Classification of Poorly Time Sampled Light Curves of Periodic Variable Stars

James P. Long, Joshua S. Bloom, Noureddine El Karoui, John Rice, and Joseph W. Richards

1 Introduction

Classification of variable stars is important for scientific knowledge discovery and allocation of resources for telescopic follow up. With increasing amounts of data, it has become difficult for this task to be done manually. Future surveys will make it even harder. For instance, GAIA is expected to discover $\sim$15 million periodic variables over the course of a five year mission [6]. Tools from the machine learning and statistics communities will be critical for processing and understanding this data. Several recent papers have used machine learning methods to aid in classification e.g. [8, 3, 5]. These studies have focused on classifying fairly well sampled light curves. Evaluation of classifier performance has been made relative to light curves of similar quality, usually from the same survey.

In practice, light curves needing classification (the test set) will be labeled using classifiers constructed on data which has already been labeled (the training set). Systematic differences in cadence, observing region, flux noise, detection limits, and number of flux measurements per light curve may exist between training and test sets. Thus it is not clear that a classifier with good performance on the training set will have good performance on the test set. These problems were noted in [4, 8]. In this paper we focus specifically on the challenge of constructing a classifier on well sampled light curves ($\geq$ 200 flux measurements) to classify poorly sampled light curves ($\leq$ 100 flux measurements). This situation is likely to be encountered when catalog data with well sampled light curves is used to classify data from ongoing

James P. Long
Statistics Department, University of California, Berkeley, CA 94720-3860
e-mail: jlong@stat.berkeley.edu

Joshua S. Bloom
Astronomy Department, University of California, Berkeley, CA 94720-7450

Noureddine El Karoui, John Rice, Joseph W. Richards
Statistics Department, University of California, Berkeley, CA 94720-3860

surveys where light curves have only tens of flux measurements. In this setting, metrics computed on the light curves and used for classification, termed *features*, may contain error for the test set. A classifier constructed on the training set that does not recognize that some features contain error in the test set may be suboptimal. Figure 1 illustrates the problem.

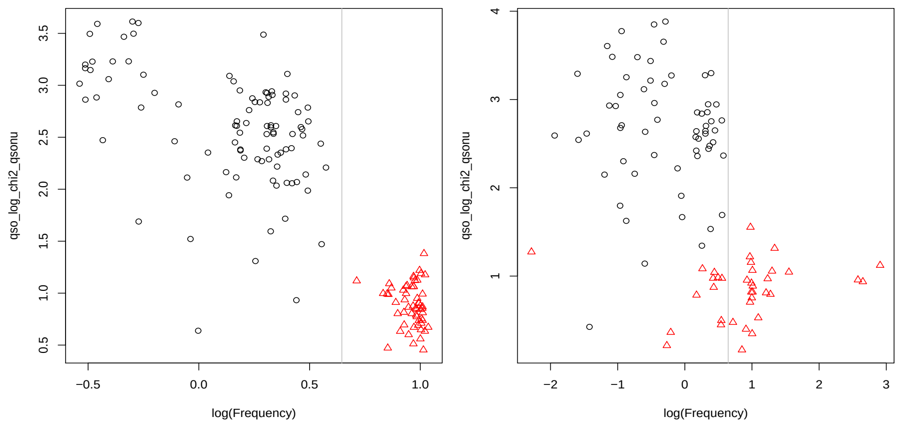

Fig. 1 The left plot shows two features, *log(frequency)* and *qso_log_chi2_qsonu* for well sampled RR Lyrae Double Mode (red triangles) and Multiple Mode Cepheid (black circles) light curves from OGLE. The right plot shows the same two features for the same sources when light curves have been truncated at the first 30 flux measurements. The right plot is different from the left due to error in estimating *frequency* and *qso_log_chi2_qsonu* for light curves with 30 flux measurements. The vertical gray line was chosen by the statistical classifier CART (a popular classifier, see [2]), trained on the well sampled curves, to separate the two classes. It works well on the well sampled curves but does poorly on the poorly sampled curves. A classifier that uses *qso_log_chi2_qsonu* to make a horizontal split will likely improve performance on the poorly sampled curves.

This paper is organized as follows. In Sect. 2 we discuss how to apply statistical / machine learning classifiers to astronomical light curves, data assumptions made by these classifiers, and two frameworks — *noisification* and *denoisification* — for overcoming the problem of training on well sampled curves with the goal of classifying poorly sampled curves. We describe an application of noisification and denoisification to two data sets — simulated and OGLE — in Sect. 3. In Sect. 4 we discuss results and challenges associated with each method. Finally we offer conclusions and suggestions for future development of this work in Sect. 5.

2 Methods

There are many methods available for performing statistical / machine learning classification. Some of the most popular include Random Forests, Support Vector Machines (SVM), Neural Nets, and Gaussian Mixture models (see [7] for an overview of these and others). Direct application of classifiers to irregularly sampled light curves (i.e. the raw data) is challenging so for each light curve we compute a set of metrics, called features. Features are chosen so as to separate light curves belonging to different classes. Features we use include estimates of frequency, standard deviation of flux measurements, and amplitude of flux measurements. We compute 62 features for each light curve. See [8] for descriptions of each feature. We call the process of converting a light curve to a feature vector *deriving features*. With each light curve represented by a class label and feature vector, standard classification methods can be applied to construct the classifier. In this work we use the Random Forest classifier, discussed in detail in [1]. However the problem of systematic differences between training and test sets and our proposed solutions hold quite generally for essentially any classification method.

Classifiers nearly always assume that the relationship between class and features is the same for the training and test sets. Specifically, if $\mathbf{x}$ is a vector of features from the training set, $\mathbf{y}$ is a vector of features from the test set, and z is a possible class, then when $\mathbf{x} = \mathbf{y}$ the probability of class z given features $\mathbf{x}$ or features $\mathbf{y}$ must be the same, i.e. $p(z|\mathbf{x}) = p(z|\mathbf{y})$. As demonstrated in Fig. 1, this may not hold when light curves in the training set have hundreds of flux measurements and light curves in the test have, say 30, due to error in estimating features in the test light curves. Violation of the assumption $p(z|\mathbf{x}) = p(z|\mathbf{y})$ will invalidate estimates of misclassification error provided by cross validation or any other method. Perhaps more importantly, the classifier constructed without modification on the training set may be suboptimal because the relationship between features and class is fundamentally different in the training and test data. We now discuss two methods, *noisification* and *denoisification*, for solving the problem.

2.1 Noisification

The principle behind noisification is simple: make the training data “look like” the test data and use the modified training data to construct a classifier. Specifically, if a test light curve has 40 flux measurements, truncate every light curve from the training set at 40 flux measurements, derive features for these 40 flux measurement light curves, and construct a classifier using these features. We call this method *noisification* because feature noise is added to the training set (by way of the truncation process) in order to match the probability of class given test features with probability of class given noisified features. Here we are assuming that the only difference between light curves in the training and test sets is the number of flux measurements observed. Cadence, flux error, and survey observing characteristics are assumed to be the same.

As stated, noisification requires constructing a different classifier for each possible number of flux measurements a light curve in the test set might have. This is computationally expensive and perhaps quite redundant since truncating training light curves at say, 40 and 41 flux measurements, deriving features for each, and then constructing two classifiers will result in two very similar classifiers. We explore this issue and possible computational savings in Sec. 4.

Truncating the training data at the first 40 flux measurements is inherently arbitrary. We could select any 40 contiguous flux measurements, or better yet repeat the entire noisification process B times, selecting different sets of 40 flux measurements and averaging the results of the B classifiers. In this work we repeat the process five times i.e. $B = 5$. Each noisified classifier outputs a vector of class probabilities which we average across the five runs before selecting the class with the highest posterior probability.

2.2 Denoisification

With denoisification, we first construct a classifier on the well sampled training data. Given a test light curve with features $\mathbf{y}$ we infer what the features for this curve would be if we had continued observing it for several hundred flux measurements. The classifier then uses these inferred features to predict a class for the test light curve. The process of inferring features is denoising, hence the name *denoisification*. More formally, in trying to estimate $p(z|\mathbf{y})$ we reframe the problem

$$\begin{aligned} p(z|\mathbf{y}) &= \int p(z|\mathbf{y},\mathbf{x})p(\mathbf{x}|\mathbf{y})\mathrm{d}\mathbf{x} \\ &= \int p(z|\mathbf{x})p(\mathbf{x}|\mathbf{y})\mathrm{d}\mathbf{x} \\ &= \frac{\int p(z|\mathbf{x})p(\mathbf{y}|\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}}{p(\mathbf{y})} \,. \end{aligned} \tag{1}$$

$p(z|\mathbf{x},\mathbf{y}) = p(z|\mathbf{x})$ because once we know the features estimated on a well sampled curve, $\mathbf{x}$, the features from the poorly sampled version of this curve, $\mathbf{y}$, do not give us any further information about the class of the light curve. $p(z|\mathbf{x})$ is estimated using the classifier constructed on the unmodified training data. Inferring the true features for a poorly sampled light curve occurs in estimating $p(\mathbf{x}|\mathbf{y})$. We interpret this quantity as given features $\mathbf{y}$ derived for a poorly sampled light curve, $p(\mathbf{x}|\mathbf{y})$ is the likelihood that if we continued observing this curve for hundreds of flux measurements the derived features would be $\mathbf{x}$. Directly estimating this quantity is challenging so we rewrite it as $p(\mathbf{y}|\mathbf{x})p(\mathbf{x})/p(\mathbf{y})$ in the last line of the equation. This suggests that we can denoise $\mathbf{y}$ by estimating the quantities $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x})$.

There are many possibilities for estimating $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x})$ in (1). We take the following approach. Let $\mathbf{x_1},\ldots,\mathbf{x_n}$ be derived feature vectors from the training set of size n. $\mathbf{x_i} = (x_i^1,\ldots,x_i^p) \in \mathscr{R}^p$. We have derived p features for each light curve. Truncate the training light curves to match the length of test light curves and rederive features. Denote these feature vectors by $\mathbf{y_1},\ldots,\mathbf{y_n}$. Here $\mathbf{y_i} = (y_i^1,\ldots,y_i^p)$. We model the relationship between a truncated feature y_i^j from light curve i and the feature vector from the well sampled version of light curve i, $\mathbf{x_i}$, as,

$$y_i^j = g_j(\mathbf{x_i}) + \varepsilon_{j,i} \tag{2}$$

The $\varepsilon_{j,i}$ are assumed to be independent normal, mean 0, variance σ_j^2. g_j is arbitrary. Using $\{(\mathbf{x_i}, y_i^j)\}_{i=1}^n$ estimate g_j. Denote this estimate $\hat{g}_j$. We use Random Forests *regression* to compute $\hat{g}_j$. We now estimate $p(\mathbf{y}|\mathbf{x})$ according to our model.

$$\hat{p}(\mathbf{y}|\mathbf{x}) = \quad \prod_{j=1}^{p} \hat{p}(y^j|\mathbf{x}) \tag{3}$$

$$= \prod_{j=1}^{p} \phi\left(\frac{y^j - \hat{g}_j(\mathbf{x})}{\hat{\sigma}_j}\right) \tag{4}$$

ϕ denotes the standard normal density. Plugging into (1) we obtain,

$$\hat{p}(z|\mathbf{y}) = \frac{\frac{1}{n}\sum_{i=1}^{n} \hat{p}(z|\mathbf{x_i})\hat{p}(\mathbf{y}|\mathbf{x_i})}{p(\mathbf{y})} \tag{5}$$

It is clear that for our problem denoisification involves many more steps and assumptions than noisification. Much of this results from us not having a closed form expression for $p(\mathbf{y}|\mathbf{x})$. As a result we must model this relationship using the training data. Predicting a vector response is inherently challenging so we made the independence assumption in (2) to turn the problem into predicting several univariate responses. Even then, estimating $p(y_j|\mathbf{x})$ requires a non parametric regression method.

3 Data and Application of Methods

We study the performance of noisification and denoisification by applying them to the two data sets outlined in Table 1. The simulated data offers a highly controlled environment for examining our methods. The light curves here were sampled 0–2 times per night using a jittered sampling model. There are no incorrectly labeled curves and all light curves in the training set have exactly 200 flux measurements. OGLE provides a more realistic setting that might be encountered when noisification and denoisification are applied to classify light curves of variable quality. We note that some of the light curves in the OGLE training set do not meet our definition of well sampled light curves ($\geq$ 200 flux measurements). The shortest light curve in OGLE training has 131 flux measurements. Over three-fourths of the 358 OGLE training light curves have more than 200 flux measurements, so the well sampled training – poorly sampled test dichotomy is present here.

After splitting each data set into training and test, we downsample the test set to mimic poorly sampled curves. This process is accomplished by taking the test light curves and truncating them at $10, 20, \ldots, 100$ flux measurements, starting from the first flux measurement in each curve. We now have 10 test sets, each containing poorly sampled light curves of different quality. Noisification and denoisification are applied to each of these 10 test sets. Additionally we compare the two methods to the naive approach of training a classifier on the training set and applying it unmodified to the test data.

Table 1 Data set characteristics

Survey	Source Classes[a]	F / LC[b]	Train Size	Test Size
Simulated	RR Lyrae, Cepheid, β Persei, β Lyrae, Mira	200-200	500	500
OGLE[c]	RR Lyrae DM, Multiple Mode Cepheid, β Persei, β Lyrae, W Ursae Majoris	261-474	358	165

[a] For simulated data set, class prototypes were modeled on these source classes.
[b] F / LC is the first and third quartiles of number of flux measurements per light curve for training
[c] Data from [8]

4 Results

The left panels of Figs. 2 and 3 show the performance of noisification and denoisification on the simulated and OGLE data sets respectively. Both methods improve on the naive approach. As expected, the benefits of noisification and denoisification are most apparent for very poorly sampled curves. Once test light curves have 60 - 70 flux measurements, modifying the classifier does not deliver much performance improvement. Noisification (blue-plus) outperforms denoisification (orange-triangles) in these examples. This may be due to the assumptions made in implementing the denoisification process. The ease of implementing noisification and its overall good performance make it quite attractive relative to denoisification at this point. The error rates for the simulated data (left panel Fig. 2) decrease more smoothly than error rates for OGLE (left panel Fig. 3). We suspect this is due to the highly controlled setting in the simulated data. Light curves in OGLE often have large gaps meaning a set of 40 flux measurements may contain 30 flux measurements taken over two months and then 10 more flux measurements taken a year later.

As mentioned in Sect. 2.1, noisification requires constructing a different classifier for every set of light curves with a different number of flux measurements. The right panels of Figs. 2 and 3 explore how necessary this is. Here the noisified classifiers constructed for 10, 50, and 100 point test light curves are fixed and applied to each of the 10 test sets. The 50-point and 100-point noisified classifiers perform well across a wide range of flux measurements (30 - 100 flux measurement test sets) in both the simulated and OGLE data. For these data sets, noisified classifiers do not appear sensitive to small changes in the number of flux measurements per light curve. In other settings where this holds, computational savings could be acheived by noisifying the training set to several number of flux measurements per curve. Then new poorly sampled light curves could be classified using the noisified classifier that had the closest number of flux measurements per curve.

Figure 4 provides insight into how noisification is affecting classifier construction. The left panel contains density estimates of source frequency for the unmodified training and 40 flux measurement test sets from the simulated data. We see that these densities are quite different. The modes (bumps) in the training density result from frequency clustering by class. A classifier constructed on the unmodified training data will use the modes in frequency to separate the classes even though

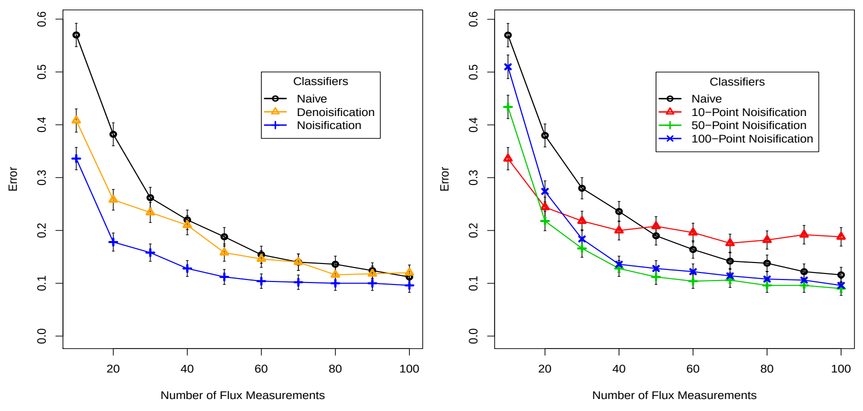

Fig. 2 Results for the simulated data. The left panel compares noisification, denoisification, and naive approaches to classifying poorly sampled test light curves. The right panel explores how robust noisified classifiers are to changes in the number of flux measurements in the test set.

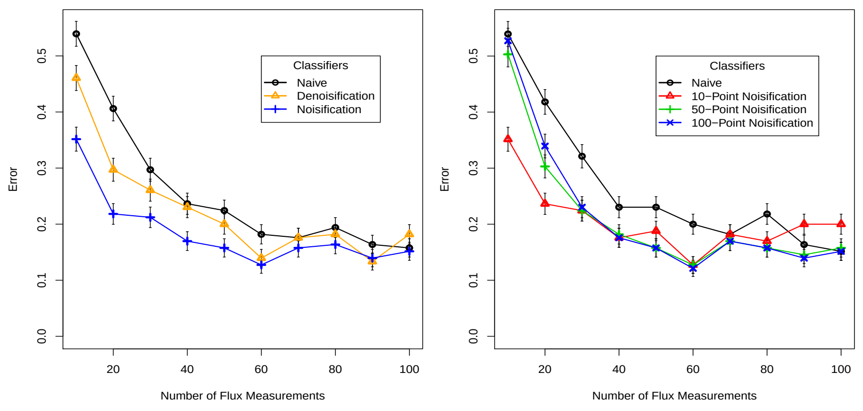

Fig. 3 Results for the OGLE survey. The left panel compares noisification, denoisification, and naive approaches to classifying poorly sampled test light curves. The right panel explores how robust noisified classifiers are to changes in the number of flux measurements in the test set.

the density of frequency in the test data has very different properties. In the right panel, the training data has been noisified. Now the frequency densities are closer. A classifier constructed on the noisified training data will capture the class-frequency relationship as it exists in the test set.

5 Conclusions

Noisification and denoisification improve classifier performance when using well sampled light curves to classify poorly sampled ones. While noisification is simpler

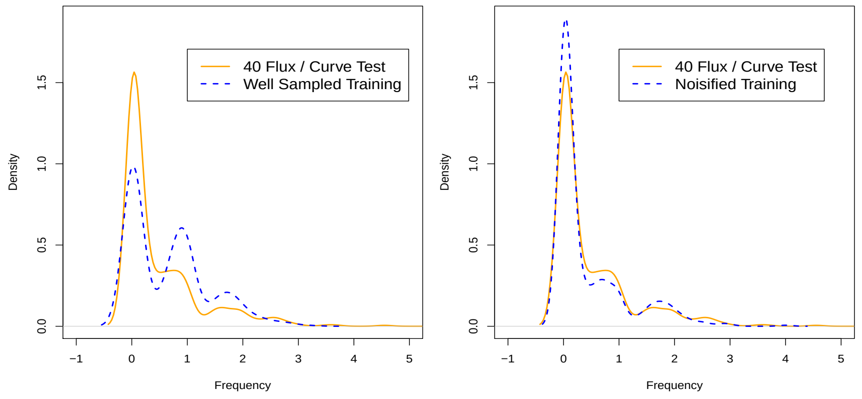

Fig. 4 The left panel displays the density of frequncy for training (blue-dash) and 40 flux test (orange-solid) from the simulated data. The right panel compares the density frequency in the test data to density of frequency in the training after noisifying.

to implement and gave better performance in this study, we believe there are reasons to continue studying the denoisification strategy. For one, an implementation of denoisification that used fewer, or more science-based, assumptions on $p(\mathbf{y}|\mathbf{x})$ might improve performance. Second, denoisfication can make use of unlabeled training data in the estimation of the integral in (1), providing a potential performance boost not available to noisification. Finally, with denoisification we only construct one classifier which can be re-used on any new test observation.

Greater understanding of noisification and denoisification will come from applying the methods to more data sets. In future work we plan to study light curves from the *Hipparcos* and ASAS surveys. While we have only studied noisification and denoisification in the context of poorly sampled light curves, both strategies could in principle be used to overcome other systematic differences between training and test sets such as noise in flux measurements, censoring of flux measurements, and varying cadences. Investigation of these issues will aid in construction of accurate classifiers which must be trained and tested on different surveys.

Acknowledgements The authors acknowledge the generous support of a Cyber-Enabled Discovery and Innovation (CDI) grant (No. 0941742) from the National Science Foundation. This work was performed in the CDI-sponsored Center for Time Domain Informatics (http://cftd.info).

References

1. Breiman, L.: Random Forests. Machine Learning **45**(1), 5–32 (2001)
2. Breiman, L., Freidman, J., Olshen, R., Stone, C.: Classification and regression trees. Wadsworth (1984)

3. Debosscher, J., Sarro, L., Aerts, C., Cuypers, J., Vandenbussche, B., Garrido, R., Solano, E.: Automated supervised classification of variable stars. Astronomy and Astrophysics **475**(3), 1159–1183 (2007)
4. Eyer, L., et al.: The variable universe through the eyes of GAIA. arXiv:1011.4527v1 (2010)
5. Eyer, L., Blake, C.: Automated classification of variable stars for All-Sky Automated Survey 1–2 data. Monthly Notices of the Royal Astronomical Society **358**(1), 30–38 (2005)
6. Eyer, L., Cuypers, J.: Predictions on the number of variable stars for the gaia space mission and for surveys such as the ground-based international liquid mirror telescope. In: IAU Colloq. 176: The Impact of Large-Scale Surveys on Pulsating Star Research, vol. 203, pp. 71–72 (2000)
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Verlag (2009)
8. Richards, J., Starr, D., Butler, N., Bloom, J., Brewer, J., Crellin-Quick, A., Higgins, J., Kennedy, R., Rischard, M.: On machine-learned classification of variable stars with sparse and noisy time-series data. The Astrophysical Journal **733**, 10 (2011)