**Introduction**

The purpose of this exercise and related activity is to compare the strengths and weaknesses of a few types of clustering algorithms. We will be looking in detail at:

* K-means clustering
* Mean-shift clustering
* Density-based spatial clustering of applications with noise (DBSCAN)

**K-means clustering**

K-means is a popular example of a centroid-based method. In a general sense, the clusters are formed by partitioning the data points based upon their closeness to each cluster centroid. Mathematically, the algorithm attempts to minimize the coherence, which can be defined as:

where is the mean (or centroid) of points belonging to set . The standard k-means algorithm (naïve k-means) can be implemented as follows:

1. *Initialize*: choose the centroids randomly from the set of data points .
2. *Assignment*: Assign each point to the cluster with the closest (least squared Euclidian distance) centroid.
3. *Update*: Calculate the new means:
4. Check for convergence, which can be determined by the change in coherence:

The advantages of k-means clustering are that it is an easy algorithm to understand and to implement in code, there exist efficient heuristic algorithms (like the standard one outlined above) that converge quickly to a local optimum, and it’s easy to apply to very large data sets.

The major disadvantage of the k-means algorithm is that the data scientist is required to select the number of clusters. This is a significant drawback as one of the primary goals of implementing a clustering algorithm is to gain insight into the data. Choosing the wrong value for will result in poor results.

Another disadvantage is that the algorithm is not guaranteed to converge to a global minimum. Convergence to different local minima based upon the initial selection of centroids can produce clusters that are wildly different. Thus, the results are not consistent or repeatable.

Finally, because points are assigned to clusters based upon their Euclidian distance to the centroids, k-means can only form spherical clusters and the algorithm will generate poor results when the data does not have the same variance in all directions. This also causes the algorithm to be sensitive to outliers, which depending on their distance, can have a large impact on the resulting clusters.

**Mean-shift clustering**

Mean-shift is also a centroid-based method, but it assigns data points to clusters by shifting them toward the local modes, defined as the maxima of the density function.

The modes can be found by building a kernel density estimation (KDE) for the data set. A kernel is essentially a weighting function that is placed at each data point. There are many different types of kernels, but one of the most popular is the Gaussian kernel, and that is what is used in the accompanying activity:

where is the kernel bandwidth.

The mean-shift algorithm starts by creating a copy of the original points. Using the KDE, the copied points are then iteratively shifted uphill against the original points until they reach a peak or local maximum.

The biggest advantage of mean-shift over k-means is that it does not require specifying the number of clusters in advance. Instead, the algorithm is data-driven and determines the number of modes based upon the kernel and the bandwidth. The cluster centers intuitively converge toward points of maximum density to generate a reasonable number of clusters.

Further, the algorithm is model free, meaning that it does not assume any shape for the data clusters. It is also simple. It relies only on the selection of a single parameter, the bandwidth/window size , which has a physical meaning and is easy to understand.

The biggest disadvantage to mean-shift is that it is an N-squared algorithm, so that it is computationally intensive and slow, especially as the number of points grows large. Moreover, the selection of an appropriate is important and not trivial. If is too large, this will result in a wide smooth KDE surface and the merger of modes. At the extreme, there will be only one peak and all the points will be assigned to the same cluster. On the other hand, if is too small, this will generate additional “shallow” modes. At this extreme, there will be a peak for each point and each point will be assigned its own cluster.

**Density-based spatial clustering of applications with noise (DBSCAN)**

DBSCAN is a density-based model that attempts to assign data points to clusters based upon densities present in the space. Essentially, it assigns points to a cluster if there is a certain minimum number of points present within a distance (usually Euclidian distance). Points that are in low-density regions are marked as outliers or noise.

The basic algorithm works as follows:

1. Select an arbitrary point.
2. Find all neighborhood points within a distance . If the number of neighbors is greater than some minimum , then start a new cluster. Otherwise, label the current point as noise and go back to step 1. (Although this point could later become part of another cluster.)
3. Each point in the neighborhood (that has not been previously assigned to another cluster) is visited and assigned to the current cluster. For each of these, find all neighborhood points within . If the number of neighbors is greater than , append each of these points (that has not been previously assigned to another cluster) to the neighborhood and assign to current cluster.
4. Repeat step 3 until all points within the neighborhood have been visited and labeled, and the density connected cluster has been fully discovered.
5. Return to step 1 and repeat until all points have been visited and labeled.

Like all clustering algorithms, the selection of parameters is important. In this case, those parameters are and .

If the value for is too small, then a large part of the data will be labeled as noise. Too large and the clusters will merge, and most of the data could be assigned to the same cluster. One technique for choosing an appropriate value for is to use a k-distance graph, although that analysis is beyond the scope of this exercise. It can be state that in general, smaller values of are preferable.

The parameter can be set according to the dimensions of the data:

Larger values often perform better for larger data sets or for data containing more noise.

Like mean-shift, a primary advantage of the DBSCAN algorithm is that it does not require specifying the number of clusters in advance. An additional advantage of DBSCAN is that it also identifies outliers as noise, whereas the mean-shift algorithm assigns all data to clusters regardless of how far afield a point might be.

Another significant advantage is that DBSCAN can find arbitrarily sized and arbitrarily shaped clusters, even if a cluster is encompassed, but not density-connected to another.

The primary disadvantage of DBSCAN is that it will perform poorly if the underlying clusters are of varying density. Also, the algorithm is also not entirely deterministic. Border points that are accessible to multiple clusters will be assigned based upon the order in which the data is processed. Although it should be noted that the interchangeability of some border points is trivial compared to the discrepancies that arise in runs of k-means due to the initial selection of the cluster centroids.