Test Clustering

Nour Mihamou

CSIS-320 Machine Learning

Dr. Dan DiTursi

April 27, 2022

Introduction

The following report was using data from Siena College's course catalogs. The main objective of this project was to discover which group (schools, departments, or course prefixes) would optimally cluster the course descriptions. With that being said, the methods that were used to test these clusters were Latent Dirichlet Allocation (LDA), KMeans, Agglomerative, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). In order to discern the accuracy of our clusters and identify the overall optimal number, the adjusted rand index score (ARI) and silhouette score were calculated when appropriate.

The data containing the descriptions was further cleaned by separating the indices and the text into their own arrays. Two new objects of the cleaned text were created to hold a scaled bag of words generated by CountVectorizer and TfidfVectorizer. The first which turns every term into a frequency of the word within the document, and the latter which creates a statistic that represents how important a term is in a document according to its frequency.

Methods & Results

Parts 1 - 3

The LDA classifying model was used on three different component sizes; 3, 33, and 57. The ARI score, calculated by making pairwise comparisons against ground truths to measure similarities, was applied to each set of components. The first test, 3 schools, yielded an ARI score of 0.103. This is a relatively low score, though it is proven the highest among the entire group. When grouping by 33 departments and 57 prefix codes, ARI scores of 0.043 and 0.044 were yielded, respectively. According to these scores, the best way to classify the course descriptions is by schools, three components.

KMeans, the most common clustering algorithm of the bunch, allows more flexibility when wanting to test for the best approach. Since the number of clusters are known, 3, 33, and 57 were plugged into the KMeans function. The following ARI values were returned: -0.030, 0.095, and 0.134. Again, they were small values, but KMeans suggested that 57 clusters was the most optimal of the group, and 3 the worst.

Agglomerative clustering was able to yield slightly higher, yet insignificant ARI scores. Three clusters came in first place with the highest score of 0.473; 33 groups were second with 0.205, and 57 groups third with 0.197. This method of hierarchical clustering showed that, again, clustering by three groups was the best option of the three cases tested. These scores further imply that the higher your cluster size, the less similar the terms will be in each group.

Part 4

The following sets of tests all had similar approaches, which were to collect an array of silhouette scores based off of a range of cluster values. Moreover, some methods used an additional step to narrow the range as optimally as possible.

Based on the knowledge from testing 3, 33, and 57 clusters, test values less than 3 and greater than 57 showed slight shifts that were further analyzed. A range of 2 to 101 clusters was applied to LDA and ran through the silhouette score method. Although three initially yielded the

highest ARI, figure 1 shows the lowest silhouette score and the highest silhouette score, which were 45 and 2 respectively. This implied that two clusters were the most optimal components for the course descriptions. It was important that the range of silhouette scores went from 0 to -0.2, where -0.002 was the highest value. Negative values typically indicate that the points are not in their rightful clusters, and values closer to 0 meant that data points were on the boundary lines of their rightful clusters. Unfortunately the latter was not observed in the LDA cases.

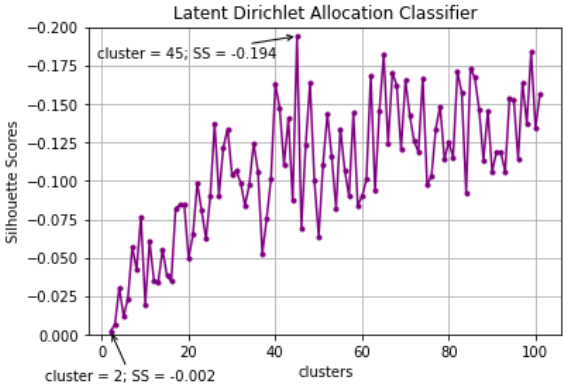

Figure 1. LDA silhouette scores

Figures 2-5 were the results after applying the Elbow Method for KMeans to narrow down the list of cluster values to test. Values were chosen by eyeballing each figure and subjectively choosing where the lines bent or showed a drastic shift. Inevitably due to the unclear nature of these graphs, the range 2 to 101 clusters were applied to the KMeans method, and the silhouette score was calculated for each *k*. Per figure 6, the scores produced a range of 0 to 0.1 and the highest value equaling 0.058 (clusters = 101). Unlike LDA the silhouette score increased as the number of clusters increased, and remained positive. This leads to the belief that the data points were getting closer to their rightful cluster.

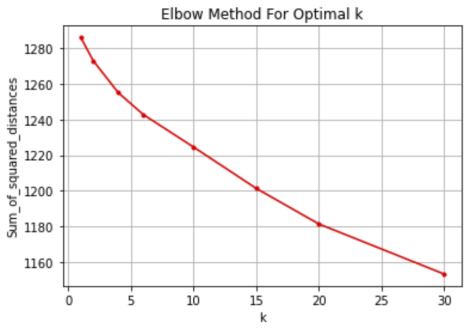

Figure 2

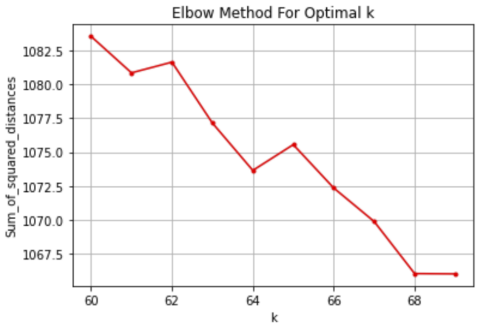

Figure 3

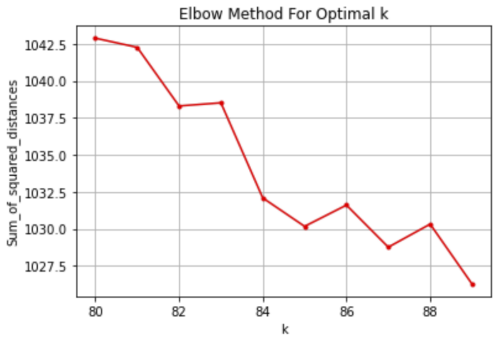

Figure 4

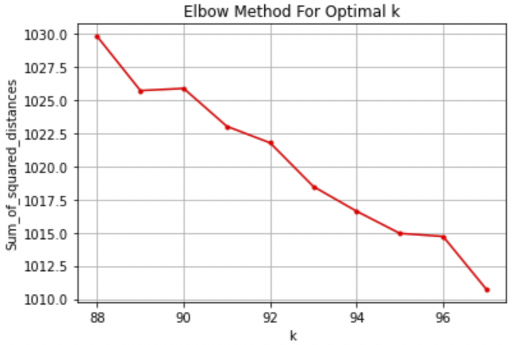

Figure 5

Agglomerative clustering was tested with a cluster range of 2 to 31. Figure 7 shows the highest silhouette score of 0.122 is at cluster 12. The score indicates since it is closer to 0 than it is to 1, then the data points are at the boundary lines of their optimal cluster. Any clusters greater than or less than 12 were not fit for the course descriptions.

Before testing anything in DBSCAN, in order to prevent any ValueErrors, 1000 was added to every label for noise points which equal -1. Unfortunately, ValueErrors were still produced when testing with an *eps* > 1. Therefore *eps* remained at 1, and *min_samples* ranged from 1 to 4. Figure 8 shows the results of testing each minimum number of samples that are required to make a cluster according to the circle's radius (*eps*). When *min_samples* = 1, the clusters generated were 968, which produced the highest silhouette score within the entire report, 0.212. Furthermore, when *min_samples* = 4, the silhouette score was 0.008 with 26 clusters being the least optimal size to group.

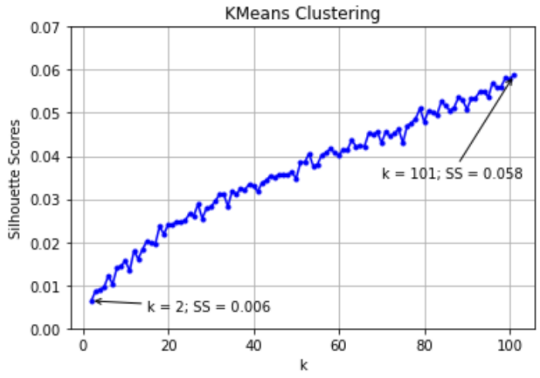

Figure 6

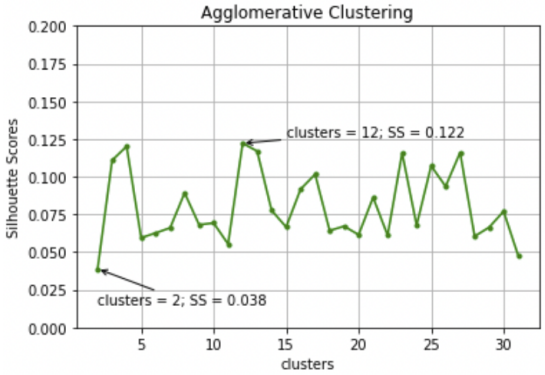

Figure 7

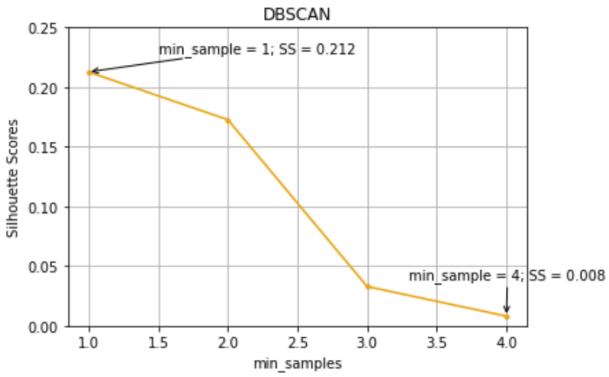

Figure 8

Conclusion

All the methods tested for clustering each have their unique power. LDA and Agglomerative clustering both suggested clustering by schools would yield better scores, yet not the most effective. To determine the most ideal clustering, DBSCAN was able to prevail and provide the optimal number of clusters for Siena's course descriptions. Although this method yielded the highest silhouette score, the number of clusters is quite inconvenient. Grouping the

descriptions in 968 clusters is intuitively not the most effective way to store courses, which is why the silhouette score is still <1. Since it is closer to 0, that means most points are noise or they lie on the boundary line. Agglomerative clustering had the second highest silhouette score but suggested a significantly lower size of clusters (SS = 0.122, clusters = 12).

Discussion

In the future, further cleaning of the data should be performed. Though there may be a number of attributes that lead to the errors faced throughout the project, possibly a well-structured dataset would help identify these issues easily despite how time consuming the activity. For KMeans, testing higher values of *k* to observe whether a decrease in silhouette score starts, and at what point. Additionally, testing DBSCAN with higher values of *eps* to understand the effect of the circle boundary's radius, which would ideally produce higher silhouette scores and lower cluster sizes. Unfortunately the method would run into errors when *eps* >1 or *min_samples* > 4

Further analysis should be done with LDA, starting by creating an algorithm that extracts the feature names from the vectorized bag of words. The most common words in each cluster would be observed and would give additional insight in whether that is an optimal clustering configuration. Finally, a dendrogram should be produced by Agglomerative clustering which visualizes the best number of clusters.

References

Bhardwaj, Ashutosh. “Silhouette Coefficient: Validating Clustering Techniques.” Medium, Towards Data Science, 27 May 2020, https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c.

“Data Visualization – How to Pick the Right Chart Type?” EazyBI, https://eazybi.com/blog/data-visualization-and-chart-types.

“Learn: Machine Learning in Python - Scikit-Learn 0.16.1 Documentation.” Scikit, https://scikit-learn.org/.

Müller, Andreas C., and Sarah Guido. Introduction to Machine Learning with Python = Python Ji Qi Xue Xi Ru Men. 2017.

“Visualizing DBSCAN Clustering.” Naftali Harris, https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/.