Clustering

Mainly for fun

Note: This is a work in progress, and is updated whenever the author feels like it

Eashan Chawla

January 23, 2024

1. Clustering

Clustering, is an unsupervised machine learning technique, that is often used to uncover patterns from data where we have no available training data. Other than the usual classification done on the basis of availability of training data (supervised, unsupervised etc.), techniques in clustering, like any other ML model, can also be classified into 2 broad categories, inductive and transductive.

Inductive models, like k-means clustering, are meant to be trained on a certain set, and inferenced on data the model has never seen before. However, transduction models, are meant to be trained and inferenced on the training data only. Example: Agglomerative clustering.

1.1 K-Means Clustering

Basic algorithm of this is as follows:

1. Initialize cluster centroids for the n clusters (for the first pass, we do random assignment; we will evaluate better way of assignment later).
2. Based on the cluster centroids, assign each point to one of the n clusters. This will be done by measuring the euclidean distance (for now), and assigning the closest centroid's cluster to each point.
3. Based off of these new assignments, we recalculate cluster centroids.
4. Calculate the delta change in each cluster's centroids (in terms of distance). We use this as a threshold to decide if we want to stop early. Repeat steps 2 to 4, till stopping condition is met.

The idea here, is to keep tuning our clusters till we either cross a preset number of maximum iterations, or if the changes to the cluster centroids reach a saturation point, i.e. or don't change too much. However, there's a risk here of us reaching and stopping at a local minimum. One way of avoiding this is to perform multiple iterations of k-means.

K-means is good in the following cases:

- Flat geometries
- Even cluster size (here).
- Spherical, nicely separated clusters.

The above are due to the assumptions we make in this algorithm:

- Even cluster size: If the cluster sizes are uneven, it can create a certain 'pull', pulling smaller cluster centroids towards the larger ones, impacting and possibly degrading clustering. Larger clusters tend to dominate the variance calculation minimization.
- Spherical: This is due to the use of euclidean distances and minimizing the within cluster variance. This by default assumes that the best clusters that we hope to form will be spherical, well defined, because they will more or less be equidistant from the centroid. In cases of non-spherical data, the euclidean distance will not accurately depict the similarity of points within a cluster.

However, it does suffer due to the random nature of cluster initialization impacting where the clustering ends up (already mentioned above).

We want to cover more details and explore the following topics:

- How do we set the best value of n or number of clusters

  - Elbow method
  - Silhouette analysis
- Better starts for k-means / k-means ++

By us calculating the means and recalculating centroids, we implicitly minimize the within cluster variance.

1.2 Density Based Clustering

Some info about Density based clustering

1.2.1 DBSCAN Clustering

This stands for Density Based Spatial Clustering of Applications with Noise.

1.2.2 HDBSCAN Clustering

This stands for Hierarchical Density Based Spatial Clustering of Applications with Noise.