Light Curve Classification using Three Machine Learning Methods

Evan Linck
University of Wisconsin-Madison
Madison, WI
elinck@wisc.edu
https://github.com/evanlinck/ece532

Abstract

Changes in light from distant stars and other astrophysical objects can tell us much about the things we see in the night sky. Every type of variable source of light has a unique signature in how the light changes over time, but it is common across different examples of that type of object. This means that observations of light from astrophysical objects over time, referred to as a light curve, are a ripe area to apply machine learning classification algorithms. In this work, I explore classifying 1,000 light curves from 6 different types of astrophysical objects using LASSO regression, k-nearest neighbors, and neural networks. I find that neural networks perform the best (up 92.6 percent accuracy), but take a longer amount of time, followed closely by k-nearest neighbors (88.5 percent accuracy), and then LASSO regression (60.5 percent accuracy).

1 Introduction

Many astrophysical objects, such as stars and accretion disks, produce a variable amount of light, known as luminosity, over time. Changes in luminosity are often periodic and can reveal much about the physical properties that are occurring in the astrophysical object. By recording objects' luminosity over many years, we can create graphs of periodic variations in light, referred to as light curves. The light curves produced by different types of objects have unique characteristics, but many similarities within their own group. For example, some may have long (on order of years) or very short (on orders of seconds) fluctuations; some may large changes in luminosity whereas others may have small dips. These sources of light are all far enough away that we only see it as a point source of light, meaning that we need to look at the characteristics of the light from the object for clues as to what sort of object it is. As such, identifying variable sources lends itself well to machine learning classifier algorithms. The goal of this project is to classify the type of object emitting light based on the features of its light curve. Here, I will be exploring three algorithms to classify objects based on the characteristics of their light curves: LASSO regression, K-nearest neighbors, and neural networks.

2 Data

The light curves for this study come from the Sloan Digital Sky Survey Stripe 82 Variable Source Catalog (Ivezic et al. 2007). The catalog contains the observations of the amount of light that came from 67,506 stars and other space objects (primarily the jets of black holes) over a 5-year period from 2000-2005.

For objects in this catalog, the observed brightness of these sources changed over the 5-year period, many with a regular pattern. The causes of changing brightness are all physical and thus can tell astronomers much about the object that is emitting light. Common causes of variable sources are

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

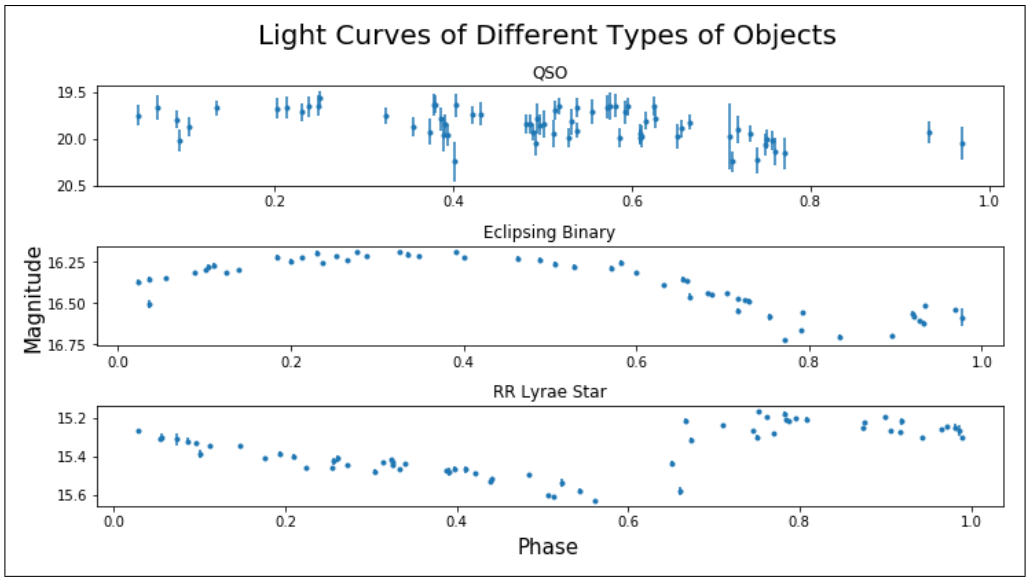

Figure 1: Example light curves for QSO, eclipsing binary stars, and RR Lyrae stars. These light curves have been folded so that periodic information can be seen. Notice differences in shape between each type, such as slope and how Gaussian they are.

Table 1: Data Groups

Category	Count
Eclipsing Binary	58
Eclipsing Variable	84
QSO	527
RR Lyrae	264
Delta Scuti	52
Other	15

eclipsing binary stars, astroseismic activity of large stars, transiting exoplanets, and rotating black holes. Each of these natural phenomena leave their own unique signature in an object's light curve. Figure 1 shows several example light curves.

Of the 67,506 objects in the catalog, 3,657 have been identified (Ivezic et al. 2007). They include eclipsing binary stars (broken up into 4 subtypes of eclipses), three types of variable star (RR Lyrae, Delta Scuti, and Cepheid Variables, each with 3 subtypes), eclipsing variable stars, and quasi-stellar objects (QSO, also known as black holes). QSOs comprise about ⅔ of the data, so have been randomly down-selected so each group has a similar number of members. This project uses 1,000 different light curves from 6 different types of astronomical objects, listed in Table 1.

3 Preprocessing

Light curves can be parameterized by a number of defining features, including photometric information (e.g., magnitude, color), general characteristics of the light curve (e.g., period, amplitude, harmonics), and the shape of the light curve (e.g., fits to various functions (like a sin wave), width, flux at various parts of the period, skew, kurtosis). To extract this information from the 1,000 light curves, I used the package Feature Analysis for Time Series package (FATS, Nun et al. 2015). FATS analyzes light curves to create a set of 64 features. The full list of features is given by Nun and Protopapas (n.d.). FATS was able to find values for each of the 64 features for every light curve.

In order to use these features with the below algorithms, I also standardized each feature in the matrix using Scikit Learn's StandardScale function (Pedregosa et al 2011). This step was important, as the values of features span many magnitudes. I then split the data into 5 equal parts, each with a

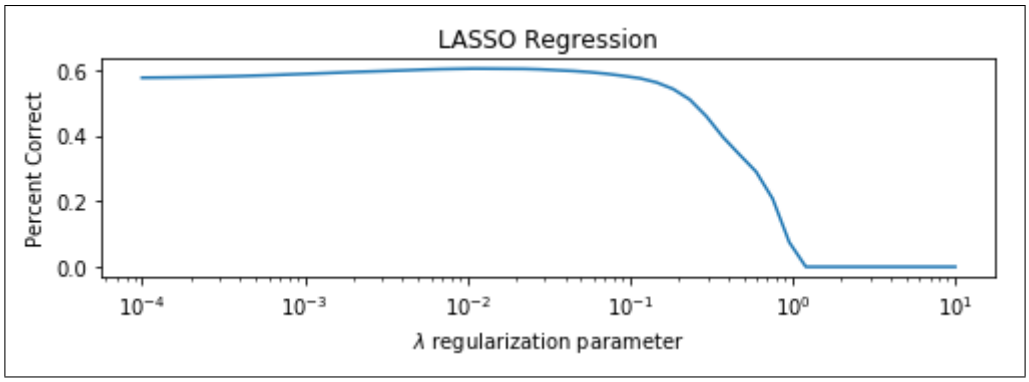

Figure 2: LASSO regression results for different regularization parameters.

Table 2: Data Groups

Feature	Coefficient
Color	0.41
Skew	0.26
Linear Trend	0.13
R_{cs}	0.12
Auto-correlation length	0.11
Freq1_harmonics_amplitude_2	0.092
Small Kurtosis	0.081
Stetson Kurtosis	0.08
Beyond1Std	0.066
Period fit	0.057

representative amount of each the 6 types of objects. These groups are used for 5-fold cross-validation for each of the algorithms below. I used Python's time package to time results below.

4 LASSO Regression

Due to the number of features, my first step was to determine which features are actually important to describing differences. To do this, I used Scikit Learn's LASSO regression (Pedregosa et al. 2011) to find a sparse solution. I evaluated regularization parameters from 10^{-4} to 10. Successful classification raters were calculated using 5-fold cross-validation. Figure 2 shows the results of this analysis. On average, each regression took 0.01 seconds.

The peak average success rate was 60.3 percent at $\lambda = 0.011$. Here, 30 of 65 features had non-zero weights, with the top 22 of these having coefficients between 0.01 and 0.41. Table 2 lists the 10 most important features and their weights. The most important feature is the color of the observation — the difference in luminosity in two band passes — which is based on the temperature of the object. Notably, most of the features describe the shape of the curve, especially how Gaussian (or non-Gaussian) a curve is, including skew, linear trend (slope of light curve, R_{cs} (range of cumulative sum, or how "spiky" a dataset is), small kurtosis, Stetson kurtosis, points beyond 1 standard deviation (Beyond1Std), and how well a Lomb-Scargle periodogram fits (Period fit) to the distribution. Auto-correlation length and Freq1_harmonics_amplitude_2 both concern the frequency of variation in the signal. From a scientific standpoint, these results show that the defining features of a light curve tend to be the shape of the curve, along with the temperature of the object and the frequency of variations, all of which make sense. I will be using these features in later analyses.

5 K-Nearest Neighbors

For the second classification algorithm, I investigated K-nearest neighbors. I used Scikit Learn's KNeightborsClassifier algorithm (Pedregosa et al. 2011). Based on my results from the LASSO

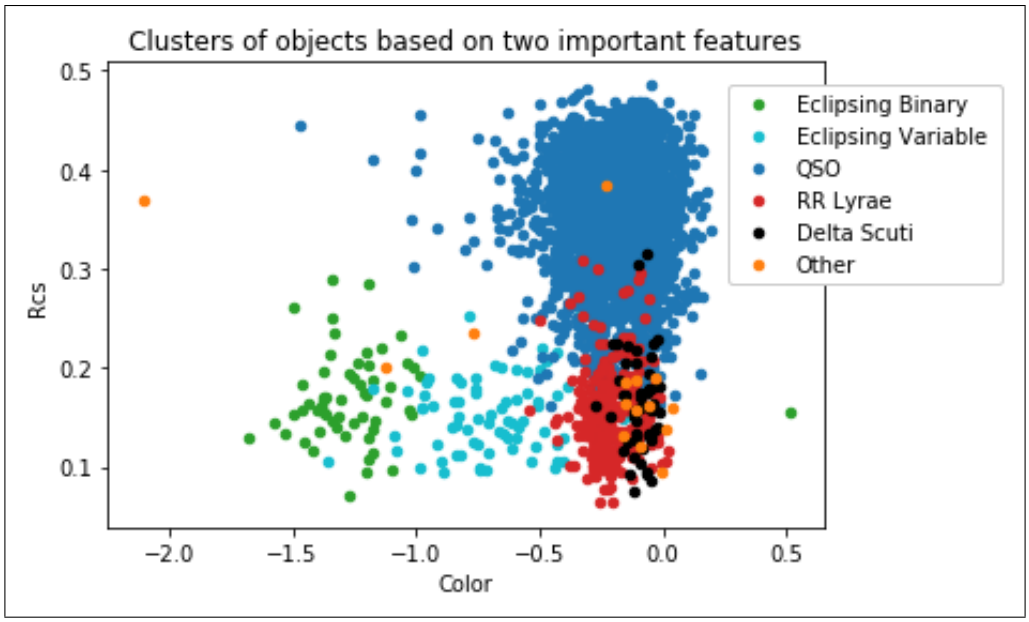

Figure 3: Clusters can be seen among two of the important features. Including the other 28 important features in the analysis gives an overall classification rate of 88 percent, primarily due to misclassification of "other" objects, which as seen here, have significant overlap with two classes.

regression, I only used the 30 features that I found to be significant. Like with LASSO, I performed a cross-validation and took the average error rate across each test. The K-nearest neighbors had an 88.5 percent success rate with a 3 percentage point standard deviation. This is a significant improvement compared to the LASSO regression. On average, each test took 0.02 seconds, twice the time of LASSO.

Notably, the algorithm performed best when it tried to form five clusters, rather than the six I gave it. In all but one case, the "other" category objects were grouped in with the defined categories. This might be explained by this being the smallest category by a significant amount, and many of these objects showing some variability like RR Lyrae and Delta Scuti stars (which the "other" objects tended to group with). Figure 3 shows the clustering for five groups in two dimensions (Color and R_{cs}) that were heavily weighted in LASSO. Although just 2 of 30 dimensions, one can see distinct groups emerge, and the overlap of the "other" category with RR Lyrae and Delta Scuti stars.

6 Neural Networks

The last algorithm I investigated was a neural network, for which I used Scikit Learn's Multi-layer Perceptron (Pedregosa et al. 2011). To choose the number of nodes and layers, I did both a depth search of number of nodes in one hidden layer and a breadth search across three hidden layers, using cross-validation to find the average score.

For the depth search in one layer, I investigated the results between 1 and 300 nodes. Every network with fewer than 11 nodes (and 5 with more than 11) had less than 90 percent successful classification. Beyond that, all other number of nodes in one layer had between 90 and 92.6 percent correct classifications. The best was 155 nodes, giving a correct classification rate of 92.6 percent. Generally, the more nodes, the better the algorithm performed, although there were several double digit number of nodes among the best solutions.

For the breadth search across 3 layers, I ran 3600 tests and varied the number of nodes in each layer by steps of 15 from 1 to 150 nodes. This meant that there were 15 combinations with only 1 layer, 225 combinations with 2 layers, and 3375 combinations with 3 layers. The best combination I found was 32 nodes in layer 1, 128 in layer 2, and 32 nodes in layer 3, with a correct classification rate of 92.3 percent, which is slightly less than the best rate I found with just 1 layer. Several 1-layer networks were among the top performers in the breadth search, with the best at 64. Notably, the best

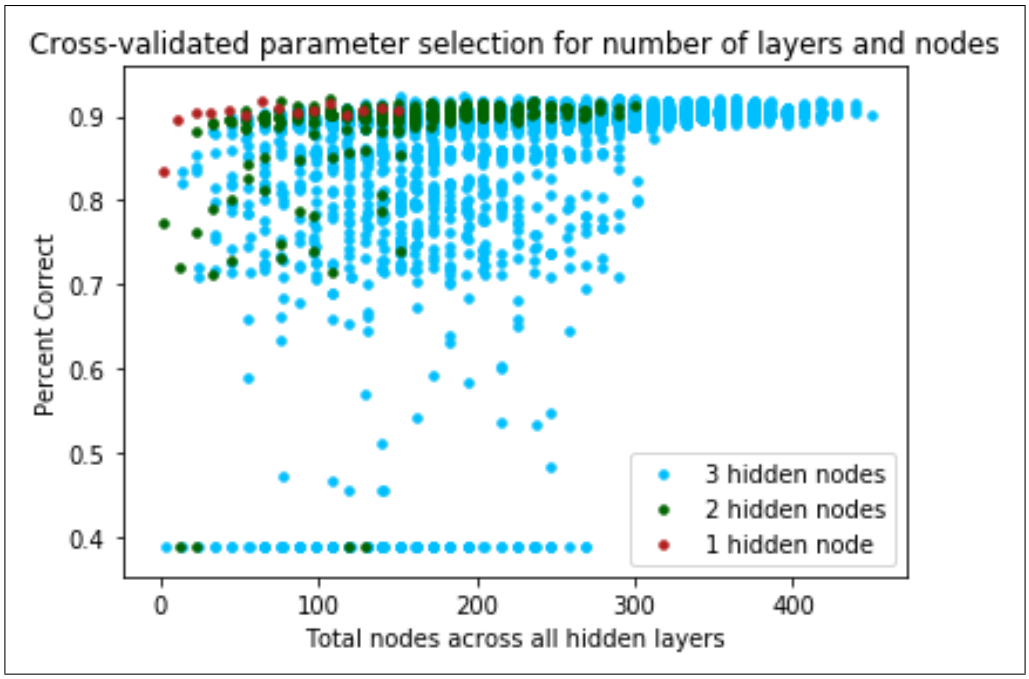

Figure 4: Results from 1-, 2-, and 3-layer tests with varying number of nodes.

Table 3: Results

Algorithm	Top Classification Rate	Parameters
LASSO Regression	60.3	$\lambda = 0.011$
K-Nearest Neighbors	88.5	5 clusters
Neural Network	92.6	155 nodes in 1 layer

3-layer network and best 1-layer network were multiples of 64, the number of features. Figure 4 shows the results of the breadth search. The "other" category was once again the most mis-classified. 1-layer networks took on average 0.1 seconds to run, 2-layer networks averaged 0.22 seconds to run, and 3-layer networks average 0.46 seconds to run, each a significant slow-down from earlier algorithms.

My results from the neural network tests show that single layer neural networks performed quite well for this data, although in general, the more nodes, the better the results. This makes sense, as more nodes can describe more complex boundaries. Although the best result was a 1-layer network, it is possible that there is a better performing network with 2 or 3 layers, which I missed due the step size of my search. In comparison to the earlier algorithms, neural networks performed the best. However, running these tests were much more time consuming than either of the other two, especially for multi-layered networks.

7 Conclusion

In my analysis, I found that neural networks were the best classifiers for light curve data, followed closely by k-nearest neighbors, and somewhat distantly by LASSO regression. Table 3 shows the peak performance for each algorithm, along with the relevant parameters. In terms of speed, both the LASSO Regression and k-nearest neighbors ran the fastest, followed by 1-layer neural networks, and then 2- and 3-layer networks. In terms of the trade-off between speed and accuracy, 1-layer neural networks seemed to perform the best, but do not take as long as multi-layered networks.

References

Ivezic, Z. et al. 2007. Sloan Digital Sky Survey Standard Star Catalog for Stripe 82: The Dawn of Industrial 1% Optical Photometry. Astrophysical Journal 134:973-998. http://faculty.washington.edu/ivezic/sdss/catalogs/S82variables.html

Nun, I., P. Protopapas, B. Sim, M. Zhu, R. Dave, N. Castro, and K. Pichara. 2015. Fats: Feature Analysis For Time Series. https://arxiv.org/pdf/1506.00010.pdf

Nun, I. and P. Protopapas, n.d. Feature Documentation. http://isadoranun.github.io/tsfeat/FeaturesDocumentation.html

Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 2011.