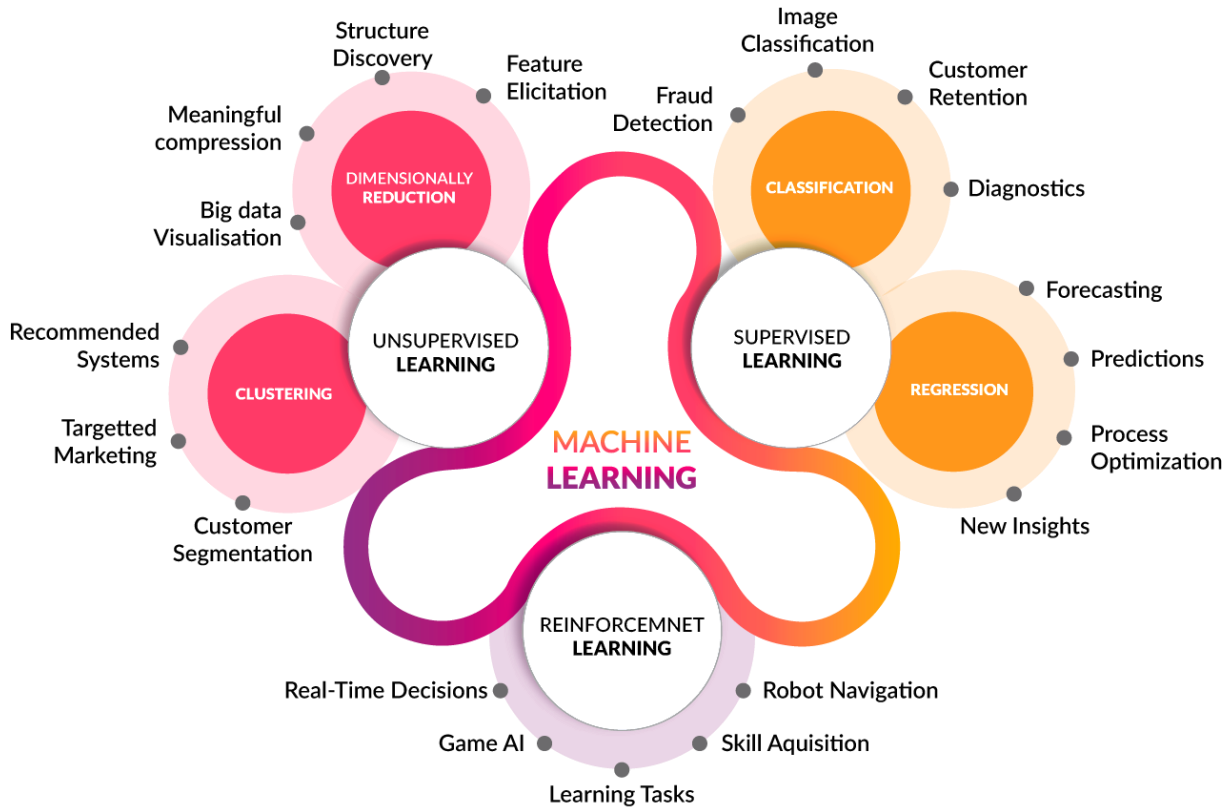


Unsupervised Learning

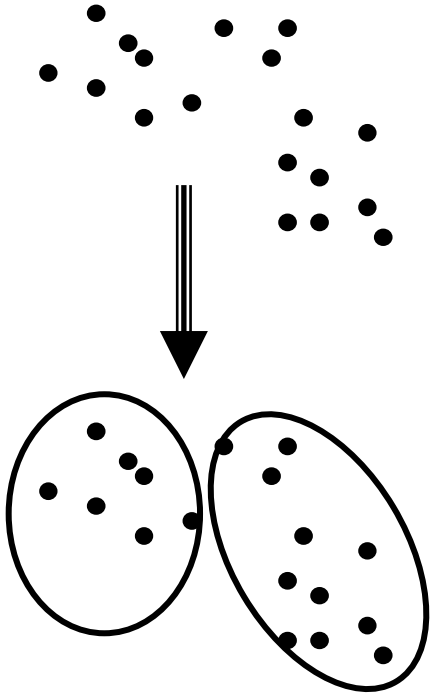
Unit 3

Introduction

- So far in our learning, a training example consisted of a set of input features and a label attached to each input record.
- Unsupervised Learning takes as training examples the set of attributes/features alone without label.
- The purpose of unsupervised learning is to attempt to find natural partitions in the training set through learning from the features.
- Two general strategies for Unsupervised learning include:
    1. Clustering
    2. Dimensionality Reduction
- As unsupervised technique doesn't use any labeled data, it learns intrinsic property from the training data.

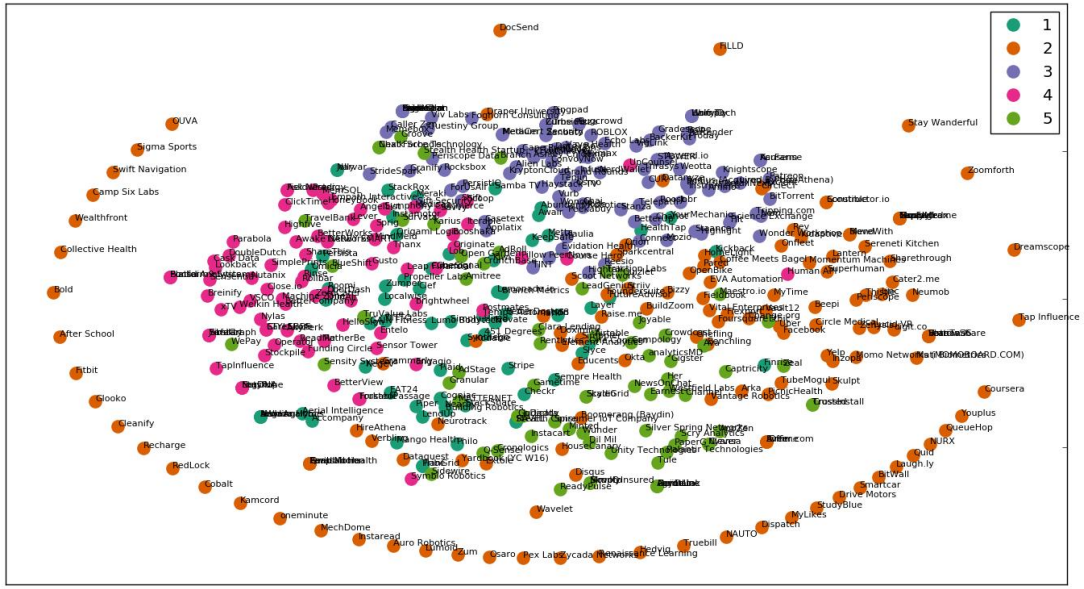
Clustering

- The goal of clustering is to group observations that are similar to each other in an unsupervised manner.
- A cluster is a group of similar observation i.e. the members within a cluster shares some similar characteristics.
- A cluster is represented by a single point known as centroid.
- **Example 1**: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.



Clustering

- **Example 2**: In marketing, segment customers according to their similarities
  - To do targeted marketing.
- **Example 3**: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy
- **Example 4**: Given a collection of ecommerce transaction, cluster them with fraud and non-fraud credit card transaction
  - To identify future fraud transaction



Clustering

- As said in the previous slide, clustering relies on the similarity/dissimilarity of data points.
- Similarity/dissimilarity is measured through various technique such as:
  - Distance
  - Density
  - Cosine Similarity etc.
- These techniques are used by various clustering algorithm depending on their nature.
- For example, K-Means algorithm uses distance measures to find similarity whereas DBSCAN algorithm uses density. Similarly, in NLP distance between two words or documents are measured using cosine similarity.

K-Means Clustering

- K-Means Clustering is the most common and simple clustering algorithm.
- It is partitioning based algorithm which uses distance as measure to find similarity.
- K in the name refers to the number of cluster expected in the given dataset where the value of K should be pre-known.
- Means in the name refers to the averaging we use to determine the centroid of cluster.
- The K clusters in K-means clustering are searched iteratively. Initially K clusters are taken in random and the cluster is identified after several iterations.
- It is highly likely that different initial cluster center lead to different results. Thus it is better to select cluster centers far away from one another.

K-Means Algorithm

1. Let X be the dataset containing m observations and n features.
2. Randomly select k cluster centers.
3. Calculate the distance between each data point to each cluster centers.
4. Compare the distance of point to each cluster centers and assign the point to cluster with minimum distance.
5. If there is no change in set membership of cluster then terminate.
6. Else, recalculate the new cluster center using mean of all the data points called centroid.
7. Repeat the process from 3 to 7.

K-Means Clustering -Numerical

- Apply K-Means Clustering algorithm to find cluster among given data points where K = 2 using K-Means algorithm.

(2, 3), (8, 2), (9, 3) (3, 1), (2, 5), (10, 3)

K-Means Clustering

- As said before, use of randomization to find initial cluster center may result differently.
- The final clusters are highly dependent on how initial clusters are declared.
- Thus to overcome this, we may use K-means++ algorithm which is improvised version of K-Means.
- Also there are other algorithm such as K-Medoids which can be used instead.
- K-means++ algorithm suggest an approach for initialization of cluster center which follows as below:

K-Means++ Initialization Algorithm

- Randomly select the first cluster center from the data points.
- For each data point in dataset, compute distance with the previously cluster center.
- Select the next cluster center from the data points which is farthest from the previously chosen cluster.
- Repeat this process until K cluster centers have been sampled from the dataset.

You are advised to research on K-Mediods and K-Mode algorithm on your own (not in the scope of syllabus.)

Hierarchical Clustering

- In hierarchical clustering, clusters are identified within a cluster itself i.e. we tend to build algorithm that finds hierarchy of clusters.
- There are two common techniques for hierarchical clustering:
  1. Agglomerative Clustering

     Uses bottom up approach, which considers each observation in dataset as single cluster initially, and gradually clusters are merged as we move up on the hierarchy tree until single cluster – covering whole dataset is found.
  2. Divisive Clustering

     This technique uses top bottom approach in which the whole dataset is first considered into single clusters and then split up into two or multiple clusters such that objects in one subgroup are dissimilar then the objects in another. This process gradually lead us until we find single member in each cluster.

Algorithms

Agglomerative Clustering

1. Consider each data point a cluster.
2. Compute the distance between each data points.
3. Repeat until K cluster remains
    I. Merge the two datapoints forming single clusters.
    II. Update the distance metrics. There are various technique to update the distance metrics.

Divisive Clustering

1. Consider all data points belong to single cluster.
2. Repeat until there is one data point in each cluster.
    I. Split the dataset into cluster using algorithms such as K-Means, K-Mediods etc.
    II. Choose the best cluster among all possibilities.

Agglomerative Clustering - Numerical Example

Apply agglomerative clustering to the following dataset.

$$A\ (1,1), B(2,3), C(3,5), D(4,5), E(6,6), and\ F(7,5)$$

Solution:

First compute the distance from each point to another point.

	A	**B**	**C**	**D**	**E**	**F**
A						
B						
C						
D						
E						
F						

Agglomerative Clustering - Numerical Example

Apply agglomerative clustering to the following dataset.

$$A\ (1,1), B(2,3), C(3,5), D(4,5), E(6,6), and\ F(7,5)$$

Solution:

First compute the distance from each point to another point.

	A	**B**	**C**	**D**	**E**	**F**
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

Agglomerative Clustering - Numerical Example

Apply agglomerative clustering to the following dataset.

$$A\ (1,1), B(2,3), C(3,5), D(4,5), E(6,6), and\ F(7,5)$$

Solution:

First compute the distance from each point to another point.

	A	**B**	**C**	**D**	**E**	**F**
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

The upper triangular section is the mirror reflection of lower triangular matrix. And you can omit the upper triangular portion.

Agglomerative Clustering - Numerical Example

Apply agglomerative clustering to the following dataset.

$$A\ (1,1), B(2,3), C(3,5), D(4,5), E(6,6), and\ F(7,5)$$

Solution:

First compute the distance from each point to another point.

	A	B	C	D	E	F
A	0					
B	2.236068	0				
C	4.472136	2.236068	0			
D	5	2.828427	1	0		
E	7.071068	5	3.162278	2.236068	0	
F	7.211103	5.385165	4	3	1.414214	0

The upper triangular section is the mirror reflection of lower triangular matrix. And you can omit the upper triangular portion.

Agglomerative Clustering - Numerical Example

- Now, considering the every point as single cluster, we merge two cluster having minimal distance.

	A	B	C	D	E	F
A	0					
B	2.236068	0				
C	4.472136	2.236068	0			
D	5	2.828427	1	0		
E	7.071068	5	3.162278	2.236068	0	
F	7.211103	5.385165	4	3	1.414214	0

C and D are having smallest distance between them. So we merge two of them.

Agglomerative Clustering - Numerical Example

After merging two cluster, we get

	A	B	{C, D}	E	F
A	0				
B	2.236068	0			
{C, D}	4.472136	2.236068	0		
E	7.071068	5	2.236068	0	
F	7.211103	5.385165	3	1.414214	0

Now, E and F are having smallest distance between them. So we merge two of them.

Agglomerative Clustering - Numerical Example

After merging two cluster, we get

	A	**B**	**{C, D}**	**{E, F}**
A	0			
B	2.236068	0		
{C, D}	4.472136	2.236068	0	
{E, F}	7.071068	5	2.236068	0

Now, E and F are having smallest distance between them. So we merge two of them.

Agglomerative Clustering - Numerical Example

After merging two cluster, we get

	A	{B, C, D}	{E, F}
A	0		
{B, C, D}	2.236068	0	
{E, F}	7.071068	2.236068	0

Now, A and {B, C, D} and {B, C, D} and {E, F} are having equal but smallest distance between them. But we take one only. Considering A and {B, C, D}, So we merge two of them.

Agglomerative Clustering - Numerical Example

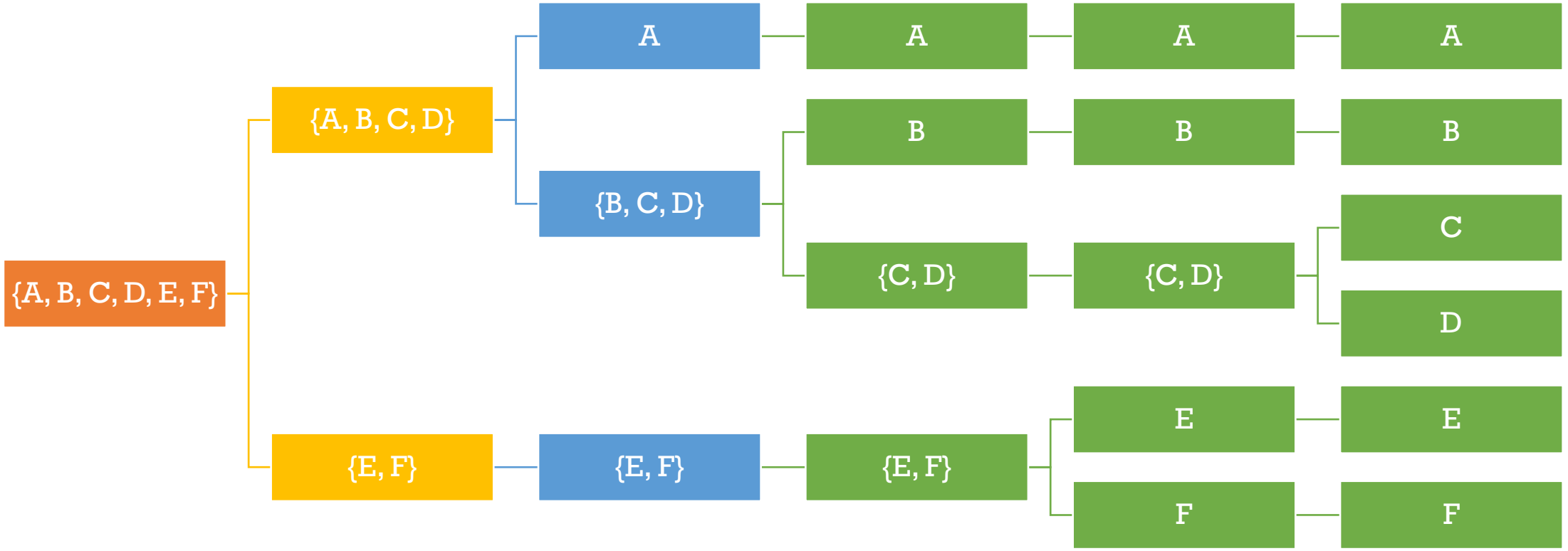
After merging two cluster, we get

	{A, B, C, D}	{E, F}
{A, B, C, D}	0	
{E, F}	2.236068	0

Finally, we merge {A, B, C, D} and {E, F} to get single cluster {A, B, C, D, E, F}.

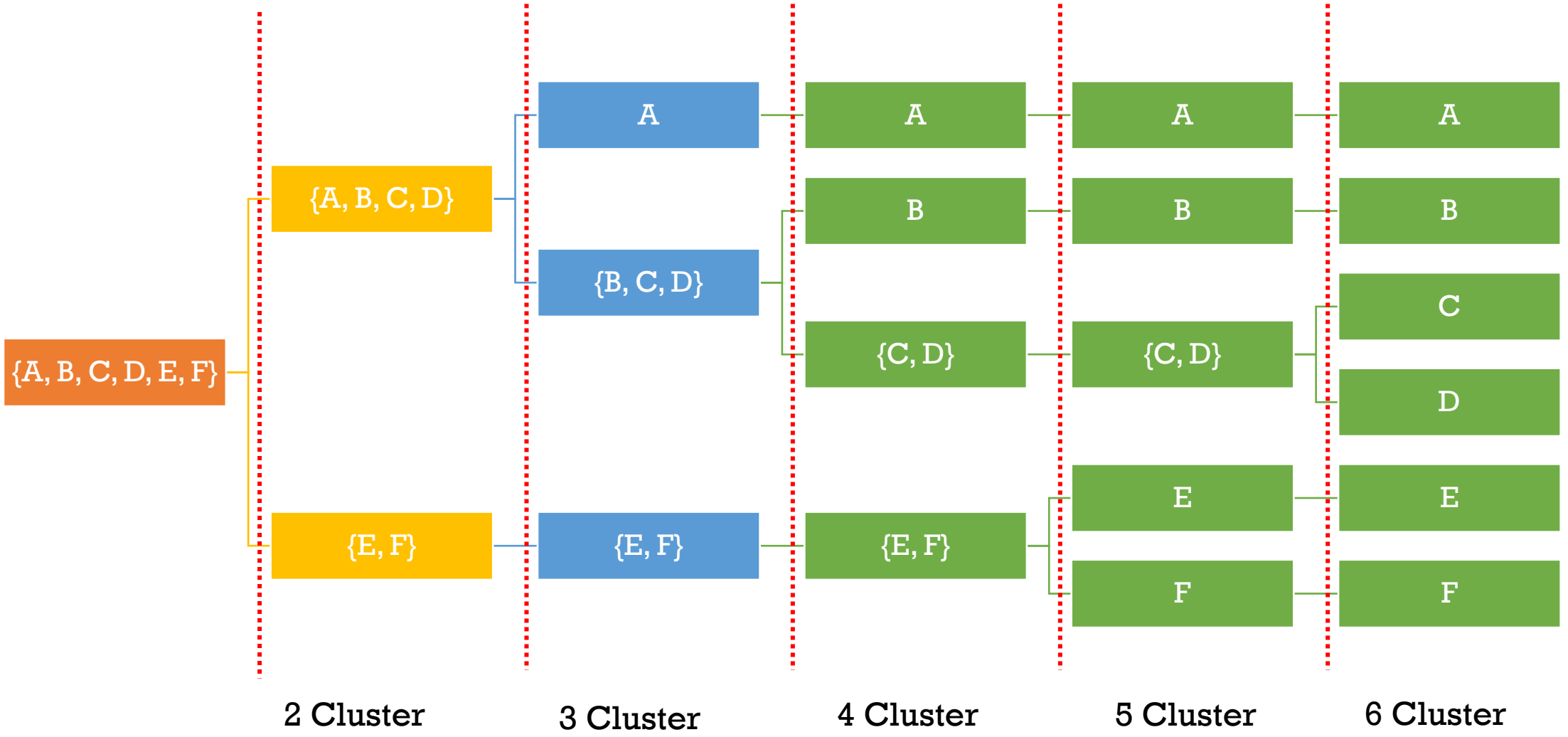
Agglomerative Clustering - Numerical Example

Constructing a dendrogram from the above we get,



Agglomerative Clustering - Numerical Example

Applying cut at different level, we get different number of cluster.



Density Based Clustering

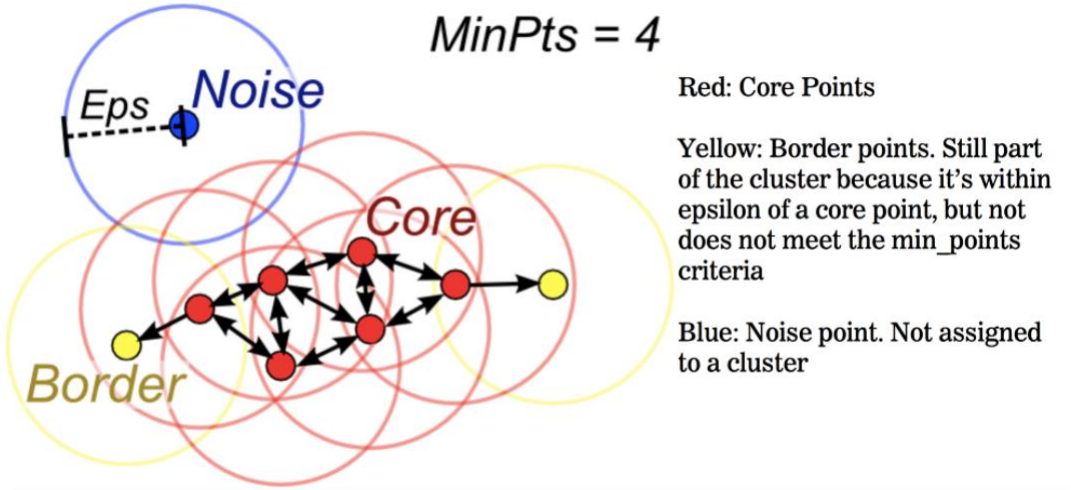
- Partition based clustering are suitable for compact and well separated clusters.
- Density based clustering is based on detecting region where observations are concentrated and where they are sparse.
- This method automatically detect patterns based on sparsity/density at particular location and the distance to some prespecified number of neighbors.
- Observation that are not part of cluster are called Noise.
- There are multiple techniques under density based clustering. Few of them are:
  - DBSCAN (In Scope of syllabus)
  - HDBSCAN (Out of Scope)
  - OPTICS (Out of Scope)

DBSCAN

- DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise.
- It groups ‘densely grouped’ observation into a single cluster and can identify clusters in large spatial datasets by looking at the local density of the data points.
- Like as Hierarchical clustering, it also doesn’t require no. of cluster to be pre-known.
- And, most importantly it is robust to outlier.
- DBSCAN uses two parameters:
  - Epsilon
  - minPoints

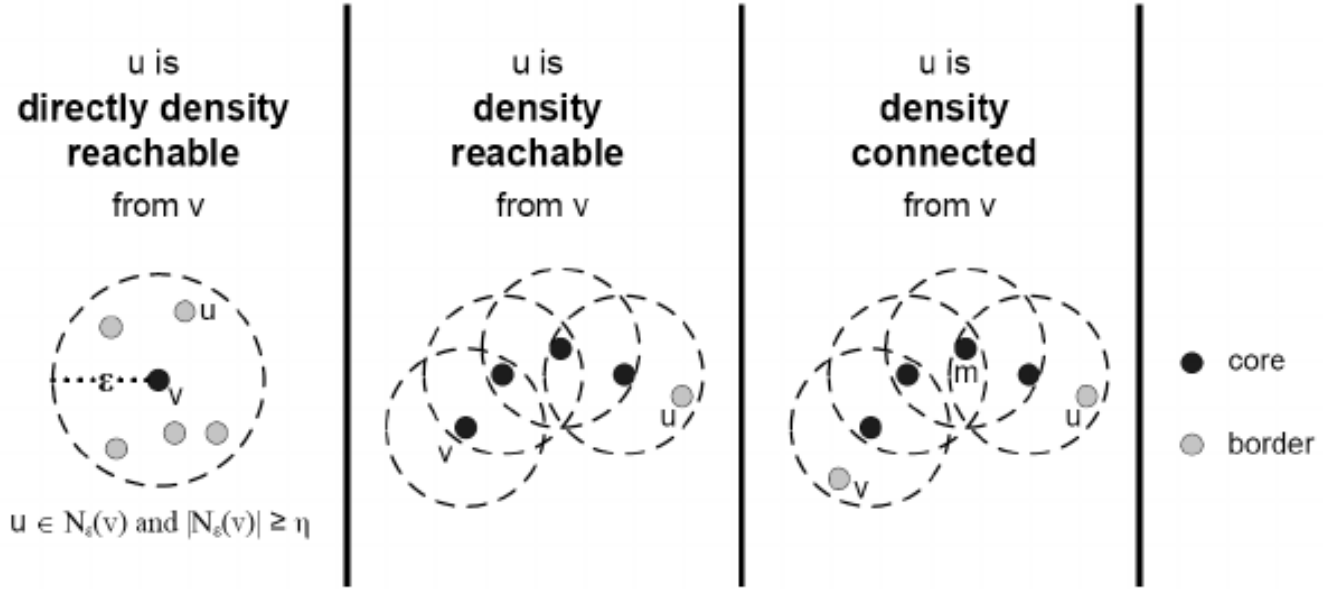
DBSCAN

- **Epsilon** (ϵ) is the radius of circular region around which the density is to be checked up.
- **minPoints** (n) is the minimum number of data points required inside that circle for that data point to be classified as a **Core point**.
- A data point is a **Core point** (x) if the circle around it contains at least n number of points.
- **Border point** (y) means Data point that has at least one point in core within epsilon (ε) distance but is lower than minPoints (n) within epsilon (ε) distance from it.
- **Noise Point** (z): Data point that has no core points within epsilon (ε) distance.



DBSCAN

- Density reachable and Density Connected
  - **Density reachable:** A point "A" is density reachable from "B" if there are a set of core points leading from "B" to "A.
  - **Density connected:** Two points "A" and "B" are density connected if there are a core point "C", such that both "A" and "B" are density reachable from "C".



DBSCAN

- Now, a density-based cluster is defined as a group of density connected points.
- We consider minPoints as a threshold for considering a cluster as a cluster. And, a cluster is only recognized if the number of observations is greater than or equal to the minPoints.
- A point x is directly density reachable from point p if a point p is a core point and x is in p's ε-neighborhood.
- All points within a ε-neighborhood of a core point belongs to same clusters and all points in a cluster are directly density reachable to core point.

DBSCAN - Algorithm

1. Choose any point p randomly
2. Identify all density reachable points from p with ε and n parameter
3. If p is a core point, create a cluster if ε and n satisfy.
4. If p is a border point, visit the next point in a dataset
5. Continue the algorithm until all points are visited.

DBSCAN – Numerical Example

Apply DBSCAN clustering to the following dataset with epsilon 3 and minimum data points 3.

$$A\,(1,1), B(2,3), C(3,5), D(4,5), E(6,6), and\ F(7,5)$$

Solution:

First compute the distance of each data point to each other.

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	1.414214
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

DBSCAN – Numerical Example

Now compare the distance to epsilon and get for which distance is greater than epsilon

For A: B
For B: A, C, D
For C: B, D
For D: B, C, E, F
For E: D, F
For F: D, E

	A	B	C	D	E	F
A	0	2.236068	4.472136	5	7.071068	7.211103
B	2.236068	0	2.236068	2.828427	5	5.385165
C	4.472136	2.236068	0	1	3.162278	4
D	5	2.828427	1	0	2.236068	3
E	7.071068	5	3.162278	2.236068	0	1.414214
F	7.211103	5.385165	4	3	1.414214	0

DBSCAN – Numerical Example

Next, we identify core points and noise

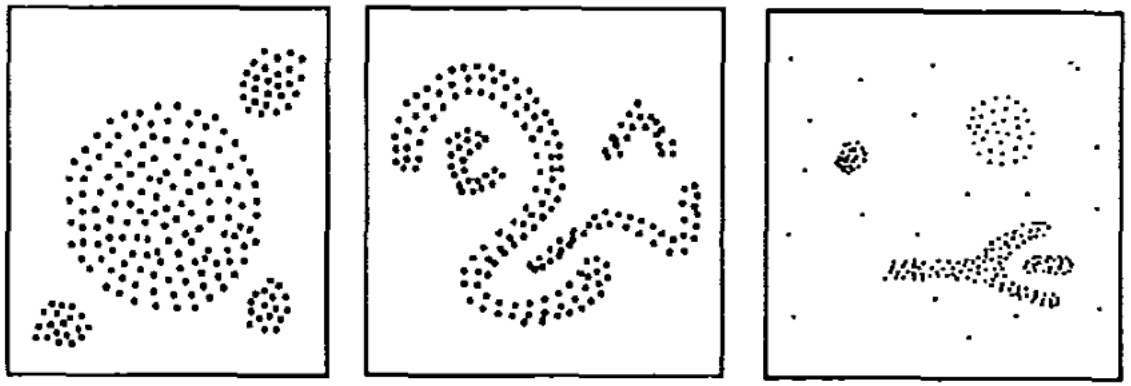
Points	Status
A	Border
B	Core Points
C	Border Points
D	Core Points
E	Border Points
F	Border

For A: B
For B: A, C, D
For C: B, D
For D: B, C, E,
For E: D, F
For F: E

And we now check for border points if any noise points are.

DBSCAN

- One of the biggest benefit of using Density based clustering is they can identify any shape of cluster. And outliers as well.



Gaussian Mixture Model

- Gaussian mixture model is soft clustering algorithm i.e. the algorithm may assign data point to multiple cluster.
- i.e. there are likely to be overlapping clusters.
- Gaussian Mixture Models assume the number of cluster to be pre-known.
- Hence, a Gaussian Mixture Model forms a cluster belonging to single distribution.
- Thus it is a statistical model that involves latent variables and hence cannot be solved using MLE method.

Gaussian Mixture Model

- A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.
- Thus, this model attempts to model the dataset as a mixture of several Gaussian Distribution
- **Gaussian Distribution**
  - Gaussian Distribution is also known as Normal Distribution.
  - It is a symmetric distribution where most of observations cluster around the centre peak and the probabilities further away from the Mean tapped off easily in both directions Extreme values of the distribution are most unlikely.
  - It has two parameters:
    1. Mean, and
    2. Standard Deviation

Gaussian Mixture Model

- Gaussian Distribution (contd.)
  - In one dimension, the pdf of Gaussian Distribution is given as:

$$G(X|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\,\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - And, in multi dimension, the pdf is given as:

$$G(X|\mu,\Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}}\exp(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu))$$

Where,

μ is d dimensional vector denoting the mean of the distribution

And Σ is the $d \times d$ covariance matrix.

Gaussian Mixture Model

- If we have only one dimension, we can easily estimate the parameters μ and σ using ***maximum-likelihood*** method.
- But if there are say K clusters i.e. there are K different distributions first we define the combined pdf of K clusters as linear function of densities of all these K distributions.

$$i.e.\, p(X) = \sum_{k=1}^{K} \pi_k G(X|\mu_k, \Sigma_k)$$

Where π_k is the mixing coefficient for k-th distribution.

- Now, we estimate the parameters by the maximum log-likelihood method.

Gaussian Mixture Model

- And, finally we get,

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n) x_n}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_k(x_n)(x_n - \mu_k)(x_n - \mu_k)\text{^}T}{\sum_{n=1}^{N} \gamma_k(x_n)}$$

And,

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_k(x_n)$$

Here, $\gamma_k(x)$ is a random variable.

Gaussian Mixture Model

- To conclude, we can see that the parameters can be estimated easily through maximum likelihood.
- For this Expectation Maximization algorithm is beneficial.
- The Expectation-Maximization (EM) algorithm is an iterative way to find maximum-likelihood estimates for model parameters when the data is incomplete or has some missing data points or has some hidden variables.
- EM chooses some random values for the missing data points and estimates a new set of data (on missing part).
- These new values are then recursively used to estimate a better first data, by filling up missing points, until the values get fixed.

Gaussian Mixture Model

- There are two steps in the Expectation-Maximization algorithm.

1. Estimation Step (E-Step)
   I. First, initialize our model parameters i.e. μ_k, Σ_k, and γ_k mean, covariance matrix, and mixing coefficients respectively.
   II. Now, calculate posterior probabilities, using bayes theorem, of data points belonging to each cluster using current values. These probabilities are represented by γ_k.

$$\gamma_k(X) = \frac{p(X|k)p(k)}{\sum_{k=1}^{K} p(k)p(X|k)}$$

   I. Finally, estimate the value of the lantent variables γ_k based on the current parameter values.

2. Maximization Step (M-Step)

Gaussian Mixture Model

2. Maximization Step (M-Step)

- Update the parameter values using the estimated latent variables (γ_k)
  - Update the mean of the cluster point (μ_k) by taking the weighted average of data points using the corresponding latent variable probabilities.

$$\mu_k = \frac{1}{K}(\gamma_1 x_1 + \cdots + \gamma_K x_K)$$

  - Similarly, update the covariance matrix by taking the weighted average of the squared differences between the data points and the mean, using the latent variable probabilities.

$$\Sigma_k = \frac{1}{K}(\gamma_1 (x_1 - \mu_1)^2 + \cdots + \gamma_K (x_K - \mu_K)\text{^}2\ )$$

  - And, finally update the mixing coefficients by taking the average of the latent variable probabilities for each component.

Gaussian Mixture Model

- Repeat E-step and M-step until convergence.

Derivation: https://core.ac.uk/download/pdf/82750424.pdf

Dimensionality Reduction

- Higher the dimension of data, more the model complexity is and difficult to train the model.
- Also, increasing the features will not always improve classification accuracy.
- Thus, in machine learning world, **Curse of Dimensionality** is common phrase which basically refers to the point where having more dimension on data doesn't help in the conclusion, rather it would be the pain.
- Thus, the purpose of dimensionality reduction is to reduced the number of dimension or features from dataset.
- But while doing so, the pattern from the data we are trying to learn shouldn't be distorted.

Dimensionality Reduction

- However, reducing the features always comes with the cost and may impact the accuracy.
- Thus it may be challenging to reduce the feature without any significance loss in accuracy.
- Thus, dimensionality reduction deals with the process of reducing the number of variables under consideration by obtaining smaller set of variables.
- There are many methods common for dimensionality reduction like:
  - Principal Component Analysis
  - Feature Selection
  - Low Variance Filter
  - Linear Discriminant Analysis etc.