CA Assignment 2

Data Clustering

(Implementing clustering algorithms)

ASSIGNMENT REPORT SUBMITTED FOR ASSSESMENT OF COMP527 (Data Mining and Visualisation)

BY
SANTHOSH ARUNAGIRI
201586816

UNIVERSITY OF
LIVERPOOL

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF LIVERPOOL
LIVERPOOL L69 3BX

K-MEANS CLUSTERING

K-means clustering is an unsupervised learning technique to separate data using cluster method. The method of k-means clustering is to separate and group a set of data points into ‘k’ clusters, where k is the number of clusters.

This algorithm selects the first chooses a data point randomly from the dataset and makes it the initial centroid for all clusters. And then the algorithm is iterated till all the data points are assigned to the nearest cluster (Euclidean distance). After all the data is assigned to their nearest clusters, the centroids are recalculated as the mean of all the data points in each cluster. This process is repeated until the centroids converge, and no data point changes its cluster assignment.

PSEUDO CODE:

Step-1 – Initialization

- Initialize n as number of datapoints and m as number of features in the dataset.
- Choose k number of random data points in the dataset as a centroid.
- create an array of n zeros called labels to store the assigned cluster for each data point.

Step-2 – Assigning datapoints to clusters.

- Calculate the Euclidean distance from data point to each centroid.
- Assign all the data points to the nearest cluster using Euclidean distance.

Step-3 - optimisation

- calculate the new centroid location using the mean of the data points assigned to the cluster.
- Check for convergence by comparing centroids to new centroids.

```
    if the centroids are close to the new centroids:
        break
    centroids = new centroids
return centroids, labels.
```

K- MEANS ++ CLUSTERING

K-means clustering is an unsupervised learning technique to separate data using cluster method. The method of k-means clustering is to separate and group a set of data points into ‘k’ clusters, where k is the number of clusters. Just like the k means clustering but differs in initialising process.

K-means++ clustering algorithm chooses its first centroid randomly and selects the subsequent centroid by calculating the distance between them. Specifically, each new centroid is chosen with a probability proportional to its squared distance from the nearest existing centroid. This ensures that the centroids are well-spaced and increases the likelihood of finding a globally optimal solution.

PSEUDO CODE:

Step-1 – Initialization

- Initialize n as number of datapoints and m as number of features in the dataset.
- create a k x m array of zeros called centroids.
- the first centroid is chosen randomly from the dataset.

Step-2 – Assigning datapoints to clusters.

- calculate the distances between each data point and the j-1 centroids using squared Euclidean distance.
- calculate the probability of each data point being chosen as the next centroid using the formula:
  probability = distances / sum of distances
- randomly choose the index of the next centroid from the dataset using the probability calculation and assign its feature values to the j-th row of centroids.

Step-3 – optimisation

- For each iteration in range(iterations):
  - For each data point i in range(n):
  - calculate the squared Euclidean distance between data point i and each centroid.
  - assign the data point to the cluster of the nearest centroid using the formula: labels[i] = argmin_j distances[j]
- For each cluster j in range(k):
  - calculate the new centroid location using the mean of the data points assigned to the cluster using the formula:
    new centroids [j, :] = sum of the feature values for all data points/number of data points
- Check for the convergence by comparing centroids to new centroids using the function np.allclose()
  - if the assigned centroids and new centroids are close, break out of the loop.
    else, update centroids to new centroids
- Return the updated centroids and assigned labels for each data point.

BISECTING K-MEANS CLUSTERING

Bisecting k-means is a hierarchical clustering algorithm. This method separates the data into clusters on iteration by repeatedly bisecting the cluster with the largest sum of squared errors (SSE).

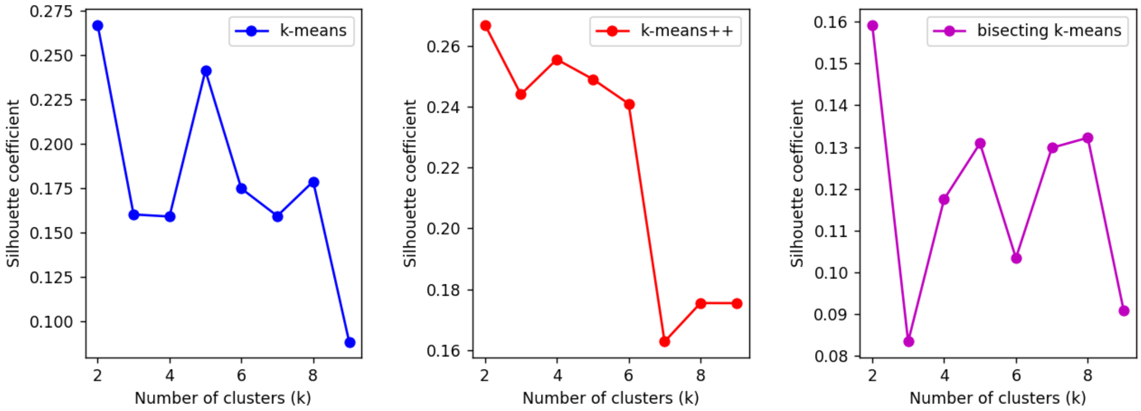
At first all the datapoints are assigned to a same cluster then using the SSE it selects the clusters with larger value and bisects them into two clusters. It then implements the previously explained k means algorithm on the selected cluster to separate them into two smaller clusters. This process is repeated till the k number of clusters are obtained.

Bisecting K-means is a popular algorithm for clustering large datasets for its scalability and handling of non-linearly separable clusters.

PSEUDO CODE:

```
• Get the shape of the dataset and set the number of clusters (k) and number of iterations.
• A list of cluster indices with all data points is created
• Calculate the mean of the dataset and set it as the centroid for the initial cluster.

Repeat until k (1-9) clusters have been formed.

    • Choose the cluster with the maximum (SSE).
    • Split the chosen cluster into two new clusters using k-means algorithm.
    • Update the cluster indices and centroids with the new clusters.
    • Repeat the process until the k clusters are formed.

Assign each data point to the closest cluster using the final centroids.

    • Return the final centroids and cluster labels.
```

OBSERVATION



CONCLUSION

Based on the results of the algorithm it is identified that the k=2 shows better value for K-means, K-means++ and bisecting k means. we can observe the value dropping after k=3 significantly. suggesting that two clusters provide a good separation of the data.

But on comparison with the methods K-means and K-means++ are relatively higher than the bisecting K-means. And k-means++ has a gradually decreasing graph. This suggests that these two algorithms are better and k-means++ is much more suited for this dataset.

Though we find these results. They are just being observed to be better comparative basis, on the whole all these clustering methods are generally low. As they are very much less in comparison to the perfect cluster value of 1. So, we need to explore other clustering methods to find a better result.

REFERENCES

1) Dabbura, I. (2018). K-means clustering: Algorithm, applications, evaluation methods, and drawbacks. [online] Medium. Available at: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.
2) www.youtube.com. (n.d.). StatQuest: K-means clustering. [online] Available at: https://youtu.be/4b5d3muPQmA.
3) Analytics Vidhya. (2021). Understanding K-Means Clustering Algorithm. [online] Available at: https://www.analyticsvidhya.com/blog/2021/11/understanding-k-means-clustering-in-machine-learningwith-examples/.
4) Simplilearn.com. (n.d.). K-Means Clustering Algorithm. [online] Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm.
5) Kumar, S. (2021). Understanding K-means, K-means++ and, K-medoids Clustering Algorithms. [online] Medium. Available at: https://towardsdatascience.com/understanding-k-means-k-means-and-k-medoids-clustering-algorithms-ad9c9fbf47ca#:~:text=K%2DMeans%20algorithm%20is%20a [Accessed 24 Mar. 2023].