Conner Yin
W1614583
May 28, 2023

COEN 140 - Program 3 Report

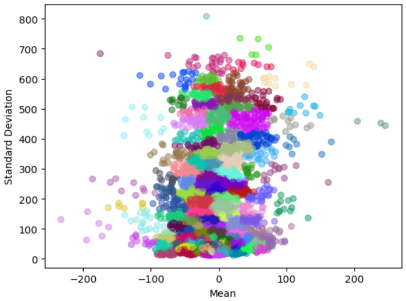
I. Introduction

Program 3 is a clustering problem that asks us to assign each sample in a list of EKG signals to a cluster. We're given 11,500 samples which will needed to be clustered into 115 clusters. The accuracy will be determined by Normalized Mutual Information score (NMI), which is a common external index metric for measuring accuracy of clustering algorithms. We will be using an existing clustering algorithms such as K-means, DBSCAN, Agglomerative, etc. along with pre-processing methods in order to produce the best results. Our prediction file will consist of 11,500 numbers representing the cluster to which the ith sample belongs to.

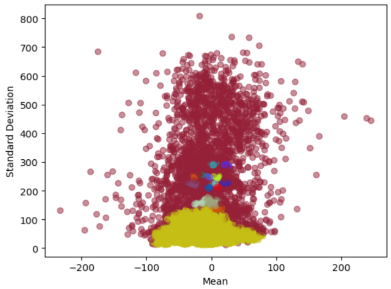
II. Approach

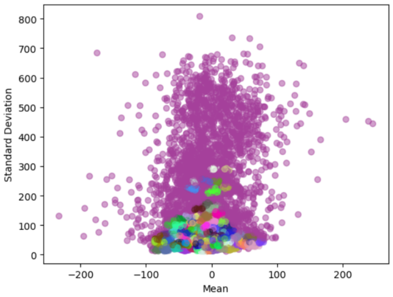
Compared to programs 1 and 2, my algorithm was significantly harder to test, as unlike classification and regression, there is no training data that we can rely upon for comparison. For those problems, we can run our model on a segment of the training data to garner an estimation on its accuracy. Therefore, testing for this program was mostly based on submissions to the CLP to verify accuracy.

I began with a naive approach of simply running the raw data through each of the primary clustering algorithms, arriving at an NMI of 0.1576 utilizing only agglomerative clustering. This provided a solid baseline before moving onto pre-processing the data via feature selection/extraction. Running the clustering algorithm on the means of each sample yielded a NMI of 0.1938 and using the mean and standard deviation yielded an NMI of 0.2311. In order to gain more insight into what the clusters looked like, I wrote a code snippet that would generate a scatterplot plotting each of the samples' mean and standard deviation and assign a random color for each of the 115 clusters. This scatterplot aided in my experimentation and optimization.

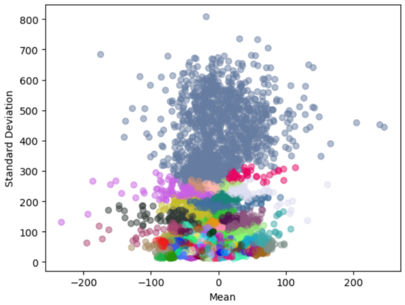


DBSCAN works in a different way than k-means or agglomerative in which points will be mostly be associated to the set of core points or outliers. I wanted to test what the scatterplot would look like with DBSCAN since there appeared to be several outlier points. With a DBSCAN on the mean and standard deviation, I noticed that the graph was actually extremely dense towards the bottom, with most of the points being in the yellow cluster at the bottom, with all the red points being considered outliers (see upper figure to the right). This demonstrates that naive agglomerative clustering is not optimal, since the the clusters are too evenly distributed considering the density of the bottom. Therefore, I attempted a new approach with first removing all the outliers found via DBSCAN, and running the remaining points under an agglomerative clustering algorithm. This resulted in a significant increase in NMI to 0.2566. The combined graph can be seen in the lower figure to the right, where outliers are represented by purple.

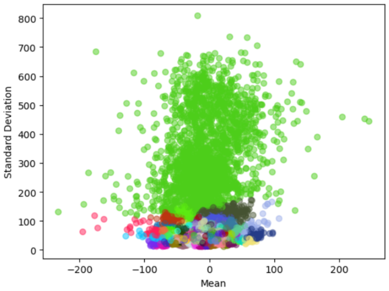




The next optimization step would be to correct some of the outlier points. Instead of placing them all in a single cluster, I found the closet point to each outlier point and assigned it to that cluster instead. For finding the closest, Euclidian distance was used since other similarity functions such as cosine don't make sense in the context of a scatterplot. This once again improved the NMI to 0.2703 and resulting in the following graph.



In the last push to increase the NMI as high as possible, I tuned the parameters of the DBSCAN in order to change the number of outliers. The value of epsilon (eps) determines the maximum distance between two points to be considered neighbors and min_samples determines the the number of neighbors a point must have to be considered a core point. After many CLP submissions, I optimized my NMI to 0.2887 using parameters eps = 5 and min_samples = 18. This tune of DBSCAN resulted in 2635 outliers, and the highest NMIs were all approximately in the ballpark of 2500 to 2800 outliers. The final figure can be seen in the figure to the right



III. Final Methodology

1) Import the data from the csv into a float type pandas dataframe.
2) Find the mean and standard deviation of each row and column stack them together.
3) Create a DBSCAN model with eps=5, min_samples=18 and fit predict to Mean SD stack
4) Remove all outliers (when DBSCAN yields -1) from Mean SD stack
5) Run agglomerative clustering on the non-outliers with 115 clusters
6) Assign a cluster to outliers by assigning to cluster of point with lowest euclidian distance
7) Print results to prediction.txt file

IV. Evaluation and Results

At the time of writing this report, my solution scored 3rd in the CLP with an NMI of 0.2887. This result could definitely be improved since 1st place currently has an NMI of 0.3157. However, I'm very proud with my results as I've slowly managed to push the NMI up through use of different methods. Also, there were a couple more approaches that I attempted that I didn't put in the earlier section as they ended up not improving my result but they are worth mentioning here. Submitting randomly generated results actually results in a NMI of approximately 0.23 (100 of each number from 1 to 115). Utilizing the quartlies (minimum, first quartile, mean, third quartile, maximum) instead of mean and standard deviation provides decent results, just not quite as high as mean and standard deviation. I also tested other feature extraction techniques that were used in Program 2 including Principle Component Analysis (PCA), Truncated SVD, Locally Linear Embedding (LLE), etc. I tested these methods on both the raw data and on the mean/standard deviation stack, but neither was able to improve results.

Overall, I've learned a lot about clustering and using intuition to try out new approaches. I feel that I've only improved my work from Program 1 to Program 3 and am definitely the most proud of my results for this program. While my solution may not be optimal, I steadily made progress through extensive experimentation with testing of different ideas. I now feel much more confident in my machine learning abilities and aim to work on my own personal projects involving classification, regression, and clustering algorithms.