COSC 757 Data Mining Assignment 3

Mary Snyder
Department of Computer & Information Sciences
College of Science & Mathematics
Towson University
msnyde8@students.towson.edu

ABSTRACT
In this paper, I will be exploring a dataset to become more familiar with data clustering through the COSC 757 Data Mining Assignment 3.

Categories and Subject Descriptors
H.2.8 **[Database Management]** Database Applications – *Data mining*

Keywords
Clustering; Multivariate; Cluster Analysis; Partitioning Approach; k-means; k-medoids; Hierarchical Approach; AGNES; Single Linkage; Complete Linkage; Density-based Approach; Density-Based Spatial Clustering of Applications with Noise (DBSCAN); Partitioning Around Medoids (PAM); Silhouette Plot; Dendrogram; Agglomerative Coefficient

1. INTRODUCTION
1.1 Dataset
I chose a dataset from the UCI Machine Learning Repository classified for the task of Clustering. This AAAI 2013 Accepted Papers dataset comprises the metadata for all the main track only papers accepted for the 2013 AAAI conference. The dataset contains information regarding the paper's title, abstract, and keywords of varying granularity. There are 150 instances with no missing values and contains 5 attributes: Title, Keywords, Topics, High-Level Keyword(s), and Abstract. The Topics and High-Level Keywords attributes were used for clustering.

1.2 Objective of Analysis
The objective of clustering, or cluster analysis, is to find similarities among the data between the characteristics found in the data and using those similarities to group the data into clusters. Many times, it is used to gain insight into data distribution, but it may also be used as a pre-processing step for other algorithms.

Good cluster methods produce high quality clusters with high intra-class similarity or cohesion within clusters. They also have low inter-class similarity or distinction between clusters. The quality of the method depends on its measure of similarity, the implementation, as well as how well it is able to discover hidden patterns within the data. Similarity/dissimilarity is expressed in terms of distance, while quality is more subjective.

2. METHODOLOGY
2.1 Preprocessing
The dataset contains two different keyword attributes as well as the title and abstract information (see Figure 1). Since in most cases the title would be a unique value, I eliminated it as a possible clustering attribute. Similarly, the abstract contained a description of the paper that would mostly likely be unique, so it was eliminated from the clustering attribute selection as well. There are two keyword attributes, which varied in their degree of granularity, one more simplistic (Keywords) and one more categorical (High-Level Keyword(s)). I chose to eliminate the more simplistic Keywords attribute in favor of the more categorical High-Level Keyword(s) in hope this would produce better clustering results. The remaining attribute Topics also seemed fairly categorized, so I chose to pair it with the High-Level Keyword(s) for the analysis (see Figure 2).

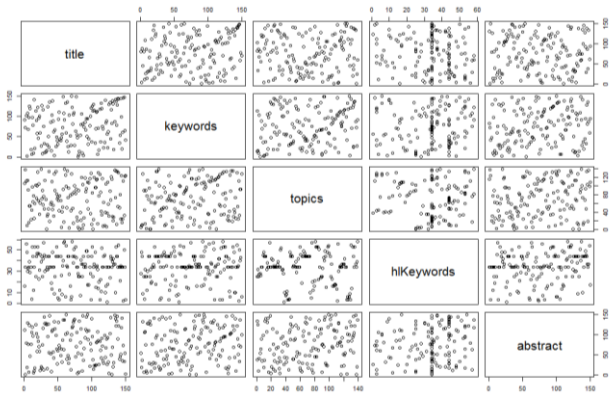

Figure 1. AAAI 2014 Accepted Papers Dataset Attributes

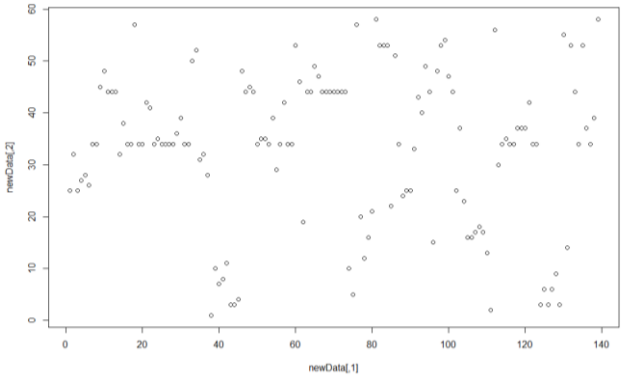

Figure 2. Topics and High-Level Keyword(s) for Clustering

2.2 Clustering Approaches
2.2.1 Partitioning Approach
The partitioning approach toe clustering constructs various partitions and then evaluated them by some criterion. The data set is divided, or partitioned, into *k* clusters, so as to optimize the chosen partitioning criterion and such that the sum of squared distances in minimized or

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - c_i)^2$$

2.2.1.1 K-means
K-Means in a partitioning approach to clustering in which each cluster is represented by the center of the cluster. There are four steps for the k-means algorithm:

1. Partition the dataset into *k* nonempty subsets
2. Compute the seed points as the mean point or centroid of the clusters current partitioning
3. Assign each object to the cluster with the nearest seed point
4. Repeat from step 2 until the assignments do not change

K-means is often considered a greedy algorithm, but it is efficient running at $O(tkn)$ where n is the number of instances, k is the number of clusters, and t is the number of iterations. K-means also has a few weaknesses including the number of clusters k needs to be specified ahead of time, it is sensitive to noisy data and outliers, and it can only be applied to objects in a continuous n-dimensional space.

2.2.1.2 K-medoids

K-Medoids is a partitioning approach to clustering in which each cluster is represented by one of the objects in the cluster. It is similar to the k-means approach, but instead of taking the mean value of the object cluster as the seed point, medoids, or the most centrally located object in a cluster, are used. In addition, while k-means only applies to objects in a continuous n-dimensional space, k-medoids can be applied to a wide range of data. K-medoids again suffers from weaknesses similar to k-means such as the number of clusters k must be specified ahead of time. K-medoids also does not scale well for large datasets due to the computational complexity of the algorithm.

The Partitioning Around Medoids (PAM) algorithm is used for k-medoids clustering. The algorithm works similarly to k-means except for the reassessment, which is completed as follows:

> Start from the initial set of medoid and iteratively replace one of the medoids with one of the non-medoids to determine if it improves the total distance of the resulting cluster

2.2.2 Hierarchical Approach

The hierarchical approach to clustering decomposes the set of data into a hierarchy using some criterion. It uses a distance matrix as the clustering criteria. Unlike the k-means and k-medoids partitioning approaches, this method does not require the number of clusters k to be provided; however, it does need a termination condition to be specified.

The hierarchical approach is not without its weaknesses. In creating the hierarchy, a previous step can never be undone. The approach also has a time complexity of $O(n^2)$, where n is the total number of objects, so the approach does not scale well.

2.2.2.1 AGNES

Agglomerative Nesting (AGNES) uses linkage and a dissimilarity matrix to cluster data. Nodes with the highest/lowest (depending on the linkage) criteria of dissimilarity are combined into a cluster, progressing in an ascending fashion, with all nodes eventually in one cluster.

2.2.2.1.1 Single Linkage

Single linkage clustering uses the smallest distance between an element in one cluster and an element in another:

$$dist(K_i, K_j) = min(t_{ip}, t_{jq})$$

2.2.2.1.2 Complete Linkage

Complete linkage clustering uses the largest distance between an element in one cluster and an element in another:

$$dist(K_i, K_j) = max(t_{ip}, t_{jq})$$

2.2.3 Density-based Approach

The density-based approach to clustering is based on specified connectivity and density functions. Some of the advantages to density-based approach over other approaches is it can handle noise, cluster discovered can be of arbitrary shape, and only one scan of the data is needed.

Density-based clustering work with two parameters:

> Eps: Maximum radius of the neighborhood
>
> MinPts: Minimum number of points in an Eps-neighborhood (N_{Eps}) of the point

Density-based approach also uses the concepts of density-reachable and density-connected. A point p is defined as density-reachable from a point q if there is a chain points such that p_{i+1} is directly density-reachable from p_i. A point p is defined as density-connected to a point q if there is a point o such that both, p and q are density-reachable from o.

2.2.3.1 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clusters data into maximal sets of density-connected points. In spatial databases with noise, clusters discovered through DBSCAN will be of arbitrary shape. The DBSCAN algorithm is as follows:

> Arbitrarily select a point p
>
> Retrieve all points density-reachable from p w.r.t. *Eps* and *MinPts*
>
> If p is a core point, a cluster is formed
>
> If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the dataset
>
> Continue until all of the points have been processed

3. RESULTS

3.1 Partitioning Approaches

For both partitioning methods, a k value for the number of partitions needs to be specified ahead of time. To determine the best value I used the Elbow method, which looks at the sum of squared error (SSE) within groups as a function of the number of clusters (see Figure 3). Looking for the bend of elbow in the plot gives a good indication of a value for k. In this case, 8 or 10 clusters seemed to be good bend/elbow locations so I used both those values for k in my analysis.

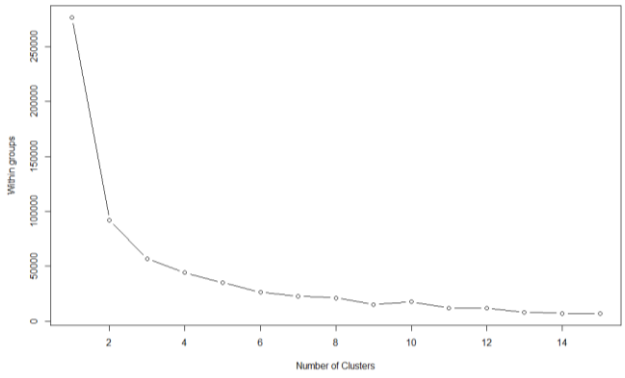

Figure 3. Number of Clusters to Determine Best *k* Value

3.1.1 *K-means*

Using the *k* values obtained from the elbow method, I ran the k-means clustering on the data. The data clustering for *k*=8 is almost perfect (see Figure 4) with no visible outliers.

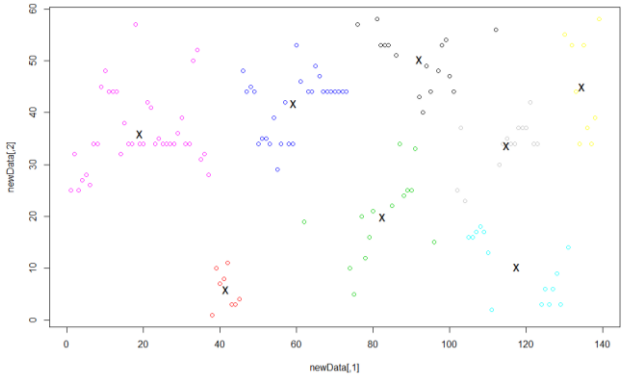

Figure 4. K-means for *k*=8

Running k-means clustering on the data for *k*=10 produced less great of results than with *k*=8 (see Figure 5) with a few possible visible outliers.

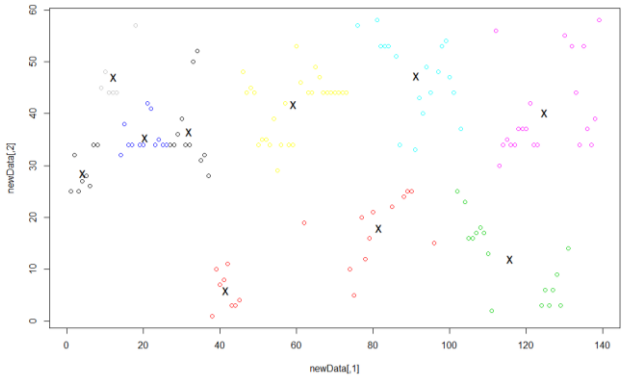

Figure 5. K-means for *k*=10

3.1.2 *K-medoids*

Using the *k* values obtained from the elbow method, I ran the k-medoids clustering on the data. The data clustering for *k*=8 is again fairly well suited (see Figure 6), but with possible outliers.

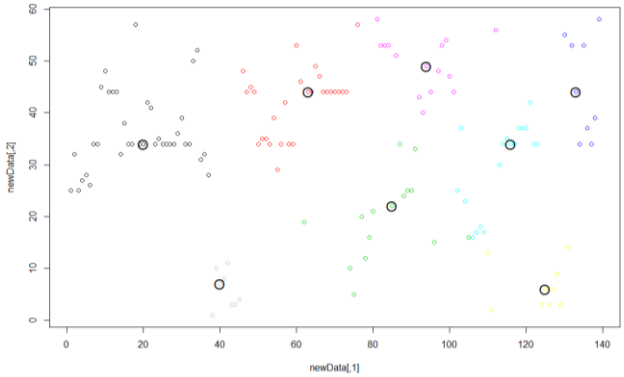

Figure 6. K-medoids for *k*=8

Running k-medoids clustering on the data for *k*=10 produced similar results to k-medoids for *k*=8 (see Figure 7), but again there were possible outliers.

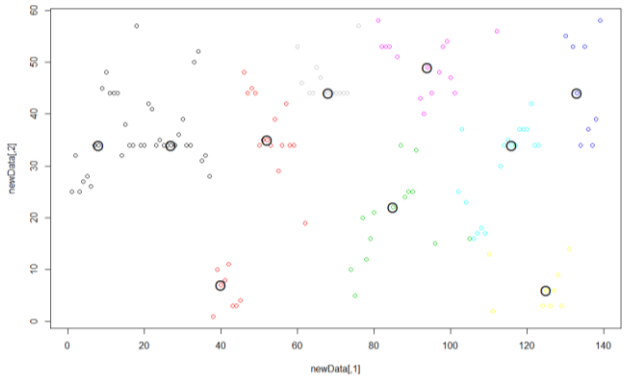

Figure 7. K-medoids for *k*=10

3.2 Hierarchical Approaches

3.2.1 *Single Linkage*

The single linkage clustering on the data did not take very many steps per data point. The dendrogram of the resulting clustering shows the max was barely over 15 steps and most of the data points were clustered in under 5 steps.

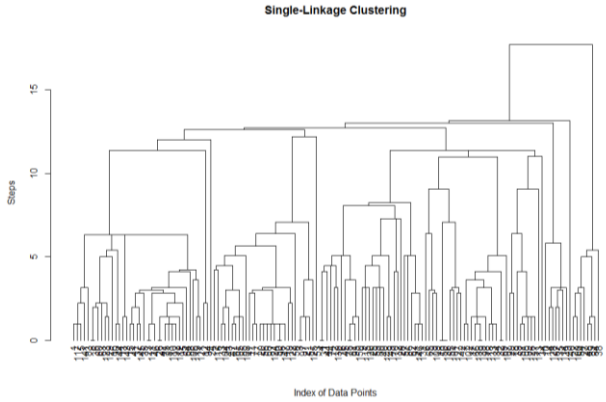

Figure 8. Single-Linkage Clustering

3.2.2 *Complete Linkage*

The complete linkage clustering on the data took many more steps than the single linkage clustering. The dendrogram shows that many of the data points took less than 20 steps, but this was still magnitudes larger than with the single linkage. The complete linkage dendrogram did appear more balanced in its clustering, but that does not mean it was more efficient or effective.

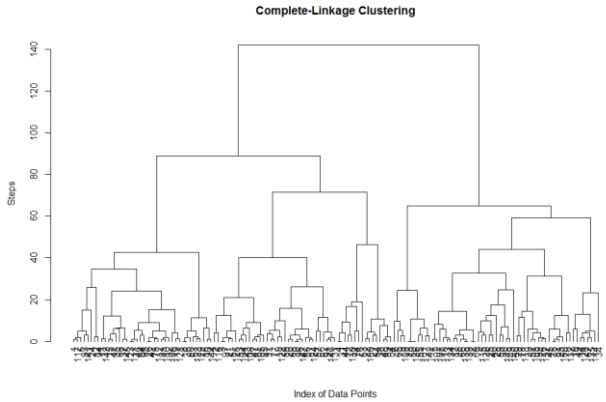

Figure 9. Complete-Linkage Clustering

3.3 Density-based Approach

3.3.1 DBSCAN

I first ran the DBSCAN with an Eps value of 5. The results (see Figure 10. DBSCAN Clustering for Eps=5Figure 10) showed some clustering but still many outliers. I then re-ran the DBSCAN with an Eps value of 10. The results (see Figure 11) were much improved; however, there were still a few outliers. I decided to see if it was possible to eliminate the remaining outliers. To do this I again increased the Eps value to 15 and re-ran the test. This result (see Figure 12) was not well clustered at all, so I knew I needed a Eps value closer to the last good clustering or Eps of 10. I tried a few other Eps values (see results for Eps 11 in Figure 13 and Eps 11.5 in Figure 14) until I found the results I considered the best for Eps value of 11.1 (see Figure 15) with as few outliers as possible.

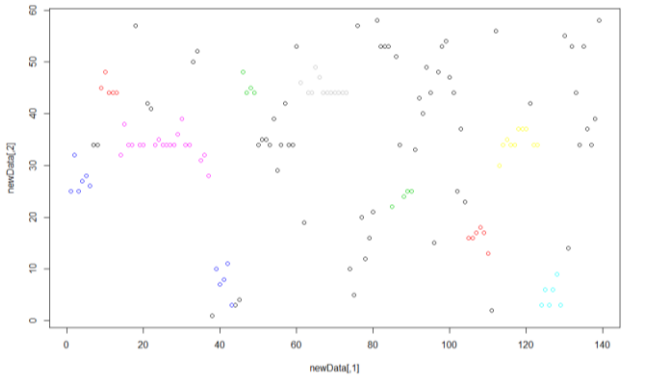

Figure 10. DBSCAN Clustering for Eps=5

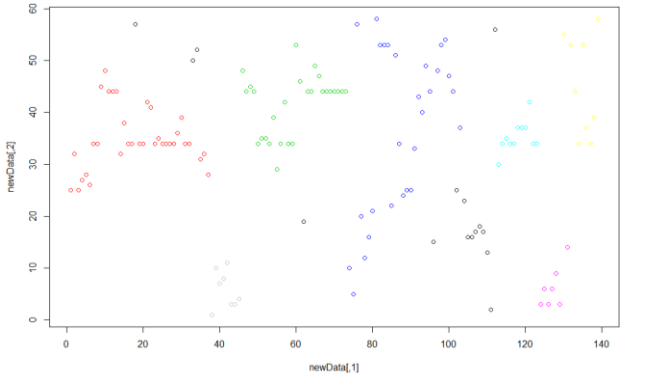

Figure 11. DBSCAN Clustering for Eps=10

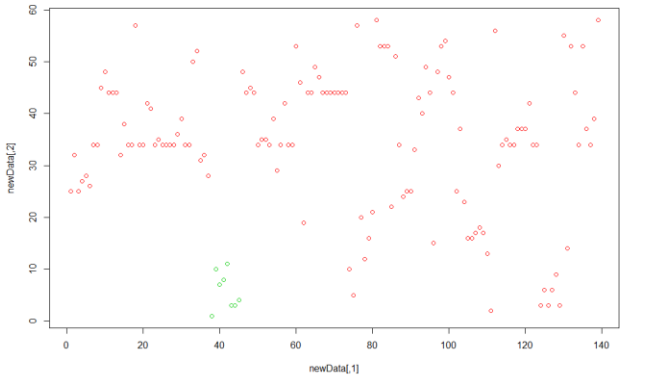

Figure 12. DBSCAN Clustering for Eps=15

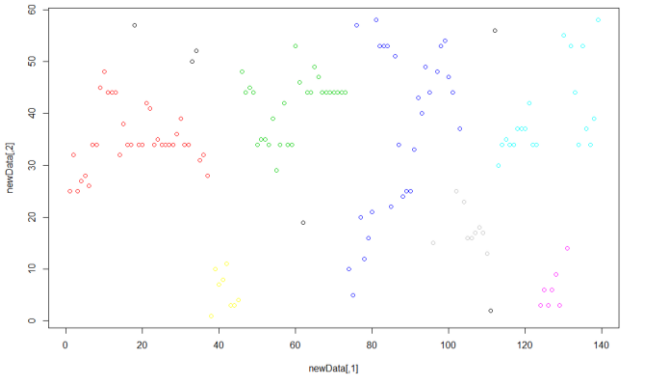

Figure 13. DBSCAN Clustering for Eps=11

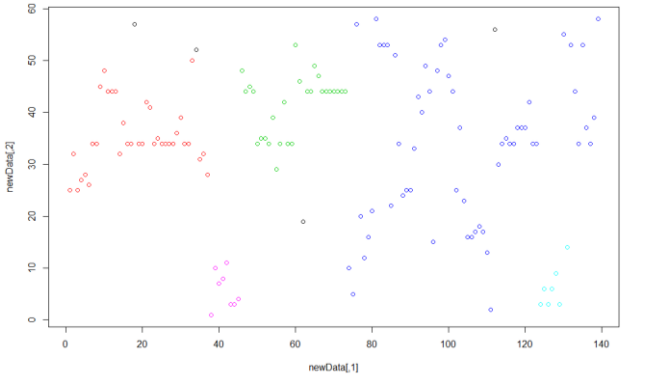

Figure 14. DBSCAN Clustering for Eps=11.5

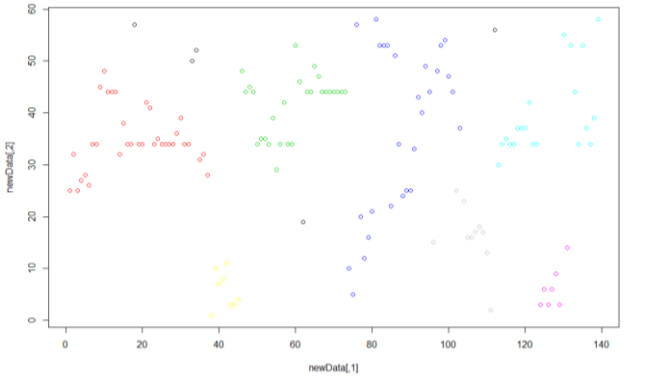

Figure 15. DBSCAN Clustering for Eps=11.1

4. CONCLUSIONS

4.1 Evaluation Metrics

4.1.1 Silhouette Plot

A Silhouette plot shows the following information for each cluster:

- The number of plots per cluster
- The mean similarity of each plot to its own cluster minus the mean similarity to the next most similar cluster
- The average silhouetted width

Large positive Silhouette widths indicate plots that fit well within their cluster, while a small positive or a negative Silhouette indicates the plot fits poorly within their cluster.

4.1.2 Dendrogram

A Dendrogram is a tree-like plot depicting the agglomeration sequence. One axis is an enumeration or identification of an entity and the other axis is the dissimilarity level at which fusion of clusters occurred. It shows the process by which the hierarchical approaches used here (single linkage and complete linkage) clustered the dataset.

4.1.3 Agglomerative Coefficient

The Agglomerative Coefficient (AC) is computed by the AGNES clustering method. The AC measures the clustering structure of the data set and can be defined as follows:

For each observation i, denote by *m*(i) its dissimilarity to the first cluster it is merged with, divided by the dissimilarity of the merger in the final step of the algorithm. The Average of all 1 – *m*(i) is the AC.

4.2 Partitioning Approaches

4.2.1 K-Means

Using *k=8* for the k-means partitioning approach gave the highest average silhouette width (see Figure 16) of the partitioning approaches as well as density-based approaches at 0.53. The silhouette plot shows no outliers and seems to fit the data best.

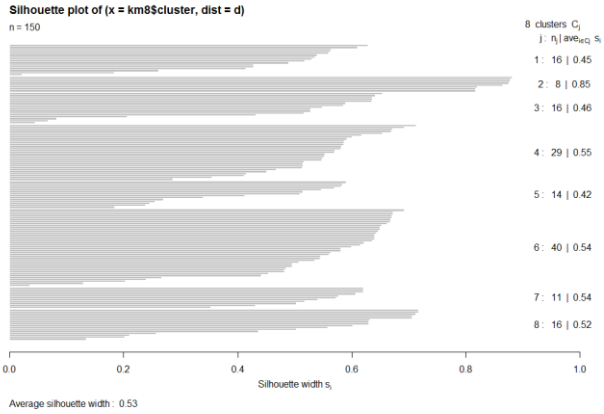

Figure 16. K-Means Silhouette Plot for *k*=8

Using *k*=10 for the k-means partitioning approach gave a slightly lower value for the silhouette width than with *k*=8, but still not bad at 0.49 (see Figure 17). The silhouette plot shows an outlier, but still seems to fit the data fairly well.

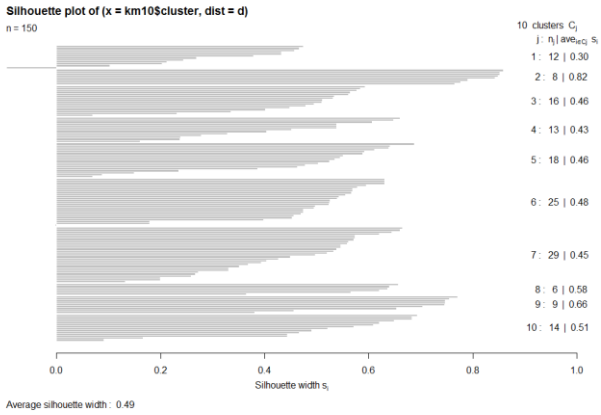

Figure 17. K-Means Silhouette Plot for *k=10*

Statistics for the both *k* values used in the k-means approach show similar average between and within values (see Figure 18). A *k* value of 8 had slightly higher results for both average between as well as average within values.

```
> cstatskm8$average.between
[1] 59.47227
> cstatskm8$average.within
[1] 14.32746
> cstatskm10$average.between
[1] 57.70776
> cstatskm10$average.within
[1] 12.90444
```

Figure 18. K-Means Statistics

4.2.2 K-Medoids

Using *k*=8 for the k-medoids partitioning approach gave the best average silhouette width for the k-medoids method, but still slightly lower than the k-means for k=8. The value for the silhouette width was 0.52 and the silhouette plot shows outliers (see Figure 19).

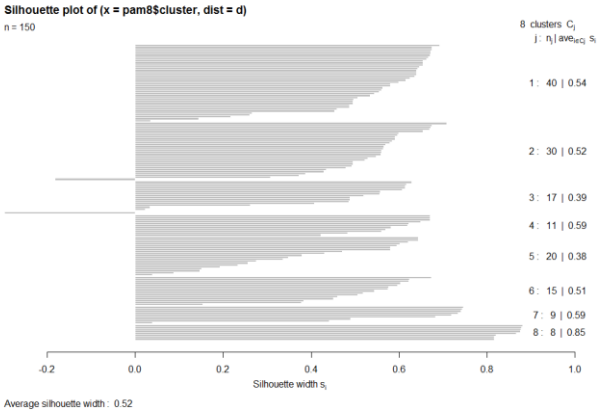

Figure 19. K-Medoids Silhouette Plot for *k*=8

Using *k*=10 for the k-medoids partitioning approach gave a slightly lower value for the silhouette width than with *k*=8, but still not bad at 0.49 (see Figure 20). The silhouette plot shows an outlier, but still seems to fit the data fairly well.

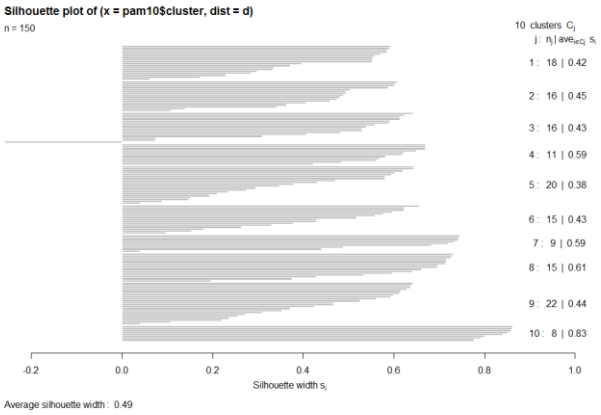

Figure 20. K-Medoids Silhouette Plot for *k*=10

Statistics for the both *k* values used in the k-medoids approach show similar average between and within values (see Figure 21). A *k* value of 8 had slightly higher results for both average between as well as average within values, with more difference between the values for each *k* value than using the k-means approach.

```
> cstatspam8$average.between
[1] 59.64209
> cstatspam8$average.within
[1] 14.5731
> cstatspam10$average.between
[1] 57.22246
> cstatspam10$average.within
[1] 11.28422
```

Figure 21. K-Medoids Statistics

4.3 Hierarchical Approach

4.3.1 Single Linkage

For single linkage, the agglomerative coefficient (AC) was used to measure how well the clustering portrays the original data structure. Values over 0.75 are generally considered to be good, so with a 0.87 (see Figure 22), this approached worked well with this dataset.

```
> agn$ac
[1] 0.8706369
```

Figure 22. Agglomerative Coefficient for Single Linkage

4.3.2 Complete Linkage

For complete linkage, the AC again was used to measure how well the clustering portrays the original data structure. As stated previously, values over 0.75 are considered good and the results on this dataset were ~0.98 (see Figure 23). This approached worked even better than the single linkage with this dataset.

```
> agn2$ac
[1] 0.9797554
```

Figure 23. Agglomerative Coefficient for Complete Linkage

4.4 Density-based Approach

4.4.1 DBSCAN

The silhouette plot for DBSCAN with Eps of 11.1 shows even with this value there were still a few outliers. In addition, the average silhouette width is lower than either of the *k* values used in both the k-means and the k-medoids approaches. So for this dataset, the density-based approach did not do as well as the either of the partitioning approaches. This may be the case because the clusters determined by the partitioning approaches are larger to encompass all the data, but less not very dense especially at the edges. This would lead to the nodes near the edges not being included by DBSCAN and therefore there would be more outliers in the resulting DBSCAN clustering.

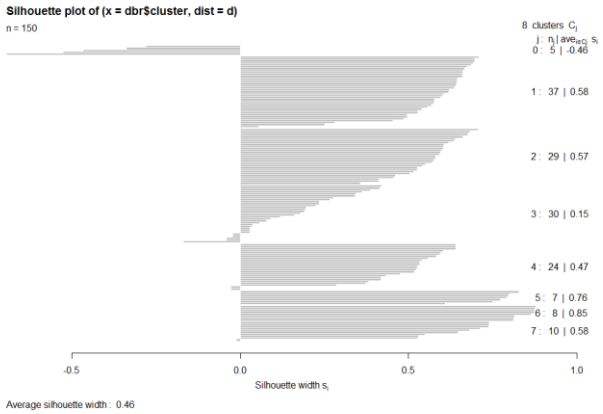

Figure 24. DBSCAN Silhouette Plot of Eps=11.1

5. REFERENCES

[1] Larose, D. and Larose, C.D. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 2014.

[2] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[3] Moran, K.H., Wallace, B.C., and Brodley, C.E. Discovering AAAI Keywords via Clustering with Community-sourced Constraints. *AAAI Conference on Artificial Intelligence (AAAI), 2014*.