S21CSE471: Statistical Methods in AI **Spring 2021**

Clustering Algorithms

Prepared by: P.Srihari, D.Shaarada Yamini, Vaibhav Gupta

1 Unsupervised learning

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected

Here, are prime reasons for using Unsupervised Learning:

- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

2 Clustering

Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data. You can also modify how many clusters your algorithms should identify. It allows you to adjust the granularity of these groups.

There are different types of clustering you can utilize:

- Hierarchical clustering
- K-means clustering
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

3 Hierarchical Clustering

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

There are two types of hierarchical clustering algorithms:

Agglomerative — Bottom up approach. Start with many small clusters and merge them together to create bigger clusters.

Divisive — Top down approach. Start with a single cluster than break it up into smaller clusters.

3.1 Agglomerative Clustering

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:
(1) Identify the two clusters that are closest together.
(2) Merge the two most similar clusters.

This iterative process continues until all the clusters are merged together. There are some linkage criteria available for merging. They are discussed below:

Single linkage: Minimizing distance between dataset clusters/observations of pairs of clusters.
Average clusters: Minimizes the average of distances between all observations of pairs of clusters.
Complete linkage: Minimizes the maximum distance between all observations of pairs of clusters.
Ward: Maximizes the sum of squared differences between all clusters. It is a variance minimizing approach and in this sense it is similar to K-means objective function but tackled with an agglomerative hierarchical approach.d

3.1.1 Single Linkage in action

Consider we have 5 samples and we need to apply hierarchical agglomerative clustering to build nested clusters. The sample data S = x1, x2, x3, x4, x5. We start with computing a 5*5 distance matrix using similarity measures (Euclidean distance).

$$S_{M_0} = \begin{pmatrix} 10^4 & d(x_1,x_2) & d(x_1,x_3) & d(x_1,x_4) & d(x_1,x_5) \\ d(x_2,x_1) & 10^4 & d(x_2,x_3) & d(x_2,x_4) & d(x_2,x_5) \\ d(x_3,x_1) & d(x_3,x_2) & 10^4 & d(x_3,x_4) & d(x_3,x_5) \\ d(x_4,x_1) & d(x_4,x_2) & d(x_4,x_3) & 10^4 & d(x_4,x_5) \\ d(x_5,x_1) & d(x_5,x_2) & d(x_5,x_3) & d(x_5,x_4) & 10^4 \end{pmatrix}$$

We select the diagonal elements to be of arbitrarily large values.
The matrix will be symmetric i.e

$$d(x_i, x_j) = d(x_j, x_i), for\ i\ \neq\ j$$

$$d(x_i, x_j) = \sqrt{(x_i - x_j)^T.(x_i - x_j)}$$

We now pick up the minimum distance from the matrix and then find the sample associated with it.

$$i^*, j^* = \arg\min_{(i,j)} d(x_i, y_i)\ , with\ i, j\ \epsilon\ 1, 2, ...|S|$$

We pick up i^*, j^* and club them together as one cluster. Let us assume in our case $i^* = 2$ and $j^* = 5$.

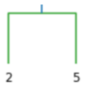

Figure 1: Points 2 and 5 are clustered

We get $S_1 = \{x_1, x_{(2,5)}, x_3, x_4\}$ and we compute the distance matrix again in the following way.

$$d(x_1, x_3) = \sqrt{(x_1 - x_3)^T.(x_1 - x_3)} \tag{1}$$

$$d(x_{(2,5)}, x_1) = \min(d(x_2, x_1), d(x_5, x_1)) \quad \text{implies single linkage} \tag{2}$$

Other distances like $d(x_3, x_4)$ and $d(x_1, x_4)$ are computed using equation(1).
Where as

$$d(x_(2,5), x_3) = \min(d(x_2, x_3), d(x_5, x_3))$$
$$d(x_(2,5), x_4) = \min(d(x_2, x_4), d(x_5, x_4))$$

which are computed as per single linkage criterion

We get the updated distance matrix as:

$$S_{M_1} = \begin{pmatrix} 10^4 & d(x_1, x_{(2,5)}) & d(x_1, x_3) & d(x_1, x_4) \\ d(x_{(2,5)}, x_1) & 10^4 & d(x_{(2,5)}, x_3) & d(x_{(2,5)}, x_4) \\ d(x_3, x_1) & d(x_3, x_{(2,5)}) & 10^4 & d(x_3, x_4) \\ d(x_4, x_1) & d(x_4, x_{(2,5)}) & d(x_4, x_3) & 10^4 \end{pmatrix}$$

We compute the minimum distance from the matrix as we computed previously. Let's suppose $i^* = 4$, and $j^* = (2, 5)$.

Then $S_2 = \{x_1, x_((2, 5), 4), x_3\}$ will be our updated sample.

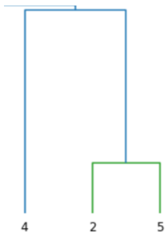

Figure 2: Point 4 and Cluster (2,5) are clustered

We will now proceed with computation of similarity matrix with new clusters. We get the updated distance matrix as:

$$S_{M_2} = \begin{pmatrix} 10^4 & d(x_1, x_{((2,5),4)}) & d(x_1, x_3) \\ d(x_{((2,5),4)}, x_1) & 10^4 & d(x_{((2,5),4)}, x_3) \\ d(x_3, x_1) & d(x_3, x_{((2,5),4)}) & 10^4 \end{pmatrix}$$

We compute the minimum distance from the matrix as we computed previously. Let's suppose $i^* = 1$, and $j^* = 3$.

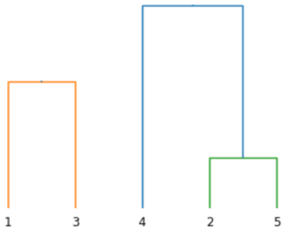

Figure 3: Cluster (1,3), Point 1 and Point 3 are clustered

We get $S_3 = \{x_{(1,3)}, x_{((2,5),4)}\}$ will be our updated sample We can compute the similarity matrix to obtain final link to complete the process of Agglomerative hierarchical clustering using single link

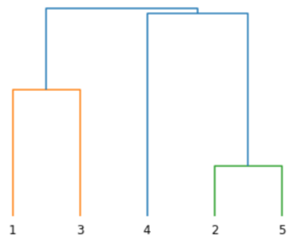

Figure 4: Cluster (1,3), Point 4 and Cluster (2,5) are clustered

Sample	**Size**	**Cluster**
$\mathrm{S}_0 = (x_1, x_2, x_3, x_4, x_5)$	5	-
$\mathrm{S}_1 = (x_1, x_{(2,5)}, x_3, x_4)$	4	(x_2, x_5)
$\mathrm{S}_2 = (x_1, x_{((2,5),4)}, x_3)$	3	$((\mathrm{x}_2, x_5), x_4)$
$\mathrm{S}_3 = ((x_1, x_3), x_{((2,5),4)})$	2	$((\mathrm{x}_2, x_5), x_4), (x_1, x_3)$
$\mathrm{S}_4 = (((x_1, x_3), x_{((2,5),4)}))$	1	$((\mathrm{x}_1, x_3), (x_2, x_5), x_4))$

3.1.2 Other Linkage Criterion

Below mentioned are the alternative linkage criterion to single linkage which can be considered for clustering.

Linkage Method	**Formula**
Single linkage	$\mathrm{d}(\mathrm{x}_{(i,j)}, x_k) = min(d(x_i, x_k), d(x_j, x_k))$
Complete linkage	$\mathrm{d}(\mathrm{x}_{(i,j)}, x_k) = max(d(x_i, x_k), d(x_j, x_k))$
Average unweighted linkage	$\mathrm{d}(\mathrm{x}_{(i,j)}, x_k) = (d(x_i, x_k) + d(x_j, x_k))/\lvert x_{(i,j)}\rvert\lvert x_k\rvert$
Average weighted linkage	$\mathrm{d}(\mathrm{x}_{(i,j)}, x_k) = 0.5(d(x_i, x_k) + d(x_j, x_k))$

4 Divisive Clustering

Divisive Clustering is a top-down approach. The process starts at the root with all the points as one cluster. It recursively splits the higher level clusters to build the dendrogram. This can be considered as a global approach as we are considering all the points to start the process. These algorithms can be more efficient when compared with agglomerative clustering.

4.1 K-means Clustering

K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter ”k,” which is fixed beforehand. The clusters are then positioned as points and all observations or data points are associated with the nearest cluster, computed, adjusted and then the process starts over using the new adjustments until a desired result is reached.

K-means clustering has uses in search engines, market segmentation, statistics and even astronomy.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

  Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The way kmeans algorithm works is as follows:

- Specify number of clusters K.
- Shuffle the dataset and select K random data points for the centroids .
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach kmeans follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.

The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2 \tag{1}$$

where $w_{ik} = 1$ for data point x_i if it belongs to cluster k, otherwise, $w_{ik} = 0$. Also, $_k$ is the centroid of $x_i^{\prime}s$ cluster.

It's a minimization problem of two parts. We first minimize J w.r.t. w_{ik} and treat $_k$ fixed. Then we minimize J w.r.t. $_k$ and treat w_{ik} fixed. Basically , we differentiate J w.r.t. w_{ik} first and update cluster assignments (E-step). Then we differentiate J w.r.t. $_k$ and recompute the centroids after the cluster assignments from previous step (M-step). Therefore, E-step is:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In other words, assign the data point x_i to the closest cluster judged by its sum of squared distance from cluster's centroid. And M-step is:

$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik}x^i}{\sum_{i=1}^{m} w_{ik}} \tag{3}$$

Which translates to recomputing the centroid of each cluster to reflect the new assignments.

4.2 K-means ++ Clustering

K-means algorithm with a better approach for choosing initial K centers.

1) Randomly choose one of the observation to be a cluster center.
2) For each observation x, determine $d(x)$, where $d(x)$ denotes the minimal distance from the x to a cluster center z_i.

$$d(x) = \min(d(x, z_i)^2)$$

3) Choose next cluster center, to choose first build a distribution with $d(x)^2$ and perform sampling to choose the center.
4) Repeat 2 and 3 until K cluster centers are identified.

Once K centers identified, K-means algorithms is applied until convergence.

4.2.1 Understanding Sampling

Instead of choosing the farthest point from centers we are sampling by forming a cumulative distribution of $d(x)$. Sampling introduces randomness which is considered to be a good thing to have any algorithm.

Consider there are 3 points x_1, x_2, x_3 and a cluster center z_1.
$d(x_1)^2 = 2$, $d(x_2)^2 = 4$ and $d(x_3)^2 = 6$

Probability of x_i being sampled:
$p(x_1) = 2/12 = 0.16$
$p(x_2) = 4/12 = 0.33$
$p(x_3) = 6/12 = 0.5$

Cumulative Distribution:

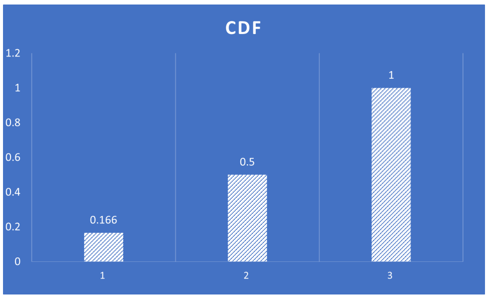

Figure 5: Cumulative Distribution

Perform sampling a value by between $[0, 1]$ and choose corresponding x_i as cluster center.
Example: If sampled value is 0.6, it falls between 0.5 to 1. So, x_3 is considered as cluster center.

5 Comparing Clustering Algorithms

There are a ton of variants and other clustering algorithms which we didn't discussed in this document. But we found a resource gives a glimpse of how each algorithm will perform on different kinds of data. Once, going through that will provoke a curious mind and one should do research on their own understand them better.

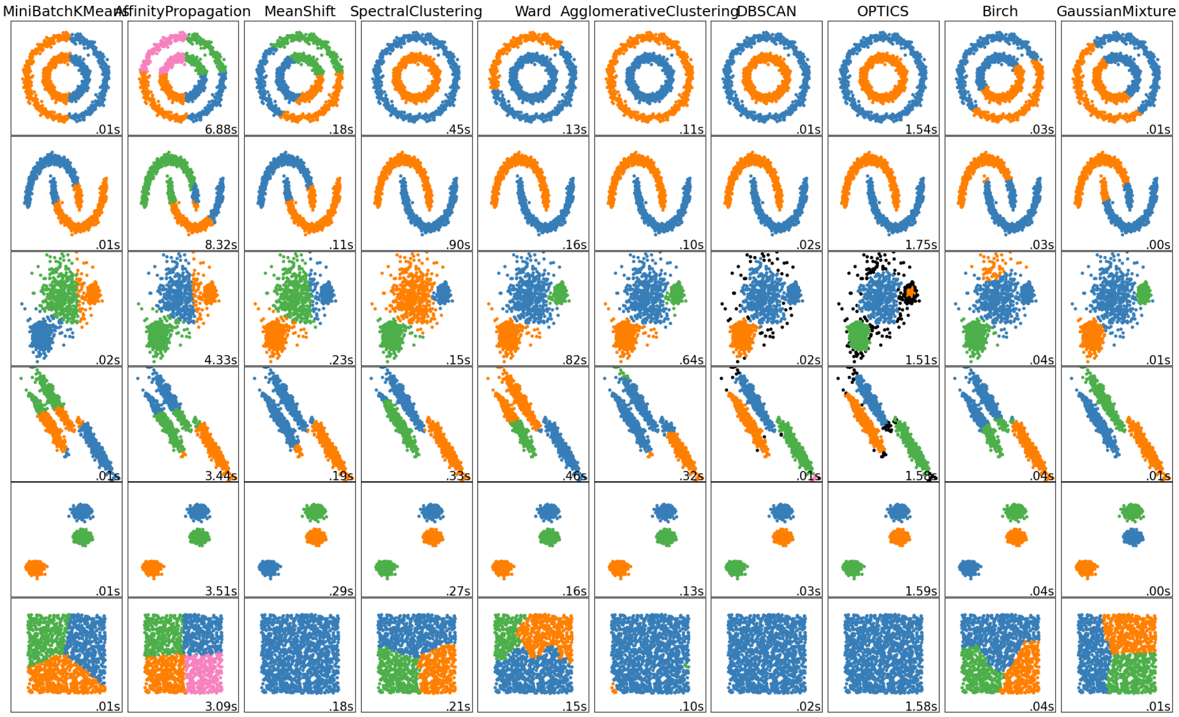

Figure 6: Visualizing different algorithms across different data

Code for above visualization : Jupyter notebook

References

[1] https://medium.com/@darkprogrammerpb/agglomerative-hierarchial-clustering-from-scratch-ec50e14c3826

[2] https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

[3] https://www.techopedia.com/definition/32057/k-means-clustering

[4] https://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-agglomerative-clustering-1.html

[5] https://stanford.edu/ cpiech/cs221/handouts/kmeans.html