Benchmarking in Clustering

K Medoids, Student T Model and DBSCAN

Math 252 Project II

Department of Mathematics and Statistics

San Jose State University

By

Qian Meng

November, 2020

1. **Introduction**

The purpose of this project is to benchmark clustering methods by performing experiments and using silhouette index(SI) and adjusted random index(ARI) as indexes. There is no clustering method which is generically the best for all types of data sets. For example, k-means clustering performs poorly with outliers since outliers have a huge influence on where the centroid of any cluster will be.

The three clustering methods which will be benchmarked in this project are k-medoids, student t model and density-based spatial clustering of applications with noise(DBSCAN). K medoids is an algorithm which creates clusters by using a data point in the data set as the centroid. Student t model is an expectation maximization (EM) algorithm which maximizes the likelihood of a student t distribution to model the data set. DBSCAN is an algorithm which looks for core data points and links those together to create clusters.

There will be four experiments performed in this project. One experiment uses real data and three experiments use simulated data. The experiments will test how well the clustering methods work with varying dimensions, outliers, number of clusters and correlation. The first experiment uses a real dataset which is the USPS MNIST handwritten digits. The remaining three experiments use simulated data. Each experiment keeps track of the SI and ARI values based on optimal number of clusters. For k medoids, the optimal number of clusters is chosen based on SI or ARI depending on which index is being recorded. For student t model, the optimal number of clusters is based on the BIC. For DBSCAN, the number of clusters is automated. The optimal number of clusters chosen by each method is recorded as well.

1. **DBSCAN**

DBSCAN will be discussed in it’s own section since the other two clustering methods have been discussed in the course MATH 252. DBSCAN is a clustering algorithm that only has two parameters which are and minimum neighbor points. DBSCAN goes through each data point and draws a boundary with radius around that point and counts how many other data points are within the boundary. Then if the number of points within the boundary are equal or greater than minimum neighbor points, the algorithm considers that a core point. The data points within the boundary are then also tested by the same process to determine if they are core points. This process is repeated until there are no more nearby core points, then all core points which are within distance from one another are considered a cluster.

Two interesting properties of the DBSCAN are that it does require the number of clusters to be input prior to the algorithm and that it will determine if data points are outliers or not. Number of clusters is not a necessary input since DBSCAN goes through each individual data point and determines if it is part of a cluster or not. Outliers are classified as data points which are neither core points nor within distance of core points. This will also affect the ARI and will be shown in the experiments.

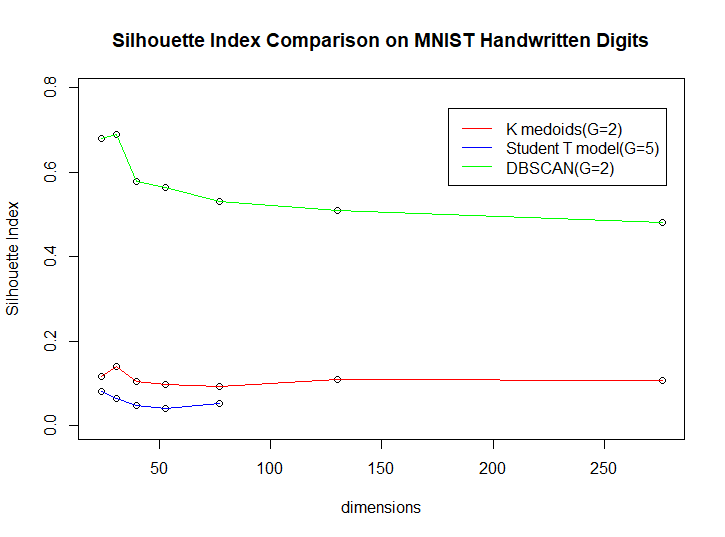
As far as determining the parameters, there isn’t any mathematical optimization. However, a common way to determine the parameter is by using the distance of k nearest neighbors. For the purpose of this project, is determined by using the distance of the kth nearest neighbor. There is an advanced algorithm called OPTICS which compares and uses every value of but was not looked into for this project. Determining the minimum neighbor points is usually done through trial and error since it depends on the size of the data set.

1. **High Dimensionality**

The first experiment was conducted with varying dimensions. The USPS MNIST handwritten digits data set comprises 28x28 images of handwritten digits from 0 to 9. These images are matricized using a gray scale and then vectorized to create a data point. There are 70,000 images total so the final data set is a 70,000x784 matrix. However, the full data set surpassed R’s limitations so a smaller subset was used. To create a usable data set, both the number of data points and dimensions needed to be reduced. Since there are 10 total classes, 100 data points from each class were chosen at random. To reduce the number of dimensions, principal component analysis(PCA) was used. PCA was performed based on the percentage scatter preserved using the formula shown below in Fig.1.

Fig.1 Percent scatter preserved by PCA

To clear up notation, refers to ordered eigenvalues of the covariance matrix of the dataset. P refers to the number of total dimensions and k is the number of dimensions in a lower dimensional plane. Using this formula, PCA was performed on the USPS MNIST handwritten digits data set preserving 70% to 99%. This gives the experiment a minimum of 24 dimensions and a maximum of 276 dimensions. Fig.2. and Fig.3. below show the results of the experiments.



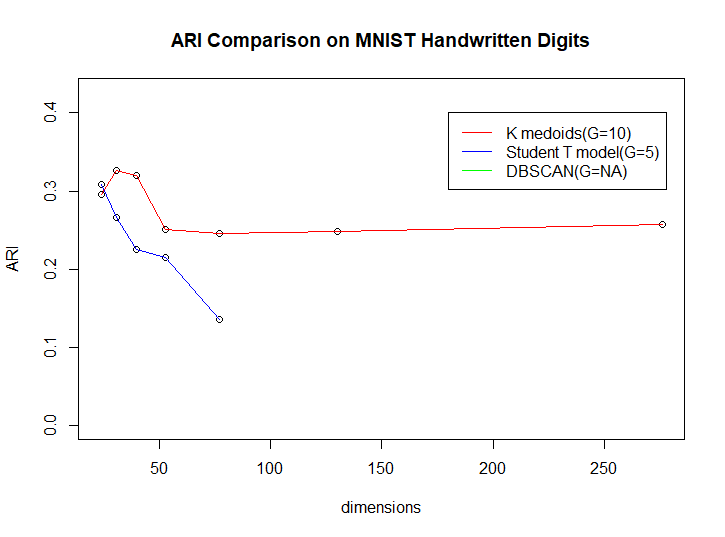


Fig.2 and Fig.3 Compares the SI and ARI to the dimensions

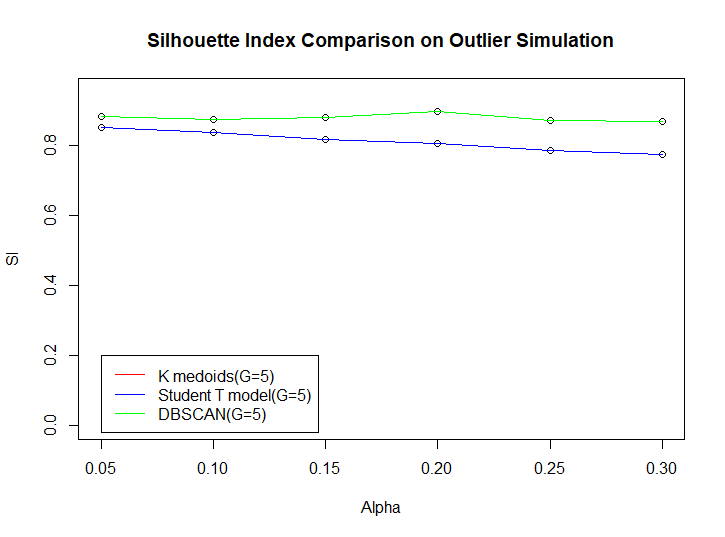
(G are number of clusters found in the best model per method)

Most of the clustering methods did not perform well in high dimensions. The highest SI was 0.69 and the highest ARI was 0.3265. Notably, the student t model performed consistently worse than the other two methods with lowest SI and ARI values. Furthermore, beyond 77 dimensions the student t model did not converge anymore. The DBSCAN provides a lot of information in both graphs. DBSCAN had the highest SI but it’s ARI was consistently zero. Addressing ARI first, the reason the ARI did not show up was because DBSCAN considered most data points to be an outlier. This also explains why the SI value is so high since there were only a few clusters. This implies that k medoids actually performed the best. K medoids was also able to pick out 10 clusters when using ARI as an index which is the true number of labels in this dataset.

A problem worth noting is that the number of data points may have affected all the clustering methods. The rule of thumb of using ,where n is the number of data points and p is the number of dimensions, was attempted but R could not handle that many data points. While these experiments suggest that all three methods performed poorly in high dimensions, using a different software and more time may have more conclusive results.

1. **Outliers**

This experiment tests how well each clustering method does in the presence of outliers. The data sets were generated using simulations of the contaminated normal distribution. The simulations were conducted using n=5, p=10 and varying levels of where is the percent of data points generated with higher variance. The outliers were generated by using variance that was ten times greater than the original variance. The mean of the distribution was randomly generated, but done so such that the means were different between clusters. Refer to Fig.4 and Fig.5 to see the results.



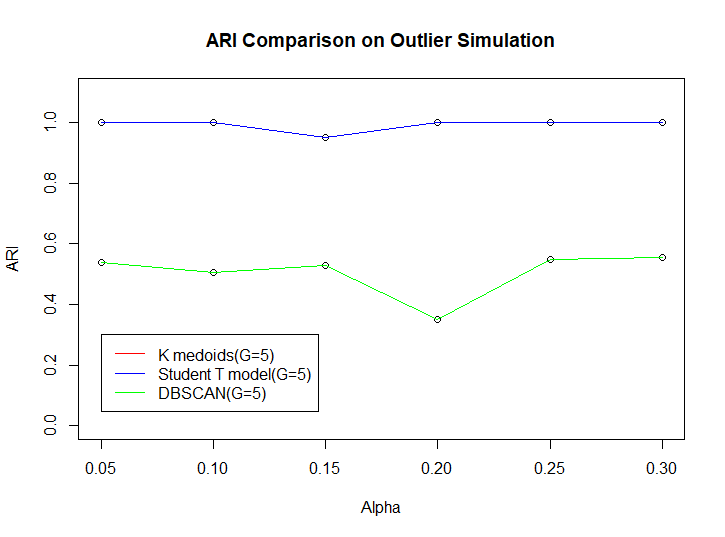
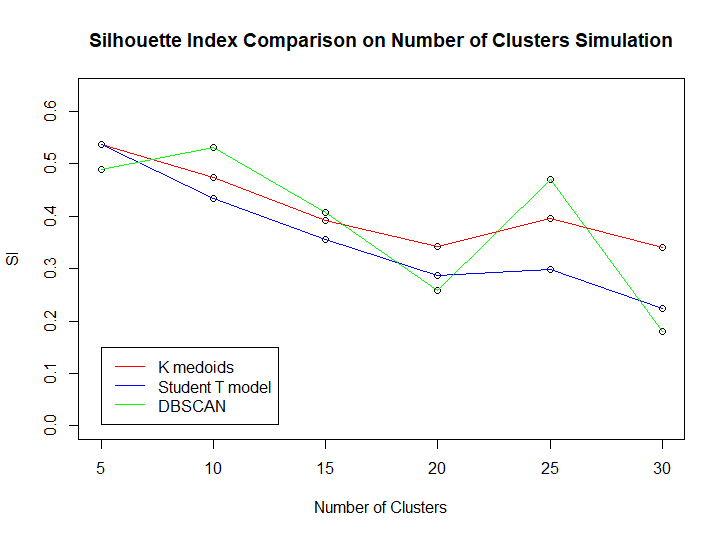


Fig.4 & 5 Compares SI and ARI to

First for clarification in the graphs above, k medoids and student t found the exact same clusters so their SI and ARI are also exactly the same. Overall, all three methods performed well with outliers. All three methods were able to find five clusters and have high SI and ARI values. This was expected since all three methods by design are supposed to be resilient to outliers. K medoid uses a median point and medians are not heavily influenced by outliers. The student t model can adjust for outliers by fine tuning the degrees of freedom. DBSCAN finds outliers and omits them from clusters completely. DBSCAN had a lower ARI value which was to be expected. DBSCAN also has a higher SI value which is probably due to the omitting of outliers thus removing the data points with low SI values.

1. **Number of Clusters**

This experiment tests how well each clustering method performs when the number of clusters changes. Each cluster was generated using the same variances but different means. Fig.6&7 shows the results.



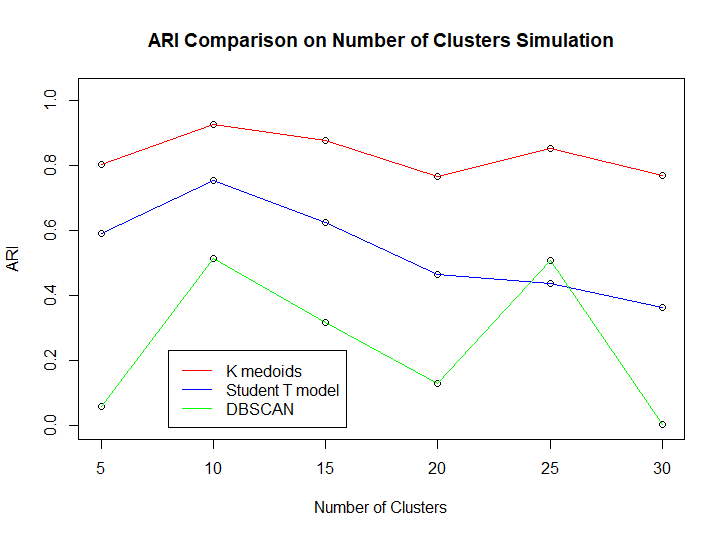


Fig. 6&7 Comparing SI and ARI to number of clusters

This time, there isn’t a clear method which performed the best and the SI and ARI values are average at best. K medoids performed better than the student t model in both SI and ARI in every number of clusters. DBSCAN has a very unstable behavior regarding SI values but the ARI is low as expected. By looking at the SI graph, it clearly has a downward sloping pattern suggesting that the more classes there are, the worse the performance of the clustering methods. The ARI appears to have a downward sloping pattern as well but it isn’t as obvious as in the SI graph.

1. **Correlation**

This experiment tests how well each clustering method performs with correlated data. These simulations are done by using n=100, p=10 and =1. The correlations tested are from 0 to 1 at intervals of 0.2 and the correlations between different variables are uniform. Negative correlations were not tested as they should behave the same way as positive correlations. Look to Fig.8&9 below for the results.

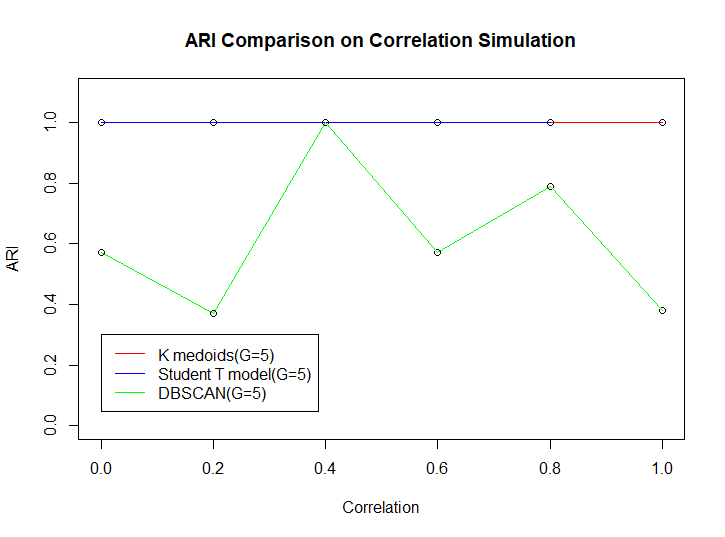
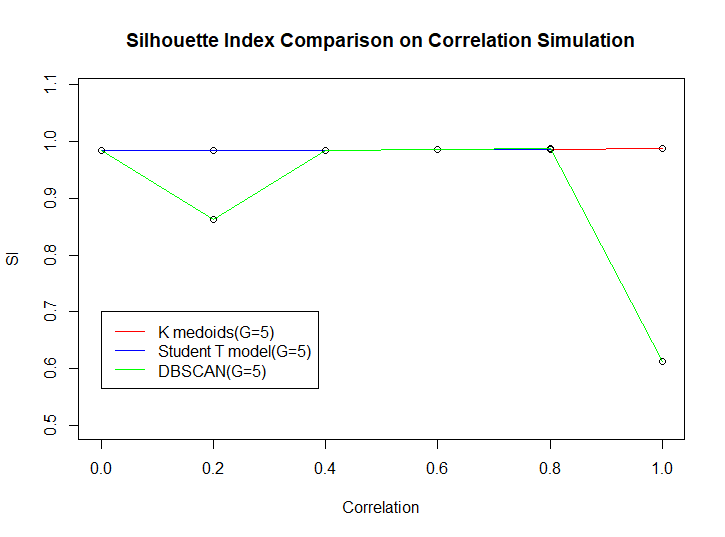


Fig.8&9 Comparing SI and ARI to correlation

Both k medoids and student t model performed very well with correlation. The k medoids and student t model lines are overlapping in the graph. The SI and ARI values for k medoids and student t model are close to 1 but aren’t actually 1. Also, it is worth noting that the student t model did not converge when the correlation was 1. DBSCAN did not perform well with correlation which makes sense. The parameter in DBSCAN is looking for a radius so it is searching for something circular. Correlation implies that the data points are in the shape of a line. While DBSCAN did not perform poorly, correlated data is not it’s strong suit.

As mostly a side note rather than a finding, we were suspicious at first since the SI and ARI values were so high but the benchmarking functions were the same as the ones used in the prior experiments and the SI and ARI values aren’t just consistently 1.

1. **Conclusion**

K medoids and student t model seem similar while DBSCAN is different. K medoids and student t model performed well with outliers and correlation but performed poorly with high dimensions and high number of clusters. DBSCAN performed well with high dimensions and outliers but poorly with correlation and high number of clusters. A table is provided below in Fig.10 to summarize the findings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Dimensions | Outliers | Number of Clusters | Correlation |
| K Medoids | X | ✓ | X | ✓ |
| Student T Model | X | ✓ | X | ✓ |
| DBSCAN | ✓ | ✓ | X | X |

Fig.10 Summary of findings