Unsupervised Learning – Clustering

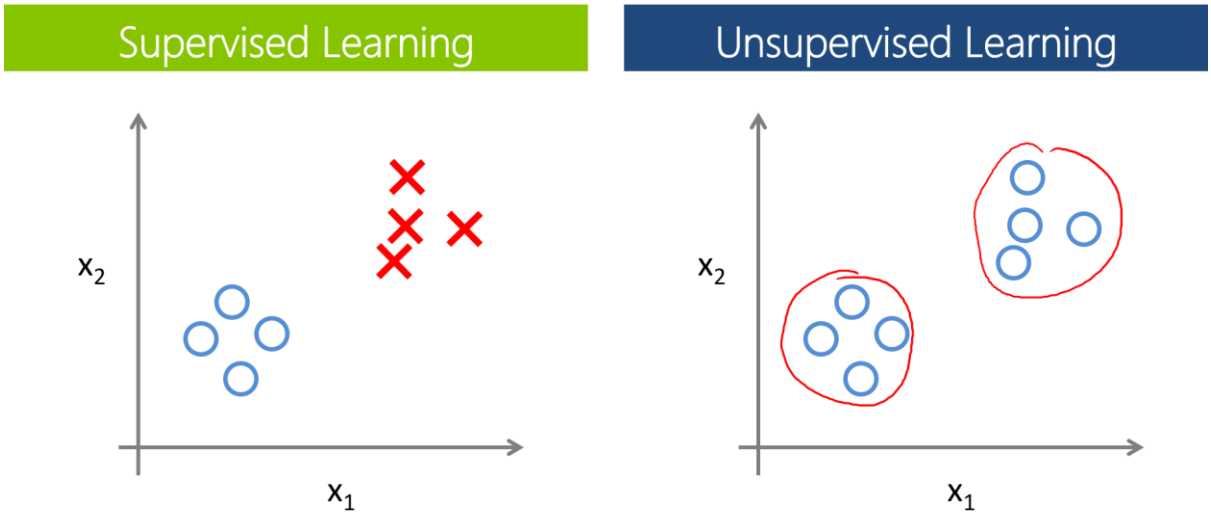
Data Mining and Data Warehousing

Shubhangi Hora

14060321085

Unsupervised Learning

Unsupervised learning is one of the three types of learning (supervised and reinforcement) under machine learning. It is an algorithm wherein patterns and inferences are drawn from input datasets without any labelled output. Due to the fact that there are no responses made available to the algorithm, and it must learn without them, a method known as cluster analysis is the most common unsupervised learning method, and it will be the main topic for this paper.



Cluster Analysis

This method is used to find hidden patterns and grouping in unlabelled data. There are a large variety of clustering algorithms that ultimately do the same thing – finding and forming groups (or clusters) of the data in an efficient manner, so as to be able to predict the cluster that a new data point will belong to. The clusters are formed in such a manner that each cluster consists of data which has the highest measure of similarity when compared with data in other clusters. This measure of similarity can be defined by Euclidian distance, probabilistic distance, etc., on which the clusters are modelled.

Clustering is one of the main tasks of EDM (exploratory data mining) and is used in several fields besides machine learning, such as pattern recognition, image analysis, data compression, and so on.

The algorithms that this paper will look at are:

i. K – means
ii. Expectation – Maximization
iii. Affinity propagation
iv. Mean – shift
v. Spectral clustering
vi. Hierarchical clustering
vii. DBSCAN

K – means clustering

This is one of the most popular and simple clustering algorithms. A plot of unclustered data would be Figure 2, wherein an input of data points, i.e. D = (x_1, x_2, x_3, … , x_n) is given. To form clusters from this data, the K – means algorithm works with the intuition that every cluster has a centre, and the points that are closest to xyz centre belong to xyz cluster. When using this algorithm, the number of clusters is decided ahead of time, and this is known as *K*. So, *K* in K – means algorithm is the number of clusters that the data points will be divided into. The notation of the centres is μ. Hence, *K* clusters have $\mu_1, \mu_2, \mu_3, \dots, \mu_k$ unknown centres. The data points D can be d-dimensional vectors, so, $x_i \in R^d$. Therefore, the centres should also reside in the same space: $\mu_i \in R^d$. The centres μ are basically the mean average of the data points *x* present in that cluster, also known as centroids.

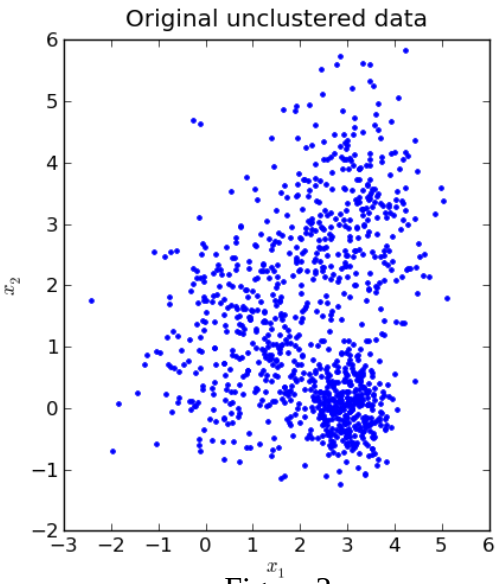


Figure 2

The clusters formed after using the K – means algorithm would be what is shown in Figure 3.

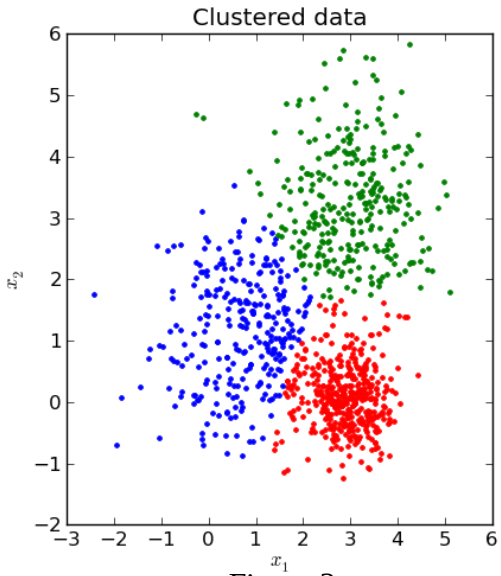


Figure 3

To check the efficiency of the centre in correctly defining a cluster, the distance (Euclidian distance) or the sum of squares between the data points of a cluster and its centre can be calculated, whose generic formula will be $||x_i - \mu_j||^2$. Hence, the total sum of the distances of all the points of a cluster to its centre will just be $\sum_{i:\, xi \text{ is assigned to } j} ||xi - \mu j||^2$. Summing this up for all the clusters will give us $\sum_{j=1}^{K} \sum_{i:\, xi \text{ is assigned to } j} ||xi - \mu j||^2$, which we will call L.

$L = \sum_{j=1}^{K} \sum_{i:\, xi \text{ is assigned to } j} ||xi - \mu j||^2 = \sum_{j=1}^{K} \sum_{i=1}^{n} aij ||xi - \mu j||^2$

where $a_{ij} = \begin{cases} 1 \text{ if } xi \text{ is assigned to } j \\ 0 \text{ if not} \end{cases}$.

So, the process and aim of K – means algorithm are as follows:

Aim –

Try to minimize L with respect to all the *a*s and all the μs (minimizing the within – clusters – sum – of – squares)

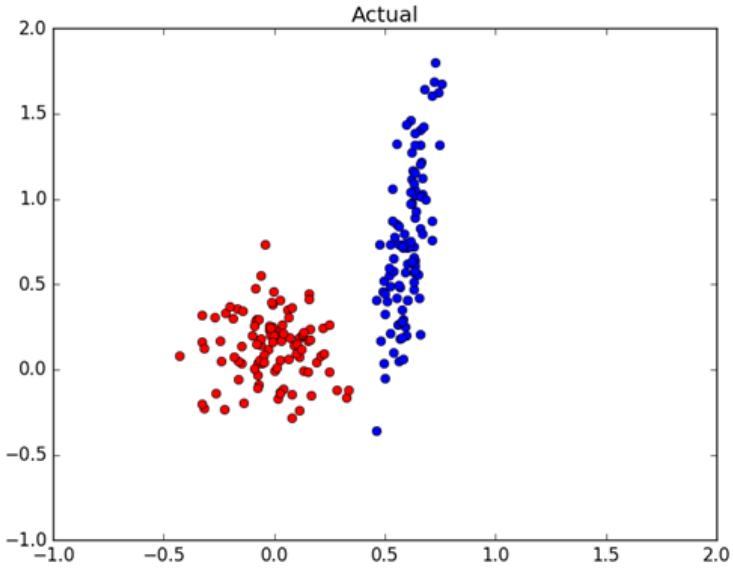
Process –

i. Initialize the centroids ($\mu_1, \ldots, \mu_k$)
ii. Choose the optimal assignments *a* for fixed centroids μ (choose the nearest centroid for that sample)
iii. Choose the optimal centroid μ for assignment *a*
iv. Repeat ii. and iii. until convergence (until things stop changing)

What the above steps means is that, once the centroids have been initialized, the samples are each assigned to a centroid based on which centroid is closest to them (Euclidian distance). Once all the samples have been assigned to a centroid, new centroids are determined based on the mean average of the newly assigned samples. This process is repeated until the samples stay fixed to their cluster and do not move, and the centroid itself does not move significantly either.

Expectation – Maximization algorithm

If you are presented with some unlabelled data that comes from a multi-variate distribution, and have to estimate the means and variance of each distribution, how would you do it? For starters, we could plot a histogram with the data from each dimension to find the means of both distributions, and by looking at the original figure we can tell that the variance of blue's x's must be small, while the red's x's will have a larger variance. However, this isn't a very robust method, which is why the Expectation – Maximization algorithm is used.



This algorithm is used to generate the best possible hypothesis for the distributional parameters of a multi-variate model. The best hypothesis is simply the maximum likelihood hypothesis, which maximizes the probability that this data comes from *K* distributions, each has a mean μ_k, and a variance σ^2_k.

A single modal normal distribution would have the following equations for the hypothesis:

i. Estimated mean = $\mu\sim = (\sum_{i=1}^{N} xi)/N$

ii. Estimated variance = $\sigma^2\sim = (\sum_{i=1}^{N}(xi - \mu\sim)\text{^}2)/N$

When the distribution becomes multi-modal (many maxima), the hypothesis equations become more complicated, since $h = [\mu_1, \mu_2, \mu_3, \dots, \mu_k; \sigma_1^2, \sigma_2^2, \dots, \sigma_k^2]$

Affinity Propagation

This is an algorithm that is based on the concept of message passing between data points, until convergence. The number of clusters doesn't need to be determined before the algorithm runs. Through the passing of messages, the algorithm finds exemplars (they are data points that are representative of the clusters), and so each exemplar corresponds to a cluster. The algorithm also takes an input of a real number which is referred to as a preference, and this preference determines if that data point is suitable or likely to be an exemplar. If the preference is high, then that data point is likely to be an exemplar. Hence, the number of clusters formed at the end depends on the message passing procedure as well as the preference values of different data points.

The process of finding exemplars is iterative, and depends on the messages passed that belong two categories. One is the responsibility category $r(i, k)$ which is the evidence that supports that sample k should be the exemplar for sample i and the second category is the availability $a(i, k)$ which is the evidence that sample i should chose sample k as its exemplar. Hence, if a sample is similar to many samples and is chosen by many samples then it becomes an exemplar for those samples.

Mean Shift

This algorithm works with a smooth density of data points and aims to find blobs or clusters within them. It is a technique used to locate the maxima of a density function and it does this iteratively. Like the K – means algorithm, Mean Shift also uses centroids, in that it repeatedly calculates the mean of the data points within a specific region to determine the centroid. These means are known as candidates, and once they have been found, they enter a post-processing stage wherein duplicates are eliminated to arrive at the final centroids.

Basically, during each iteration, the kernel (a kernel function is one that finds the dot product of two vectors in a feature space, also known as the generalized dot product function) is shifted until convergence, hence the name mean shift. A mean shift vector *m* is computed for each iteration, and it always points in the direction of the maxima. Hence, the entire algorithm focuses on shifting these candidates constantly until they reach the point of highest density.

A candidate for the centroid is determined by the following equation:

$x_i^{t+1} = x_i^t + m(x_i^t)$ where x_i is the candidate, *t* is the iteration number and *m* is the mean shift vector. The candidate it calculated keeping in mind its neighbouring samples within a given distance, noted by $N(x_i)$.

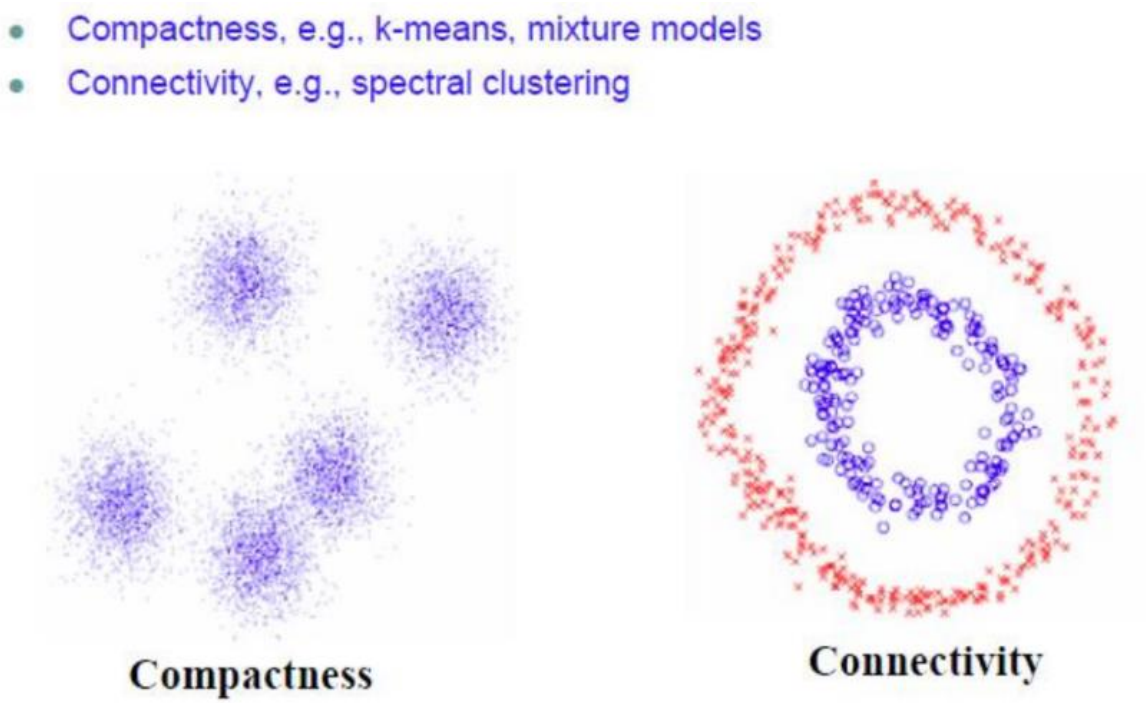
The mean shift vector for each shift is calculated by

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}$$ wherein K is the number of clusters that is automatically set by the algorithm.

Since the calculation of the candidates for centroids depends on the number of samples in their neighbourhood, the algorithm requires a lot of nearest neighbour searches.

Spectral Clustering

This clustering method is also known as the subspace clustering method, since it finds a subspace of the current data points and then performs K – means on that subspace to form clusters. Hence, the main goal of spectral clustering is to cluster data that is connected but not necessarily compact.



This method uses an affinity matrix (also known as a similarity matrix), which determines how close or similar two points in the space are. After this, the eigenvalues of the matrix are found, from which the highest ones are taken to derive the eigenvectors which from the subspace. The K – means clustering method is then applied to this subspace, which forms the clusters. The number of clusters has to be predefined.

Hierarchical Clustering

This is a method that consists of several algorithms that work on the basic premise of building a hierarchy of clusters. There are two broad types:

i. Agglomerative – these algorithms work in bottom up approach, i.e. each observation is its own cluster and then these clusters are merged together as they move up the hierarchy.

   There are three strategies to perform this merge:

   a. Ward linkage – minimizes the sum of the squared differences within the clusters (similar to the K – means method)
   b. Maximum linkage – minimizes the maximum distance between observations within clusters
   c. Average linkage – minimizes the average distances between all observations within clusters

ii. Divisive – these algorithms work in top down approach, i.e. all observations are in one cluster, and splits are formed as they move down the hierarchy.

DBSCAN

DBSCAN stands for Density – Based Spatial Clustering of Applications with Noise. As the name states, it is a clustering algorithm based on the density of data points. Given a set of data points, it clusters those that are compact and close together (points with many neighbours), and marks points that are distant from these clusters and are in low – density regions (points whose nearest neighbours are further away than a given distance) as outliers. Due to groupings based on density, the shape of clusters can be anything, as compared to k – means clusters which are always convex shaped. Hence, DBSCAN forms clusters that consist of core samples that are close to each other, and non – core samples that are the neighbours of those core samples. If there are several points that define a sample as their nearest neighbour, then that sample becomes a core sample of that highly dense region, forming a cluster.

So basically, a cluster formed by the DBSCAN algorithm is made up of a core sample, that checks for its nearest neighbours which also become core samples, which check for their nearest neighbours and also become core samples themselves, until they begin to reach a low density region wherein non – core samples are recognized and the cluster formation stops there. Hence, the non – core samples are always found on the fringes of the cluster.

Comparison of the aforementioned clustering algorithms:

Clustering Algorithm	**Parameters**	**Metric used**
K – Means	Number of clusters	Distance between points
Affinity Propagation	Preference	Nearest neighbour distance
Mean – Shift	Bandwidth	Distance between points
Spectral	Number of clusters	Nearest neighbour distance
Hierarchical	Number of clusters	Distance between points
DBSCAN	Neighbourhood size	Distance between nearest points