Q1.

K-means is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid.

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

The centroids have stabilized — there is no change in their values because the clustering has been successful.

The defined number of iterations has been achieved.

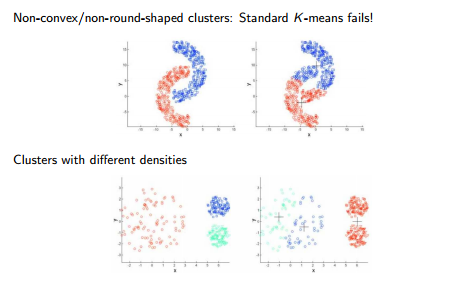
### OBSERVATIONS

For the first and second situation we have changed the distance metric only and not the centroid. We observe that even though manhattan converges in less counts but has more SSE as compared to Euclidean.

But after keeping the distance metric fixed as Manhattan and if the initial centroids are positioned more closely to the actual centroids then it performs better with lower SSE.

Q2.

K-Means clustering algorithm fails to give good results when the data contains outliers, the density spread of data points across the data space is different and the data points follow non-convex shapes.



1. On comparing the SSE’s of Jaccard, Euclidean and cosine similarity, **Cosine>Jaccard>Euclidean** , which means the Euclidean has the least SSE(sum of Squared errors) and hence it is the best.

Cosine similarity takes a unit length vector to calculate dot products i.e(angle). Even though the cosine similarity has lower SSE , it tends to give more accuracy. This is because the cosine also requires normalization to perform better also cosine is inaccurate if the points are collinear.

1. After observing the accuracies of **Cosine > Euclidean > Jaccard**. The Cosine has the highest accuracy with 97.3 followed by Euclidean with 88.667 and Jaccard (88.000).

The cosine measures the difference between two points on the basis of angle between vectors whereas Manhattan and Euclidean consider the distances therefore cosine performs more accurately.

When plotted on a multi-dimensional space, where each dimension corresponds to a word in the document, the cosine similarity captures the orientation (the angle) of the documents and not the magnitude. If you want the magnitude, compute the Euclidean distance instead.

The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance because of the size (like, the word ‘cricket’ appeared 50 times in one document and 10 times in another) they could still have a smaller angle between them. Smaller the angle, higher the similarity.

1. The Euclidean requires 8 iterations whereas Jaccard has a count of 10 iterations and **cosine** performs the **best** requiring only 7 iterations and giving the best accuracy as well.

Distance between 2 close points from one cluster is the same as between 2 remote points from different clusters within the standard error. This is one aspect of the "**curse of dimensionality**" resulting in inaccurate clustering by Euclidean and Manhattan .

Possible explanation why cosine performs better –

While performing K-means clustering, the metric used is Euclidean distance, since points may be in opposite directions but they may fall into the same cluster, if the distance of both points from the centroid is the same. Thus it converges quicker and takes less time.

1. Table 1-

|  |  |  |  |
| --- | --- | --- | --- |
| SSE | Centroids | SSE | Max Iteration =100 |
| Euclidean | 78.945 | 244.220 | 78.945 |
| Cosine | 92.078 | 250.34 | 92.078 |
| Jaccard | 79.186 | 247.47 | 79.186 |

1. Table 2-

|  |  |  |  |
| --- | --- | --- | --- |
| Counts | Centroids | SSE | Max Iteration =100 |
| Euclidean | 8 | 1 | 100 |
| Cosine | 7 | 1 | 100 |
| Jaccard | 10 | 1 | 100 |

1. Table 3-

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy | Centroids | SSE | Max Iteration =100 |
| Euclidean | 0.886 | 0.72 | 0.86 |
| Cosine | 0.9733 | 0.793 | 0.9733 |
| Jaccard | 0.880 | 0.753 | 0.88 |

The algorithm converges quickly for sse and centroid as compared to max iterations topping criteria, because it is very likely to converge and perform more accurately by just looking at centroid if it is static or not.

On the basis of SSE centroid and max iteration perform best.

Whereas in terms of count sse takes least count whereas max iteration takes the most number of iteration i.e 100.

Comparing accuracies centroids performs most accurately and with fewer counts.

The basic idea of k-means is to minimize squared errors. There is no "distance" involved here.K-Means procedure - which is a vector quantization method often used as a clustering method - does not explicitly use pairwise distances between data points at all (in contrast to hierarchical and some other clustering which allow for arbitrary proximity measure).

It amounts to repeatedly assigning points to the closest centroid thereby using Euclidean distance from data points to a centroid. However, K-Means is implicitly based on pairwise Euclidean distances between data points, because the sum of squared deviations from centroid is equal to the sum of pairwise squared Euclidean distances divided by the number of points.

The "centroid" is multivariate mean in euclidean space. Euclidean space is about euclidean distances. Non-Euclidean distances will generally not span Euclidean space. That's why K-Means has less SSE for Euclidean distances only.

Q3.

## Advantages

Easy to implement

With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).

k-Means may produce tighter clusters than hierarchical clustering

An instance can change cluster (move to another cluster) when the centroids are recomputed.

## Disadvantages

Difficult to predict the number of clusters (K-Value)

Initial seeds have a strong impact on the final results

The order of the data has an impact on the final results

Sensitive to scale: rescaling your datasets (normalization or standardization) will completely change results. While this itself is not bad, not realizing that you have to spend extra attention to scaling your data might be bad.

Other strategy -

* The elbow method is used to determine the optimal number of clusters in k-means clustering. The elbow method plots the value of the cost function produced by different values of k. It runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters. By default, the distortion score is computed, the sum of square distances from each point to its assigned center.
* The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.
* It compares the total within intra cluster variation for different values of k with their expected values under null reference distribution of data. The estimate of the optimal clusters will be value that maximizes the gap statistic.

The codes are available here - <https://github.com/sidmal11/ml/tree/master/Assign6>