Mid-Trem Report

Addressing proposal feedback

1) What do you exactly mean by “outliers in each class”? Are there different classes considered? How does this help the outlier detection?
We were thinking about grouping quasars based on their colors and then running our algorithms on each group. But this idea was refined and now we run our algorithm on the full dataset.
2) In the evaluation plan it is not entirely clear how success would be measured. Is it the overlap with other outliers that other studies have found?
Yes we compare our outlier with a previous study which on a smaller dataset. We also test the success in these ways:
A) Comparing the labels that different methods find, expecting they have a consensus.
B) Comparing the spectrum of an outlier object with the expected spectrum of inlier quasars
3) Are those 183 features derived from the spectrum? or independent? or a mix? how about using the spectrum directly? It would be interesting to see what can the non-spectrum related features can do vs. the spectrum ones vs. using the spectrum directly.
Looking more carefully at the features we found that our study would be very limited rather than taking all features into account. Some features come from spectrum and some from photometry (measuring color). We decided to focus on the colors ending up with 21 features. We preserve spectrum for testing our results, in a way that outliers should have spectra very different from normal objects.
4) How is the Agglomerative Clustering tie into the main objective? Perhaps you may want to combine it with the Isolation Forest and run clustering before and after outlier reduction also as a measure of estimating to what extent you have been able to get rid of (most of) the outliers (with the argument here being that data that can be clustered better have fewer outliers)
Agglomerative clustering can find the main big cluster of objects by demanding very few clusters. Also we compare the labels of agglomerative clustering with DBSCAN and the scores of isolation forest.

Data Reduction and Feature selection (Reza Monadi)

We selected 7 measured fluxes available in the catalog. Each flux is proportional to the amount of radiation we get from a quasar in a specific range of wavelengths. For example a quasar is red if we get higher flux in longer wavelengths and lowe flux in shorter wavelengths. Therefore, colors can be obtained by by looking at the ratio of each pair of these fluxes → $color_i = \log(Flux_1 / Flux_2)$
We use logarithm because it is easier to understand especially for very large and very small ratios. For each measured flux, there is a reported variance in the catalog. Keeping only fluxes with Signal/Noise>2 we got 140,000 objects which are more reliable. Having 7 fluxes we will get 21 colors (7!/(2!*5!)). However, not all of the pairs are useful, especially if two fluxes have a close range of wavelengths. We can use 10 less reliable fluxes which have wavelengths far enough from each other. Pair-plots of these 10 features show that some of them are very correlated, so we can do more dimentiolaty reduction.

Implementation of DBSCAN Algorithm in Python (Hasin Us Sami)

Since our project focuses on detecting outliers in quasars, or in other way, data points that deviate from the normal behavior of quasars, a number of outlier detection algorithms has been studied. Among them, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) has been chosen because it is one of the most efficient algorithms in detecting outliers.
Before applying self-implemented algorithm from scratch, DBSCAN built-in function in python has been experimented on the pre-processed dataset with 144582 data points and 21 features just to verify how much accurate this detection technique would be in the case of our quasar dataset. Optimum values for DBSCAN parameters have been chosen using trial and error method. After performing DBSCAN, Principal Component Analysis (PCA) function mapped the data points into a 2D plane for the purpose of visualization.

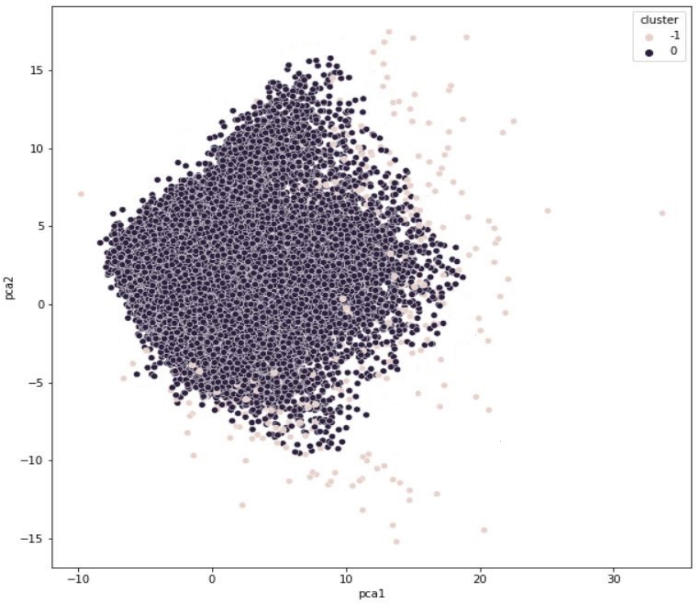


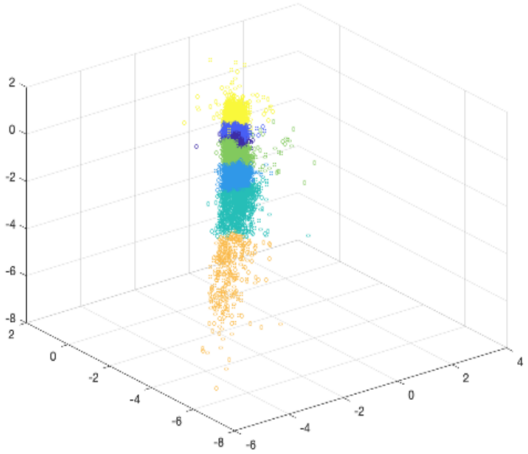
Figure: 2-D Visualization of the data points (-1 circles are outliers)

The work is in progress and focus is on to further increase the accuracy. The algorithm has been implemented from scratch in python but unfortunately there are several errors observed in the code that have to be dealt with. The next step is fixing the errors and runnig the algorithm on our quasar dataset to detect outliers.

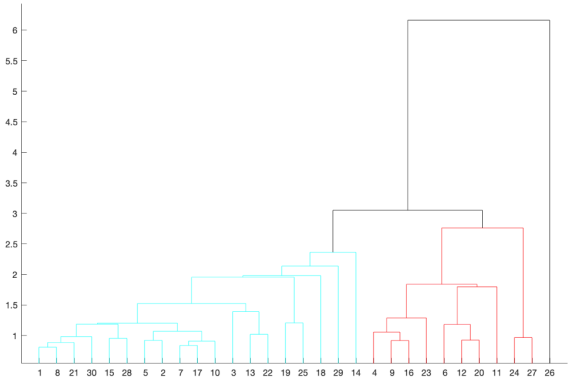
Agglomerative hierarchical cluster tree (Terrance Kuo)

A hierarchical clustering group has data over a variety of scales which it represents by creating a cluster tree or dendrogram. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. Since I am trying to do agglomerative clustering, I am using MATLAB to process my data. Some difficulties have been prevalent in trying to do this, as I do not have as much experience with MATLAB. Additionally the dataset is enormous as it is a 144582 by 21 table. Trying to represent this is difficult and I will need more information on what segments to process. The current methods to represent it in a 3d environment also cannot process all 144582 points without giving an error from insufficient memory so right now I have to put them through in segments. However I have been able to get visual data and split them into 7 clusters like with the pictures below.

Data plot of segment



Dendogram



Isolation Forest (Arman Irani)

After preprocessing our dataset to contain only a select amount of features, an Isolation Forest will be designed in Python from scratch in order to discover “anomalies” in our dataset of quasar features. Currently the algorithm is in production, and as is with a standard Isolation Forest will consist of an ensemble of randomly generated Isolation Tree’s and an anomaly will be defined using a depth-based approach as those with short average paths, with a score between [0,1]. The instances with the highest anomaly score will be considered to be outliers. This will hold true under the assumption that anomalies will require less splits than normally required. The Forest will be designed to deal only with continuous data. Parameter tuning will be utilized to optimize the subsampling value and the number of trees during the training period. It will be difficult to tune these parameters as there is no efficient way to confirm our outlier results so our parameters will be selected carefully.