Project Milestone - Classifying Blazars and Cataclysmic Variables from the Catalina Real-Time Transient Survey

Adrian Markelov
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
amarkelo@andrew.cmu.edu

Kai Wen Wang
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
kaiwenw1@andrew.cmu.edu

Yizhou Xu
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yizhoux@andrew.cmu.edu

1 Introduction

In the cosmological sciences there is a growing abundance of recorded transient and variable events from large, digital, synoptic sky surveys. Many follow-up facilities are specialized to analyze specific types of astrophysical phenomenon for various characteristics in terms of distance, cadence, wavelengths, light intensity, etc.[1]. To delegate the follow-up facilities without wasting and duplicating their efforts, an automated, probabilistic classification of the detected variables and transients becomes necessary. These decisions must be made fast, as many of the events need rapid follow-up for proper analysis. One of these important classifications to make is Blazars vs. Cataclysmic Variables (CVs). [1]

As of now, approaches for classifying significant celestial phenomena such as Blazars and CVs require very long time series of a single object to make a reliable inference. In many cases, a full time series is collected in periods of around 8 years. As a result, any available data for this classification problem is highly limited and likely to be incomplete. Even with long streams of data, on the order of hundreds of data points, the classification accuracy can be improved. Although Blazars and CVs are very different in their physical appearance and their behaviors, the observed light magnitudes are very similar in nature, and almost seemingly random. Another challenge is the imbalance in data, as CVs are almost 3 times more likely to occur then Blazars. This inherent bias towards classifying CVs along with the shortage of data makes this problem highly difficult.

The ability to make very accurate inferences, especially on shorter streams of data, can significantly boost the productivity of facilities to capture and discover useful astronomical information. Success with the binary classification of Blazars and CVs can lead to promising approaches to be extended to other astronomical phenomenon, such as supernovae, asteroids/flares, etc.

[1] Blazars are a very compact quasar associated with a presumed supermassive black hole at the center of an active, giant elliptical galaxy. CVs are stars which irregularly increase in brightness by a large factor, then drop back down to a quiescent state.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2 Dataset

We will use the Blazars and CVs data from the Catalina Real-time Transient Survey Dataset [2]. The Survey is a synoptic exploration of thirty three thousand square miles of the sky to observe transient events. While it is the most comprehensive dataset for this problem, there are only around 938 reliable observations to date, 704 CVs and 234 Blazars. After basic preprocessing of the data, around 100 of these observations are highly incomplete and comprises of less than 50 data points. As the observatories usually collect data about multiple astronomical phenomenon simultaneously, the observed data is highly imperfect with irregular time collections and an inconsistent number of data points per sample. In general, there are times when data is not collected at all when the observing facilities are not looking in this area of the sky. Each data point contains relevant information about the time of the observation in Modified Julian Date Time (MJDT), the observed light magnitude (Unfiltered CSS Magnitude), and error bars for the observed light magnitude.

Our model will be a standard classification model, with inputs being a stream of observed datapoints, or a feature transformation of it. The output will be a binary class of "Blazar" or "CV".

3 Related Works

Not much work has been put into analyzing our specific problem but many of the factors that will be a part of our solution have been well researched. First we would like to point out that we have been given this data set by the CMU Statistics Professor Chad Schafer. He has done some previous work with this data achieving around 90% accuracy on the full length data. In the limited amount of work that he has done, he used both neural networks and random forests to achieve the above accuracy. His models also used various feature transformations including the structure function to filter out unknown human made time irregularities during the data collection process and a quantile regression transformation to reduce the dimensionality of the data. However, to our knowledge, there has not yet been an extensive and careful investigation into this specific classification problem.

4 Method

4.1 Feature Transformations:

Structure Function Transformation
Summary: The structure function is a feature transformation of the data that takes in a single data point, which is a function (light curve) of light magnitude with respect to MJDT and returns a function of the log of the absolute differential magnitudes with respect to the corresponding absolute time differentials.
NOTE: From now on all of the transformations and models we work with are done on the basis of the structure function space and one structure function corresponds to a single cosmological object.

Light Curve Function: $\phi : \mathbb{T} \to \mathbb{M} \quad s.t.$

$\mathbb{T} = \{\text{MJDT}\}$

$\mathbb{M} = \{\text{Light Magnitudes}\}$

Structure Function: $\psi : \mathbb{T}' \to \mathbb{M}' \quad s.t.$

$\mathbb{T}' = \{t' : t' = |t_i - t_j| \quad s.t. \quad t \in \mathbb{T}, i, j \in \mathbb{N}\}$

$\mathbb{M}' = \{m' : m' = \log_{10} |m_i - m_j| \quad s.t. \quad m \in \mathbb{M}, i, j \in \mathbb{N}\}$

Quantile Regression Transformation

The quantile regression technique essentially tries to estimate the quantile function of the y-axis. It is analogous to linear regression, which tries to learn $y = f(x) + \epsilon$. Here, f is the quantile function defined by

$$Q : [0, 1] \to \mathbb{R} \qquad Q(p) = \inf\{x \in \mathbb{R} : p \leq F(x)\}$$

where F is the cumulative distribution function.

We performed a linear quantile regression for the quantiles 0.05, 0.15, ..., 0.95 on the output of the structure function. Each regressor produces a slope and an intercept for the regression, for a total of 20 points.

PCA Transformation
Because our data is highly irregular and very high dimensional we regularize our structure function into a 100x100 pixel density image. Though this can be very hard to sift through for any machine learning algorithm especially because we have very few data samples. So, we have used Principle Component Analysis (PCA) to transform our features into a much smaller dimensional space. First, though we must use spectral analysis to find what proportion of the features are actually contributing a significant amount of information. Spectral analysis gives us a mapping between each feature and its contribution to the explained variance of the data. This explained variance is proportional to the amount of information that the feature provides with respect to all of the other features. Using this mapping we can decide on the number of dimensions we want to keep in our new space after the PCA transformation. We do not want redundant features nor do we want to get rid of too many features and lose information. So we will pick a dimension size that will provide us with at least 90% of the information that all of the dimensions could provide us.

4.2 Classification Models:

Our first tests used classical models, including SVMs, Random Forests, Multi-layer Perceptrons, Adaboost with Decision Trees. These models were adapted from scikit-learn. To combat the issue of manual feature engineering and extraction, we sought to do this automatically by using a convolutional neural network (CNN). We took the 2D array outputted by the structure function and performed a distribution estimation with 2D histograms of evenly-spaced bins, 100x100 number of them. We then created a 100x100 image where each pixel corresponds to a bin and the intensity of the pixel corresponds to the bin size. Finally, we ran a simple 3-layer CNN using Keras on the image to perform feature extraction. We also included a 0.5 Dropout for regularization.

A previous issue that we have yet to address is that the dataset is imbalanced, with only around 25% of the training data consisting of blazars. To counter this particular problem, we investigated into classifiers specifically suited for imbalance datasets. Several common techniques for dealing with imbalanced datasets include over-sampling, under-sampling, as well as increasing the weights on minority samples. One particular classifier that we have found to be particularly useful is the balanced bagging classifier. The purpose is to balance the dataset prior to training the classifier, in order improve the classification performance. Another advantage of this model is that it avoids focusing on the majority class, which is a common problem for many classifiers such as random forest.

5 Preliminary Results

Model Results on Quantile Regression Transformation:

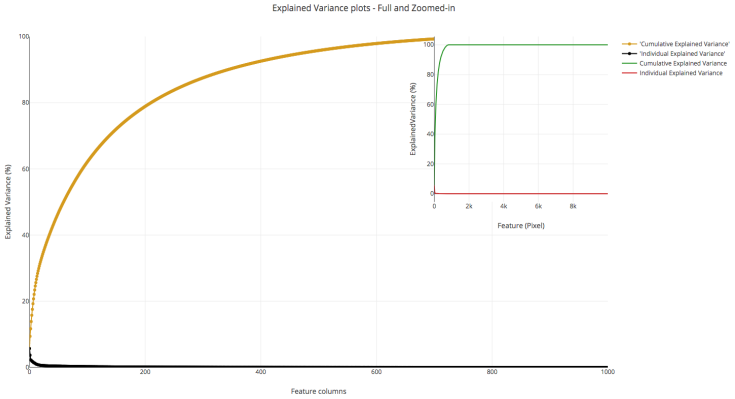
model	kNN	SVM	Adaboost	Random Forest	Neural net
accuracy	86.1	85.8	84.5	84.6	85.0

The table shows the testing accuracies for models trained and tested on the quantiles extracted from quantile regression. Most of these models produce consistent accuracies, hovering around the 85%.

Spectral Analysis Results and Model Results on PCA Transformation:
Spectral Analysis Results: Using spectral analysis on the covariance matrix of each objects structure function we can see how much information each of our features is providing compared to all of the

other features. The provided graph below shows the cumulative explained variance of each of the features starting from the feature that provides the most explained variance to the least. This graph tells us that 92.6% of the explained variance comes from 400 of the 10,000 features. Thus, we will use PCA to reduce our 10,000 dimensional feature space to 400 dimensions.

Model Results on PCA Transformation to 400 dimensions: So far using our new PCA transformed data we have only tested our CNN model and have had approximately 90% accuracy.



Model Results on Balanced Bagging classifier:

We trained this model on 80% of the data, and achieved an accuracy of 86.8%. There are more useful metrics to evaluate performance of imbalanced dataset than accuracy, such as precision-recall and the ROC AUC curve. For precision vs. recall, our model has the following confusion matrix:

	Predicted CVs	Predicted Blazars
Actual CVs	101	14
Actual Blazars	7	37

Our model has an ROC AUC score of 86.0%, which demonstrates good performance from our classifier.

6 Timeline

Although we did not achieve our initial optimistic goal of well-over 90% accuracy, we have made progress on understanding the nature of our dataset. We now realize that our metric should not be accuracy, since the classes of our dataset is highly imbalanced towards CVs. After trying out many models with limited initial success, we realized that it may not even be feasible to achieve such high classification accuracies without some machinery to extract many useful features.

Division of work We all worked on the initial processing of the data and running simple models. After that, Adrian worked mainly on PCA, Kai worked on mainly the CNN, and Yizhou worked mainly on imbalanced classification.

Below is our plan for the last two weeks:

Time	Plan
Apr 15-Apr 21	Change gears towards optimizing recall and accuracy, begin testing on short length data, try data augmentation.
Apr 22-Apr 28	Wrap up model testing for full-length data. Focus on short length data. Prepare poster and presentation. Document 7-8 pages.

7 Appendix

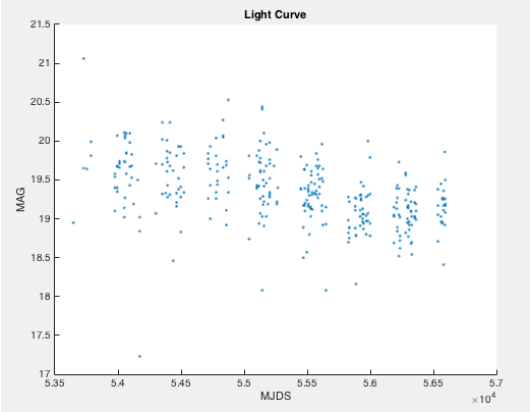


Figure 1: Example plot of raw Blazar data.

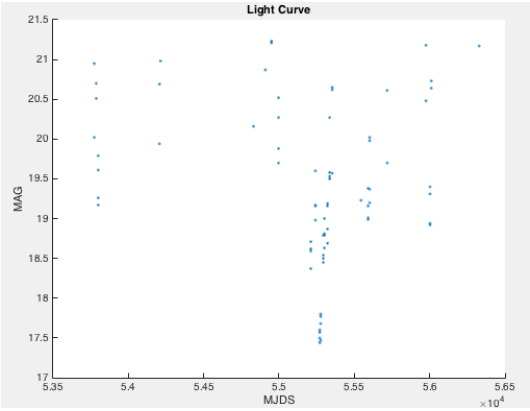


Figure 2: Example plot of raw CV data.

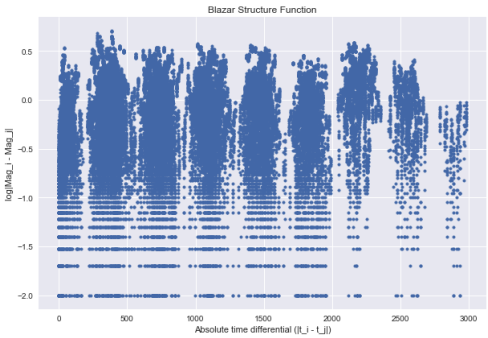


Figure 3: Plot of the structure function for Blazar 77.

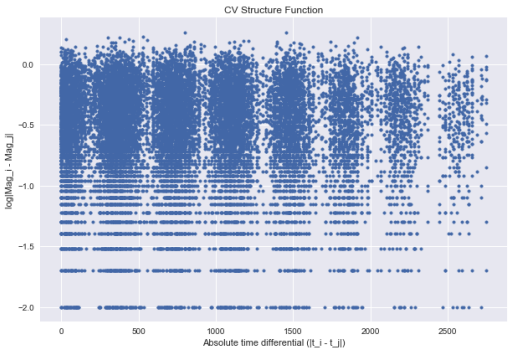


Figure 4: Plot of the structure function for CV 251.

The above figures show example plots of the output of the structure function applied to Blazar and CV data. The gaps we see in the data are superficial and are artifacts of the irregularity in observations.

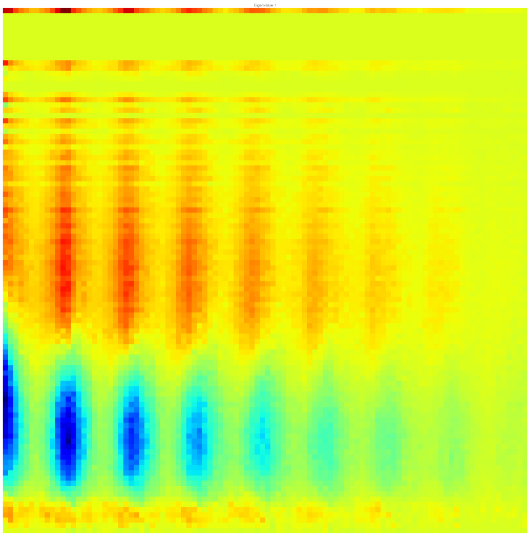


Figure 5: Plot of the first eigenspace of a structure function image.

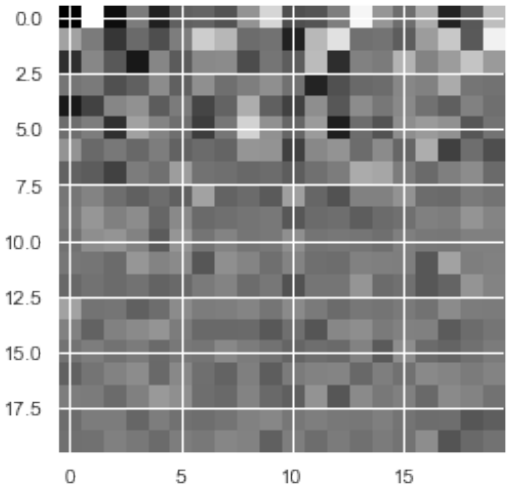


Figure 6: Plot of a low dimensional structure function transformed by PCA (100x100) -> (20,20).

References

[1] Drake, A.J. et al. First Results from the Catalina Real-time Transient Survey, 2009, ApJ, 696, 870.

[2] Drake, A. J. “Data of Blazars and Cataclysmic Variables.”

[3] Esteban, Cristóbal, Stephanie L. Hyland, and Gunnar Rätsch. "Real-valued (medical) time series generation with recurrent conditional GANs." arXiv preprint arXiv:1706.02633 (2017).

[4] Guss, William H., and Ruslan Salakhutdinov. "On Characterizing the Capacity of Neural Networks using Algebraic Topology." arXiv preprint arXiv:1802.04443 (2018).

[5] Schafer, Chad, director. *Statistical Challenges in the Search for Dark Matter*. Birs, Banff International Research Station, 26 Feb. 2018, www.birs.ca/events/2018/5-day-workshops/18w5095/videos/watch/201802261547-Schafer.html.