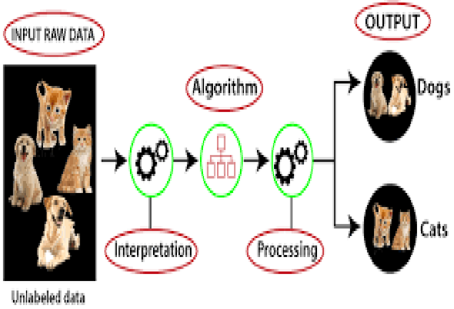
Course: Machine Learning Techniques (CSE3008) - Module4

Dr.Mohana S D,
Assistant Professor,
(Course In-charge - CSE3008)
School of Computer Science and Engineering & Information Science,
Presidency University Bengaluru.

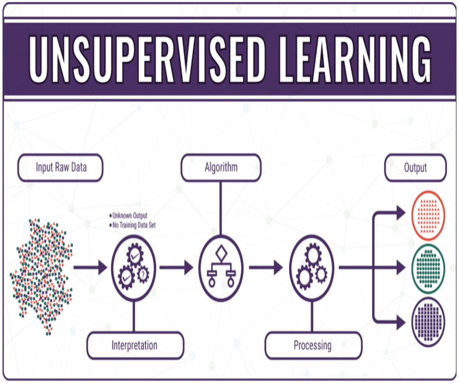
Unsupervised Learning


INPUT RAW DATA
Unlabeled data
Interpretation
Algorithm
Processing
OUTPUT
Dogs
Cats

Unsupervised Learning

Definition

- Unsupervised learning is a type of machine learning where the model is trained on a dataset without any labeled output.
- The goal of unsupervised learning is to find patterns, structures, or relationships in the data that can be used to gain insights, make predictions, or classify new data.


UNSUPERVISED LEARNING
Input Raw Data
•Unknown Output
•No Training Data Set
Interpretation
Algorithm
Processing
Output

Unsupervised Learning

There are several common techniques used in unsupervised learning, including:

Clustering:

Clustering: Clustering is a technique used to group similar data points together based on some similarity metric. The most common clustering algorithms are k-means, hierarchical clustering, and density-based clustering.

Dimensionality Reduction:

Dimensionality Reduction: Dimensionality reduction is a technique used to reduce the number of features or variables in a dataset while preserving the most important information. Principal Component Analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) are common dimensionality reduction techniques.

Unsupervised Learning

Anomaly Detection:

Anomaly Detection: Anomaly detection is a technique used to identify data points that are significantly different from the majority of the data. Isolation Forest and Local Outlier Factor (LOF) are common anomaly detection algorithms.

Association Rule Mining:

Association Rule Mining: Association rule mining is a technique used to identify patterns in data that occur together frequently. Apriori and FP-Growth are common association rule mining algorithms.

Generative Models:

Generative Models: Generative models are algorithms that attempt to model the underlying probability distribution of the data. Gaussian Mixture Models (GMM) and Variational Autoencoders (VAE) are common generative models.

Unsupervised learning

Unsupervised learning has a wide range of applications, including customer segmentation, image and speech recognition, anomaly detection, and natural language processing. By finding patterns and relationships in large and complex datasets, unsupervised learning can provide valuable insights and help solve real-world problems.

Unsupervised learning

- K-Means clustering is a popular unsupervised learning algorithm that aims to partition a set of data points into K clusters based on the similarity between them.
- The algorithm works by iteratively assigning each data point to its nearest cluster centroid and updating the centroids based on the mean of the assigned data points.
- There are two main variants of the K-Means algorithm: simple K-Means and mini-batch K-Means.

Unsupervised learning

K-Means

In simple K-Means, the algorithm starts by randomly selecting K data points as the initial centroids. Then, it iteratively assigns each data point to its nearest centroid and updates the centroids based on the mean of the assigned data points. The process continues until the centroids no longer move significantly or a maximum number of iterations is reached.

K-Means

Mini-batch K-Means is a variant of K-Means that is designed to handle large datasets more efficiently. Instead of using all the data points in each iteration, mini-batch K-Means randomly selects a small subset (batch) of the data points to update the centroids. This reduces the computational complexity of the algorithm and allows it to handle large datasets in a reasonable amount of time.

Unsupervised learning

K-Means

To update the centroids incrementally, we can use the following steps:

- Initialize the centroids: Randomly select K data points as the initial centroids.
- Assign data points to centroids: For each data point, compute its distance to each centroid and assign it to the nearest centroid.
- Update the centroids: For each centroid, compute the mean of the assigned data points and use it as the new centroid.
- Incremental update: When a new data point is added to the dataset, we can update the centroids incrementally by only computing the mean of the new data point and the current centroid. This avoids the need to recompute the mean of all the data points assigned to the centroid.
- Repeat steps 2-4 until the centroids no longer move significantly or a maximum number of iterations is reached.

Unsupervised learning

K-Means

Overall, K-Means clustering is a powerful technique for partitioning data into clusters based on their similarity. The algorithm can be further improved by using mini-batch K-Means to handle large datasets and updating the centroids incrementally to handle streaming data.

K-Means

Finding the optimal number of clusters in K-Means clustering is an important task, as it can greatly impact the accuracy of the clustering. Two commonly used methods for determining the optimal number of clusters are the Elbow method and the Silhouette coefficient.

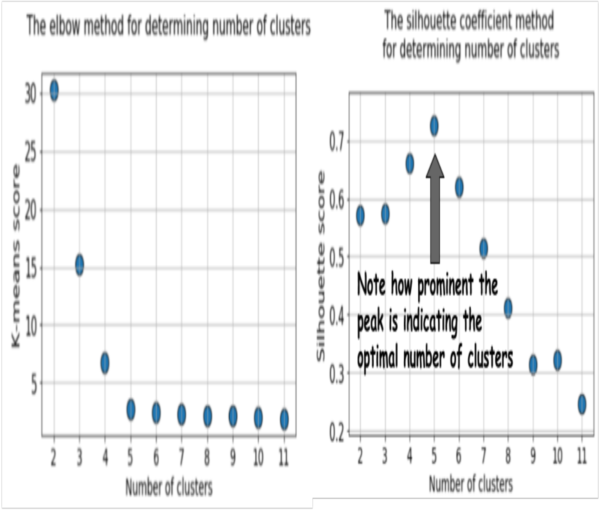
Unsupervised learning

Elbow method:

The Elbow method is a heuristic technique that involves plotting the Within-Cluster Sum of Squares (WCSS) against the number of clusters, K. WCSS is the sum of the squared distances between each data point and its assigned centroid. The idea is to choose the number of clusters, K, at the "elbow" of the curve, which is the point where the reduction in WCSS starts to diminish significantly.

Silhouette coefficient:

The Silhouette coefficient is a metric that measures the quality of a clustering solution. It takes into account both the cohesion (how close the data points are to their assigned centroid) and the separation (how far the data points are from other centroids). The Silhouette coefficient ranges from -1 to 1, with higher values indicating better clustering solutions. To find the optimal number of clusters using the Silhouette coefficient, we can calculate the coefficient for each value of K and choose the one with the highest average coefficient.


The elbow method for determining number of clusters
K-means score
30
25
20
15
10
5
2 3 4 5 6 7 8 9 10 11
Number of clusters
The silhouette coefficient method for determining number of clusters
Silhouette score
0.7
0.6
0.5
0.4
0.3
0.2
Note how prominent the peak is indicating the optimal number of clusters
2 3 4 5 6 7 8 9 10 11
Number of clusters

Unsupervised learning

Drawbacks of K-Means:

Although K-Means is a popular and effective clustering algorithm, it has several limitations, including:

- Sensitivity to the initial centroids: The quality of the clustering solution can be highly dependent on the initial randomly selected centroids.
- Inability to handle non-linearly separable data: K-Means assumes that the clusters are convex and separable, which may not be true for all datasets.
- Difficulty in determining the optimal number of clusters: Choosing the number of clusters can be a challenging task, as it requires manual intervention or the use of heuristic techniques.

Unsupervised learning

K-Means++ :

- K-Means++ is an improvement over the original K-Means algorithm that addresses the sensitivity to the initial centroids problem.
- Instead of randomly selecting the initial centroids, K-Means++ uses a smarter initialization strategy that selects the initial centroids with a higher probability of being far from each other.
- This helps to improve the quality of the clustering solution and reduce the number of iterations needed to converge.

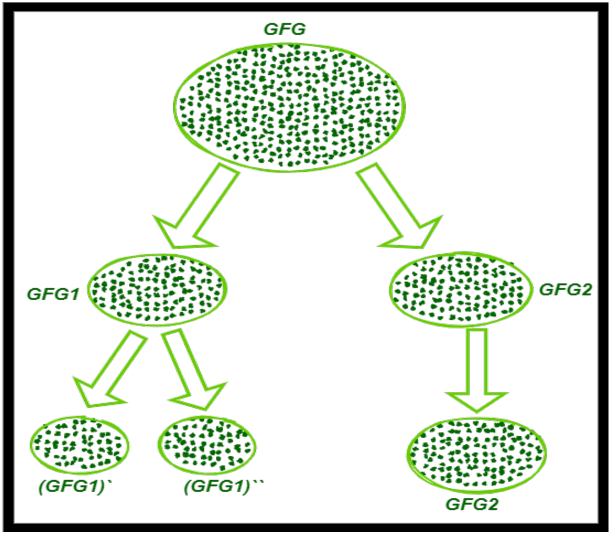
K-Means++

Divisive hierarchical clustering is a type of hierarchical clustering algorithm that works by recursively dividing a dataset into smaller and smaller subsets until each subset contains only one data point. Two commonly used methods for performing divisive hierarchical clustering are bisecting K-Means and clustering using Minimum Spanning Tree (MST).

Unsupervised learning

Bisecting K-Means:

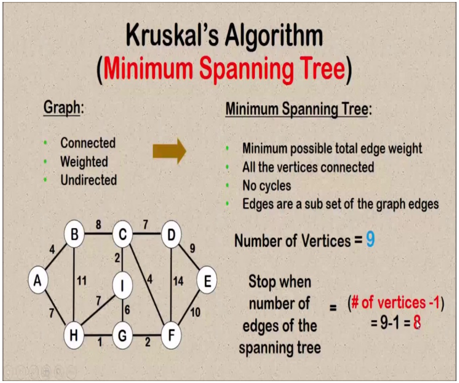
- Bisecting K-Means is a type of divisive hierarchical clustering that works by recursively dividing the dataset into two subsets using K-Means clustering.
- The algorithm starts by treating the entire dataset as one cluster and applies K-Means clustering to it.
- The resulting clusters are then bisected into two subsets by running K-Means clustering again on each of the clusters.
- The process continues until the desired number of clusters is reached.


GFG
GFG1
GFG2
(GFG1)`
(GFG1)``
GFG2

Unsupervised learning

Clustering using Minimum Spanning Tree (MST):

- Clustering using Minimum Spanning Tree (MST) is a type of divisive hierarchical clustering that works by constructing a Minimum Spanning Tree of the dataset and recursively dividing it into smaller subtrees.
- The algorithm starts by constructing a Minimum Spanning Tree of the dataset, which is a tree that connects all the data points with the minimum total edge weight.
- The tree is then recursively bisected into two subtrees using a clustering criterion such as K-Means. The process continues until the desired number of clusters is reached.



Kruskal's Algorithm (Minimum Spanning Tree)

Graph:

- Connected
- Weighted
- Undirected

Minimum Spanning Tree:

- Minimum possible total edge weight
- All the vertices connected
- No cycles
- Edges are a sub set of the graph edges

Number of Vertices = 9

Stop when number of edges of the spanning tree = (# of vertices -1) = 9-1 = 8

Bisecting K-Means and clustering using MST

Both bisecting K-Means and clustering using MST have their advantages and disadvantages.

- Bisecting K-Means is a faster algorithm and can handle large datasets, but it may not always produce the best clustering solution.
- Clustering using MST, on the other hand, produces more accurate clustering solutions but can be computationally expensive for large datasets.
- The choice of algorithm depends on the specific characteristics of the dataset and the desired level of accuracy and computational efficiency.

Competitive Learning:

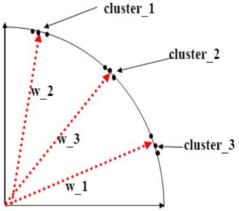
Competitive learning is a type of unsupervised learning that involves training a neural network to learn a set of features or patterns from the input data.
One popular competitive learning algorithm is Kohonen's Self-Organizing Maps (SOM), which is a type of neural network that can be used for clustering.

Clustering using Kohonen's Self-Organizing Maps:

- Kohonen's Self-Organizing Maps (SOM) is a type of neural network that can be used for clustering.
- The algorithm works by mapping the input data onto a low-dimensional grid of neurons.
- Each neuron is connected to the input data through a set of weights, which are adjusted during the training process to learn the patterns in the data.
- The neurons are organized in such a way that neurons that are close to each other on the grid respond to similar input patterns.
- The resulting clusters are represented by groups of neurons that are close to each other on the grid.

Self-Organizing Maps (SOM) (§ 5.5)

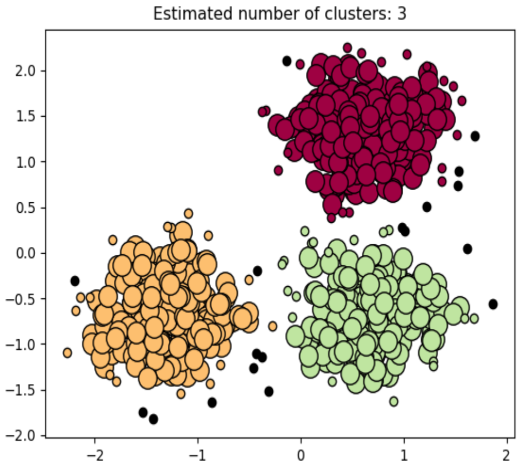
- Competitive learning (Kohonen 1982) is a special case of SOM (Kohonen 1989)
- In competitive learning,
  - the network is trained to organize input vector space into subspaces/classes/clusters
  - each output node corresponds to one class
  - the output nodes are not ordered: ***random map***



- The topological order of the three clusters is 1, 2, 3
- The order of their maps at output nodes are 2, 3, 1
- The map does not preserve the topological order of the training vectors

Density-Based Spatial Clustering - DBSCAN:

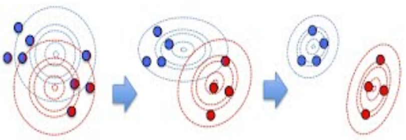
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm that can be used to identify clusters of arbitrary shape.
- The algorithm works by identifying regions of high density in the data and grouping the data points that belong to these regions into clusters.
- DBSCAN has two parameters: epsilon ϵ, which determines the radius of the neighborhood around each data point, and minPts, which determines the minimum number of data points required to form a cluster.


Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

Gaussian Mixture Models (GMM):

- Gaussian Mixture Models (GMM) is a clustering algorithm that models the data as a mixture of Gaussian distributions.
- The algorithm works by estimating the parameters of the Gaussian distributions using the Expectation-Maximization (EM) algorithm.
- The resulting clusters are represented by the Gaussian distributions that best fit the data.
- GMM is a flexible algorithm that can model clusters of different shapes and sizes.

Gaussian Mixture Model



- Data with D attributes, from Gaussian sources $c_1 \dots c_k$
  - how typical is $\mathbf{x}_i$ under source **c** $$P(\vec{x}_i \mid c) = \frac{1}{\sqrt{2\pi|\Sigma_c|}} \exp\left\{ -\tfrac{1}{2} \underbrace{(\vec{x}_i - \vec{\mu}_c)^T \Sigma_c^{-1} (\vec{x}_i - \vec{\mu}_c)}_{\sum_a \sum_b (x_{ia} - \mu_{ca}) [\Sigma_c^{-1}]_{ab} (x_{ib} - \mu_{cb})} \right\}$$
  - how likely that $\mathbf{x}_i$ came from **c** $$P(c \mid \vec{x}_i) = \frac{P(\vec{x}_i \mid c) P(c)}{\sum_{c=1}^{k} P(\vec{x}_i \mid c) P(c)}$$
  - how important is $\mathbf{x}_i$ for source **c**: $w_{ic} = P(c \mid \vec{x}_i) / \left(P(c \mid \vec{x}_1) + \dots + P(c \mid \vec{x}_n) \right)$
  - mean of attribute **a** in items assigned to **c**: $\mu_{ca} = w_{c1} x_{1a} + \dots + w_{cn} x_{na}$
  - covariance of **a** and **b** in items from **c**: $\Sigma_{cab} = \sum_{i=1}^{n} w_{ci} (x_{ia} - \mu_{ca})(x_{ib} - \mu_{cb})$
  - prior: how many items assigned to **c**: $P(c) = \frac{1}{n} \left(P(c \mid \vec{x}_1) + \dots + P(c \mid \vec{x}_n) \right)$

Outlier Detection methods:

Outlier detection is a type of unsupervised learning that involves identifying data points that are significantly different from the majority of the data. Two popular outlier detection methods are Isolation Forest and Local Outlier Factor (LOF).

Isolation Forest:

Isolation Forest is an outlier detection algorithm that works by randomly partitioning the data points into subsets until each subset contains only one data point. The algorithm then computes the path length for each data point in the tree and uses it to determine the anomaly score.

Isolation Forest:

Local Outlier Factor (LOF) is an outlier detection algorithm that works by measuring the local density of a data point relative to its neighbors. The algorithm computes a score for each data point based on its distance to its k-nearest neighbors and their average distance to each other. Points with a high LOF score are considered outliers.

Algorithm 2 $LOFk, m, D$

Input: k - number of near neighbor, m - number of outliers, D - outlier candidate dataset.

Output: $topm$ outliers.

```
1: for $j = 1$ to $lenD$ do do
2:     compute $k$ - $distp$
3:     compute $N_k p$
4: end for
5: calculate $reach$ - $dist_k p, r$ and $lrdp$
6: calculate $lofp$
7: sort the $lof$ values of all points in descending order
8: return the $m$ data objects with the large $lof$ values, which
   are the outliers
```

thank
you