PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE

SCHOOL OF ENGINEERING

AUTOMATIC CLASSIFICATION OF POORLY SAMPLED VARIABLE STARS

NICOLÁS PABLO CASTRO LEAL

Thesis submitted to the Office of Research and Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science in Engineering

Advisor:

KARIM PICHARA

Santiago de Chile, September 2016

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE

SCHOOL OF ENGINEERING

AUTOMATIC CLASSIFICATION OF POORLY SAMPLED VARIABLE STARS

NICOLÁS PABLO CASTRO LEAL

Members of the Committee:

KARIM PICHARA

PAVLOS PROTOPAPAS

TOMÁS REYES

PATRICIA SÁNCHEZ-BLAZQUEZ

MIGUEL NUSSBAUM

Thesis submitted to the Office of Research and Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science in Engineering

Santiago de Chile, September 2016

To my family, friends and Kai

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisors Karim Pichara and Pavlos Protopapas for helping me throughout this whole process. For always having their doors open, founding the time to hear me out, do some brainstorming and encourage me when my motivation was running low.

Then, I would like to thank my family and girlfriend, for always believing in me and supporting me, not only in this work but in every decision I make in my life.

I would also like to thank my classmates and lab partners Cristóbal Mackenzie and Andrés Riveros. For being good friends, always sharing there knowledge with me, and also hearing my ideas and doubts.

Finally I would like to thank all the staff from the Computer Science Department. They work very hard to keep everything running smoothly, always with a smile in their faces and a warm salute.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Automatic classification methods applied to sky surveys have revolutionized the astronomical target selection process. Most surveys generate a vast amount of time series, or "lightcurves", that represent the brightness variability of stellar objects in time. Unfortunately, astronomical data take several years to be completed, producing partial time series that generally remain without analysis until the observations are completed. This happens because state of the art methods rely on a variety of statistical descriptors or features that present an increasing degree of dispersion when the number of observations decreases, which reduces their precision. In this paper we propose a novel method that increases the performance of automatic classifiers of variable stars by incorporating the deviations that scarcity of observations produces. Our method uses Gaussian Process Regression to form a probabilistic model of the values observed for each lightcurve. Then, based on this model, bootstrapped samples of the lightcurves' features are obtained. Finally a bagging approach is used, based on this samples, to improve the overall performance of the classification. The output of our model is a classification vector that associates a probability of belonging to different variability classes. We realized tests on the MACHO and OGLE catalogs; the results show that our method effectively improves the classification performance of standard models. We believe this results prove that, when studying lightcurves, it is important to consider the features' error and how the measurement process impacts it.

Keywords: Astronomy, Variable Stars, Machine Learning, Data Mining

RESUMEN

La aplicación de métodos de clasificación automática en catálogos de observación astronómica ha revolucionado el proceso de identificación de estrellas. Hoy en día, muchos estudios generan catálogos conformados por un gran numero de series de mediciones, o ”curvas de luz”, que representan los cambios en el brillo de objetos estelares en el tiempo. Desafortunadamente, las observaciones toman varios años en completarse, lo que produce series de tiempo parciales que normalmente no son analizadas hasta que todas las observaciones son completadas. Esto sucede porque los métodos de clasificación más modernos dependen de una variedad de descriptores estadísticos que presentan un grado creciente de dispersión a medida que el numero de observaciones decrece, lo que disminuye su precisión. En este trabajo, proponemos método que mejora el rendimiento de los clasificadores automáticos de estrellas variables al incorporar las desviaciones producidas por la escasez de observaciones. Nuestro algoritmo utiliza Procesos Gaussianos de regresión para formar un modelo probabilístico de los valores observados para cada curva de luz. Luego, basado en este modelo, se generan muestras aleatorias de los descriptores de las curvas. Finalmente, a partir de estas muestras, se utiliza una técnica de bagging para incrementar la precisión de la clasificación. El resultado de este modelo, es un vector de clasificación que representa la probabilidad de pertenecer a cada una de las posibles clases de estrellas variables. Realizamos pruebas en los catálogos MACHO y OGLE; los resultados muestran que nuestro método logra mejorar las predicciones de modelos clásicos. Consideramos que estos resultados muestran la importancia de tomar en cuenta el error de

los descriptores estimados, al clasificar curvas de luz, y como los procesos de observación los impactan.

Palabras Claves: Astronomía, Estrellas Variables, Aprendizaje de Máquina, Minería de Datos

1. INTRODUCTION

1.1. Machine Learning in Astronomy

We are currently living an age of technological revolution, the advances in different areas of science are happening at a rate that has never been seen before. The case of information technologies is particularly surprising. The amount of data present in the world is already so big, that it exceeds any chance of being analyzed with human supervision. This creates the necessity for superior algorithms capable of making complex inferences and human like analysis, with the speed of a computer. This is why the area of machine learning appears, as an intent to close the gap between computer logic and human reasoning.

The area of astronomy is a clear example of this data deluge. Modern synoptic sky surveys, are projects that observe giant portions of the sky for long periods of time (some times over ten years), with no specific target in mind. This way they record information of millions of objects at the same time (Alcock et al., 2001; Aubourg et al., 1993; Udalski et al., 2008; Drake et al., 2009), which result in very valuable information, but also in an immense amount of data that is increasing exponentially. While some of the first synoptic surveys like MACHO (Alcock et al., 2001) recorded around 10 terabytes of data over its life span, newest ones like the LSST (Matter, 2007) are expected to record over 100 Petabytes of data.

In order to analyze the data obtained from the surveys, extensive procedures must be performed first. From this need, a particular field of astronomy has emerged, called Time Domain Astronomy (TDA). TDA studies astronomical objects and phenomena that change trough time (Huijse et al., 2014).

Among the many tasks TDA works on, automatic classification of variable stars through lightcurve analysis has been heavily studied (Debosscher et al., 2007; Wachman et al., 2009; Kim et al., 2009; Wang et al., 2010; Richards et al., 2011; Bloom & Richards, 2011; Kim et al., 2011; Pichara et al., 2012; Bloom et al., 2012; Pichara & Protopapas, 2013; Kim et al., 2014; Nun et al., 2014; Mackenzie et al., 2016; Pichara et al., 2016). This task aims to identify certain specific and valuable type of objects, within the hundred of thousands a sky survey may contain, so later they can be analyzed with greater scrutiny.

In this line, supervised learning techniques have proved to be particularly effective, due to their precision and speed (Debosscher et al., 2007). This kind of tools, train classification models over a group of labeled objects, for example, a significant group of stars whose specific variability type has been determined trough spectroscopy. The training process seeks to teach models to recognize underlying patterns that allow them to separate among a set of variability classes. These patterns can be very complex and high dimensional relations. Fortunately, Machine learning approaches have shown capabilities to discover very complex underlying patterns, that are imperceptible for human beings (Jiawei & Kamber, 2001). The training phase can be very demanding in computational time, but once the models are ready, the process of classifying a new instance is extremely fast. And although this models have proved to be very effective, non of them is 100%

accurate, and thus the development of new algorithms and analysis techniques is still an open problem

1.2. Contribution of this thesis

For the task of automatic classification, lightcurves are represented as a vector of statistical features that describe different aspects of them, like brightness variability, color, periodicity, and auto-correlation, among others(Richards et al., 2011; Pichara et al., 2012; Nun et al., 2015). However, the value of those features is highly dependent on the quality of the measurements on which they are calculated (Kirk & Stumpf, 2009). Inherent errors in the values of photometric time-series, as well as the amount of observations, may affect the values of their descriptors. Therefore, the errors committed by classifiers in their predictions can be attributed, at least in part, to the lack of precision of the features used to represent them. For example, an insufficient amount of observations in lightcurves may result in wrong estimation of periods, in spurious auto-correlations values, or in poorly calculated variability patterns.

Figure 1.1 shows the value of three different features, for two different lightcurves, calculated at different moments of the observation process. The values of each individual feature has been normalized and centered around zero, in order to make the variations comparable. It is not surprising that the values change considerably as the number of observations increases, but it is worth noting that stronger changes occur at the beginning, when the number of observations is smaller. This holds for any statistical estimate. What is particularly interesting is that this effect is not consistent for different features and for

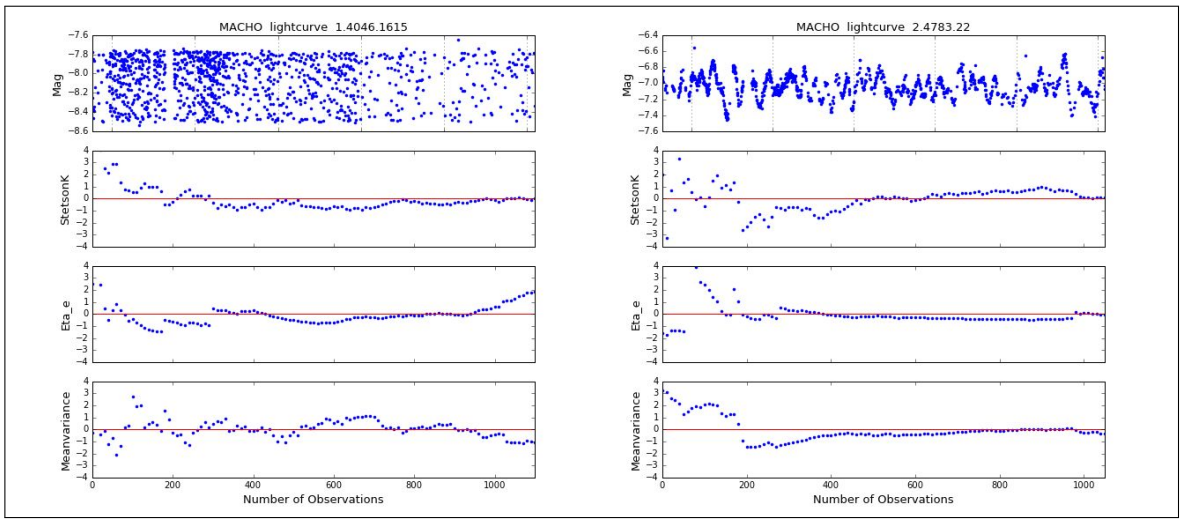

FIGURE 1.1. Normalized feature values over time. Features do not always tend to a specific value (as shown by the mean) as the number of observations increases. Also, not all features converge at the same time.

different lightcurves. In fact, it is easy to find cases where the same feature takes longer to stabilize than others or even cases where features do not appear to converge at all.

The implications of this fact are evident. Photometric lightcurves are noisy, non homogeneously measured, with differences in the number of observations among them, and with several observational gaps within them. If the value of the features used to describe them are not robust and based on long periods of time, then they vary considerably as more observations are added. This kind of features are not reliable to perform classification. In the case of ongoing surveys, the problem is even bigger. The shorter the time series being analyzed, the scarcer the information it contains, in fact, not even an expert astronomer can correctly classify a lightcurve that consists of only a couple of measurements. This matter is of utmost importance because photometric sky surveys normally take various years to be completed. And, although in some cases, the data may not be sufficient to make good

use of it, it would be very useful to have a model that is able to distinguish when there is enough information to make reliable predictions and when not.

In order to tackle the problem of automatic classification with "incomplete" lightcurves we work with two real sky surveys (MACHO and OGLE), where sets of approximately five thousand labeled lightcurves of each catalog are used. Then, the curves are shortened into smaller versions of themselves by selecting only few observations. This way we simulate the scenario of surveys that are barely beginning their measurement process.

Features that are calculated over some statistical sample are often assigned some measure of accuracy (Street et al., 1993; Efron & Tibshirani, 1994). For simple features like the mean or the standard deviation, closed form equations exist for the associated error. Unfortunately this is not the case for the majority of the time series features used for classification. In the cases where closed form equations are not available, bootstrapping techniques are an adequate alternative (Efron & Tibshirani, 1994). This techniques allow to assign measures of accuracy to any statistical quantity by doing random subsampling of the data where it is estimated.

In this investigation, a parametric time series bootstrapping technique is proposed in order to generate many different lightcurve samples from the training set. Then, various random training sets are built from this samples, where an automatic classifier is trained on each of them. This approach allows to overcome the different biases each training object may posses in its feature values, by averaging over the predictions among different random models.

The objective of this work is to demonstrate the advantages of taking into account the error present in the statistical features used for classification. And show how that error relates to the quality of the time series used for classification. The framework presented in this work proves that valuable predictions can be made with very poor time series conformed by few observations.

1.3. Overview of this thesis

The rest of this thesis is organized as follows: Chapter 2 explains the relevant background theory; Chapter 3 shows a small review of the work done in supervised classification in astronomy and bootstrapping techniques for order dependent data and Chapter 4 gives a precise description of the method presented. Within Chapter 5, Section 5.1.1 presents a real dataset used for a synthetic experiment, the lightcurves catalogs and the training set used for the experiments. In Section **??** the results obtained are presented. Finally, the conclusions of this work are presented in Chapter 6

2. BACKGROUND THEORY

2.1. Astronomical Background

Sky surveys equip astronomers with massive amounts of information for them to analyze and make discoveries, even for several years after this are completed. Part of this information comes in the form of photometric catalogs. This are periodical measurements of the brightness of several million of objects over the time the catalog was constructed. One of the first and most important analysis that can be made is to classify this amazing number of objects into different type of variability classes. What follows is a brief description of the most important objects that can be detected in sky surveys.

2.1.1. Variable Stars

Among the group of observable stars there is a group called ”variable stars”, which are of particular interest for astronomers. As described in Huijse et al. (2014), these are stars whose brightness fluctuates in time over a certain threshold, defined by the observing instruments. Analysis of this type of stars is fundamental for the study of stellar structure and properties, stellar evolution and the distribution and size of the Universe.

Variable stars can be divided in two main groups: intrinsic and extrinsic. On one hand intrinsic variable stars get there luminosity variation from internal physical changes, such as fluctuation in size or temperature. For example, Cepheids are radially pulsating supergiant stars that expand and contract periodically. On the other hand, extrinsic variable stars get their variation due to external physical influences, such as other moving objects

around them. This is the case of eclipsing binaries stars, which are actually a system of two stars rotating around each other, with their orbital plane aligned with the earth.

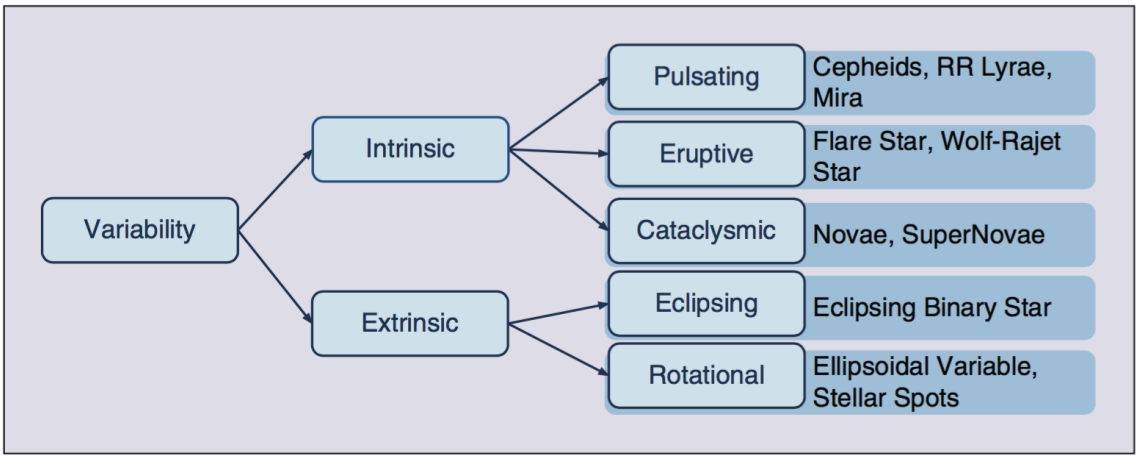

FIGURE 2.1. Variable star topological classification as in Huijse et al. (2014). Intrinsic variable stars get there luminosity variation from internal physical changes, whereas extrinsic variable stars get their variation due to external physical influences.

Two particularly interesting intrinsic variable stars are RR Lyrae and Cepheids. This stars can be used as distance markers in the universe, due to the relation between their pulsation period and their absolute brightness (Percy, 2007).

2.1.2. Variable Non-stellar Phenomena

Beside variable stars, there are other kind of phenomena that are perceived as lightcurves with brightness fluctuation. These phenomenon are scarce, but are also very important to identify. For example, a gravitational lensing effect is an increase of several order of magnitude in the brightness of an object. This is produced by the presence of dark matter between the earth and the object that emits the light. This big amount of matter acts as a giant lens that distorts the light on its way to earth. When the amount of dark matter is

similar to that found in a planet, this are called microlensing events. Finding this kind of objects is the main goal of some sky surveys (Aubourg et al., 1993; Udalski et al., 2008; Alcock et al., 2001).

2.2. Machine Learning Background

Machine learning is an area of computer science that evolved from the broader topic of artificial intelligence. Its goal is to allow computers to learn from experience, so that they can later make decisions without being specifically programed for them. In order to do this, a set of example inputs, from the phenomenon of interest, is required for the algorithms to train on. Then the algorithm tunes some kind of model (a mathematical function or data structure for example), over this so called training set. Machine learning algorithms can solve many different problems, some of the most important ones are: classification, regression, outlier detection, clustering and extraction of association rules. The work of this thesis falls under the category of automatic classification methods and, as such, of supervised learning.

2.2.1. Supervised Learning

Supervised learning refers to all of machine learning algorithms where the model is learned from data whose desired output is known. In other words, the data is conformed by a group of instances $\mathcal{X} = \{\mathbf{x}, y\}$, where $\mathbf{x}$ is a vector of input variables and y is the variable which we want to predict. The case where y is categorical is known as classification, whereas the case where y is a continuos variable is referred as regression. One of the most

popular and simple classifiers is the decision tree (Breiman et al., 1984; J. R. Quinlan, 1986). This classifiers aim to build a tree structure, where each internal node has a test on a variable that determines which branch to descend to, and every leaf has an output value. Then when an unknown objects wants to be classified, it is simply put through the corresponding tests until a leaf node is reached, and its output with it. Figure 2.2 shows a simple decision Tree classifier.

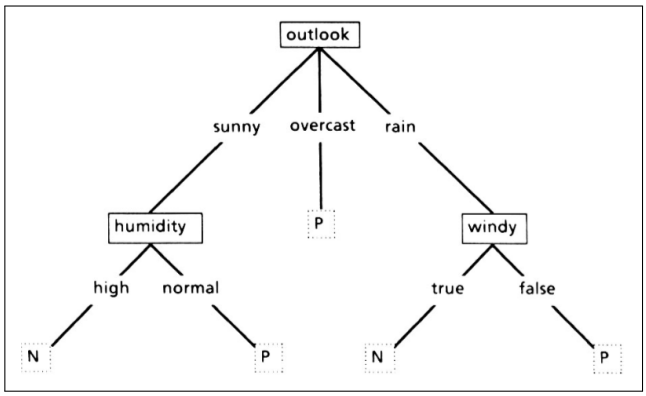

FIGURE 2.2. Decision Tree classifier. This models makes a series of test over the input variables, until finally an output value is reached. Figure taken from J. R. Quinlan (1986).

2.2.2. Model Selection

One of the most important phases of the classification process is the model selection. There are many theoretical foundations that support different classifiers in different scenarios, but in practice one must always evaluate the options before deciding. In order to test the different models, a group of instances whose classes are already known beforehand is needed. This is called a ”training set”. Then the models are trained on a subset of this group, also called ”training split”, and used to classify the remaining instances. Because

the real class of the remaining "test split" is already known, the precision of the different classifiers can be precisely evaluated.

There are many strategies to divide the training set into train and test splits. One of the most used is the K-Fold cross validation (Kohavi et al., 1995). This consists in dividing the training set into k groups. The model is then trained with $k - 1$ of those groups and tested on the remaining one, in a round robin fashion. This way all the training instances are used as a test variable at least once. Finally the model which proves to be the most precise is chosen.

2.3. Theoretical Foundations for our Method

As shown in 1.1, the value of time series features used for classification fluctuates importantly when the number of observations is small. And, normally, it tends to converge into more stable values as the lightcurves grow in length. This stabilization process varies for each object and for each of its features. If the value of a feature is not consistent, or varies greatly do to little observational differences, like the moment measurements began (in the case of stellar photometry), the exact moment the observations where realized, or small deviations on their values, then it is harder for a classifier to make accurate predictions. Therefore it is important to find a method able to assign measures of confidence to the estimated descriptors, and a way for classifiers to adjust their predictions accordingly.

For some simple statistical estimates (like the sample mean for example), closed form equations for the error of the estimate are available. This is not the case for the vast majority of features used in time series classification. The descriptors used in this context,

are normally very complex and exact theoretical values can not be obtained. To solve this type of cases, is that bootstrapping methods exist.

In the case of lightcurves, further complications arise. Normal bootstrapping approaches assume that realizations of the random variable are independent of each other, which is not the case of time series data. Lightcurves are measurements of the intensity with which an object shines at subsequent times. Therefor each point is clearly related to the ones that are close to it. In fact, the closer they are the more information they give from each other. It is because of this that only special time series bootstrap methods, are suitable for this task. Also, the fact that lightcurves are non uniformly sampled, not aligned, have uneven lengths and noisy observations puts even more restrictions to the techniques that might be used.

It is for these reasons that Gaussian Process Regression seems as the most reasonable approach. Gaussian Process is a very strong and flexible non parametric model that can be used for regression analysis. Because it is non parametric, it works based on a kernel function that defines the correlation between any two given observations. This kernel function can be chosen in many different ways in order to adjust to the particular characteristics where it wants to be applied.

The rest of the section shows a more detailed explanation of the topics previously mentioned. In particular Gaussian Process Regression, its application to time series bootstrapping and a simple explanation of the concept of bagging in machine learning.

2.3.1. Gaussian Process Regression

The regression problem corresponds to finding a function $f(\mathbf{x})$ that describes the relation between a vector of input variables $\mathbf{x}$ and a target variable y. In practice, however, the process by which data is obtained introduces noise to the values of y. In the following review a zero mean gaussian noise on y will be assumed. Therefor:

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma_n^2)$$

It is important to mention that modern astronomical instruments are normally able to estimate the measurement error ε associated to each observation. Although this is rarely the case in real applications, it does not affect the concepts presented.

One manner to try and solve the regression problem, and probably the most common one, is to restrict the class of functions for $f(\mathbf{x})$. Then the parameters that govern the model are optimized, so that it fits the observed data as best as possible. This is what it is called a parametric approach and, although they are usually easy to interpret, they lack expressive power in more complex scenarios.

Another approach, and the method we use in this work, is to define a probabilistic model on the functions f that might fit the data, and perform inference directly in the space of functions. This kind of techniques are known as "non parametric Bayesian models" because they establish a prior that reflects the type of functions we expect to see (periodic or soft curves for example), and then make bayesian inference by combining the data that we posses with the prior. This strategy is more flexible because it does not impose any

particular type of shape to the curves that might fit the data. Unfortunately, a function, may be evaluated in any number of locations, therefore it is unfeasible to track a probability distribution which describes its values, over a possibly infinitely large input vector **x**. But, when realizing regression, knowledge over the complete domain of **x** is unnecessary. In practice one is only interested in making predictions on a vector **x*** of limited size. This fact make Gaussian Processes able to solve the problem.

Whereas a probability distribution describes the possible outcomes of a random variable (discrete or continuous), a stochastic process governs the properties of functions. A Gaussian process, in particular, is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen & Williams, 2005). This means gaussian processes satisfy what is called a marginalization property, which states that if the gaussian process specifies $(y_1, y_2) \sim \mathcal{N}(\mu, \Sigma)$, then it must also specify $y_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})$. In other words, if it implies a distribution over a (possibly infinite) set of variables, then that same distribution applies for a smaller set of those variables. Therefore this property allows to make the same inference as if one was dealing with the infinite set of variables, when only working with the ones that are of interest.

A gaussian process is completely defined by its mean and covariance functions $m(x)$ and $k(x, x')$. On one hand, the mean function specifies the general tendency of the functions that will arise. To make an example, in many real applications the mean function is simply defined as $m(x) = 0$. Which means the average value of the functions perceived, at any given point x, is 0. On the other hand, the covariance function $k(x, x')$ defines the shape of the curves that appear, by determining the covariance between any two points.

More formally, the mean and covariance functions that govern a real process $f(\mathbf{x})$ are:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})],$$

$$k(x, x') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

and the Gaussian process

$$\mathbf{x} \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

Then in order to sample functions from a gaussian process prior, one must simply build a multinomial gaussian distribution, by replacing the x_* where one wants to sample in the mean function, and covariance function of choice, and with that build the corresponding $\mu(x_*)$, and $\Sigma(x_*)$. Assuming $m(x) = 0$, and a number of input points X_* then the function evaluated at those points f_* satisfies:

$$\mathbf{f}_* \sim \mathcal{N}\big(0, K(X_*, X_*)\big).$$

To further increase the understanding, lets assume a Gaussian process prior with a mean function $\mu(x) = 0$ and the following kernel function:

$$\mathrm{cov}(f(\mathbf{x}_p), f(\mathbf{x}_q)) = k(\mathbf{x}_p, \mathbf{x}_q) = \exp(-\frac{1}{2}|\mathbf{x}_p - \mathbf{x}_q|^2).$$

This function is called squared exponential and is one of the most common kernel functions. Figure 2.3 shows three samples taken at random from this prior.

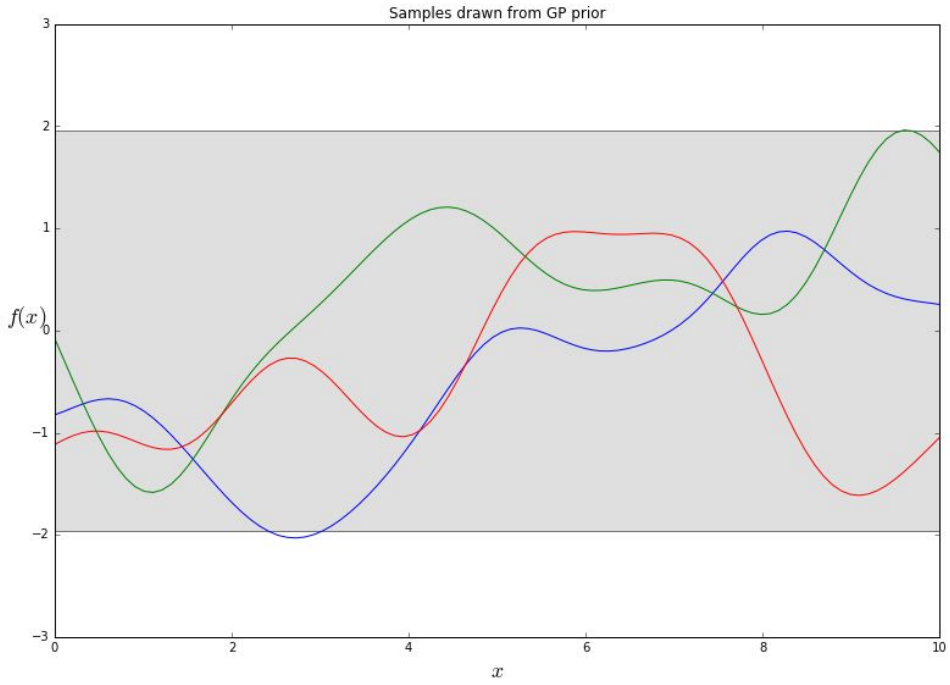

FIGURE 2.3. GP samples from a Squared Exponential prior. The squared exponential prior produces functions with a smooth behaviour.

Finally, having assumed a given GP prior one must be able to incorporate the information the training data provides from the phenomenon. In bayesian terms this corresponds to combine the likelihood of the functions, given the observed points, with the prior that has been chosen, in order to get the posterior distribution. The joint distribution of the training outputs $\mathbf{f}$ and the test outputs $\mathbf{f}_*$ according to the prior is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

To get the posterior distribution, the joint distribution must be conditioned to produce only those functions that are consistent with the observed data points. This becomes simply

$$\mathbf{f}_*|X_*, X, \mathbf{f} \sim \mathcal{N}\big(K(X_*, X)K(X, X)^{-1}\mathbf{f},$$
$$K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)\big).$$

Now, in real cases where observations are noisy, this equations can very easily be updated to incorporate this deviations. The covariance function, regardless of the one that is being used must be updated to

$$\mathrm{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}$$

$$or$$

$$\mathrm{cov}(\mathbf{y}) = K(X, X) + \sigma_n^2 I$$

where δ_{pq} is the kronecker delta. Then the joint distribution becomes:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

and one can finally arrive to the key predictive equations for Gaussian process regression.

$$\mathbf{f}_*|X, \mathbf{y}, X_* \sim \mathcal{N}\left(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)\right),$$

where

$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_*|X, \mathbf{y}, X_*] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}\mathbf{y},$$

$$\mathrm{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X_*).$$

For the complete derivation of this equations please refer to Rasmussen & Williams (2005).

In the case of regression problems the mean of the distribution formed by the posterior is taken as the function that best represents the relation between the input and the objective variable. One of the main advantages of this regression model, other than its flexibility, is that it does not only gives the values of the function evaluated on some locations X_*, but also, because it is probabilistic, the prediction has a deviation assigned to it. As figure 2.4 shows, this deviation reflects very accurately the knowledge data provides. As it tends to be smaller near the data points, and grows in the intervals where there are not any observations.

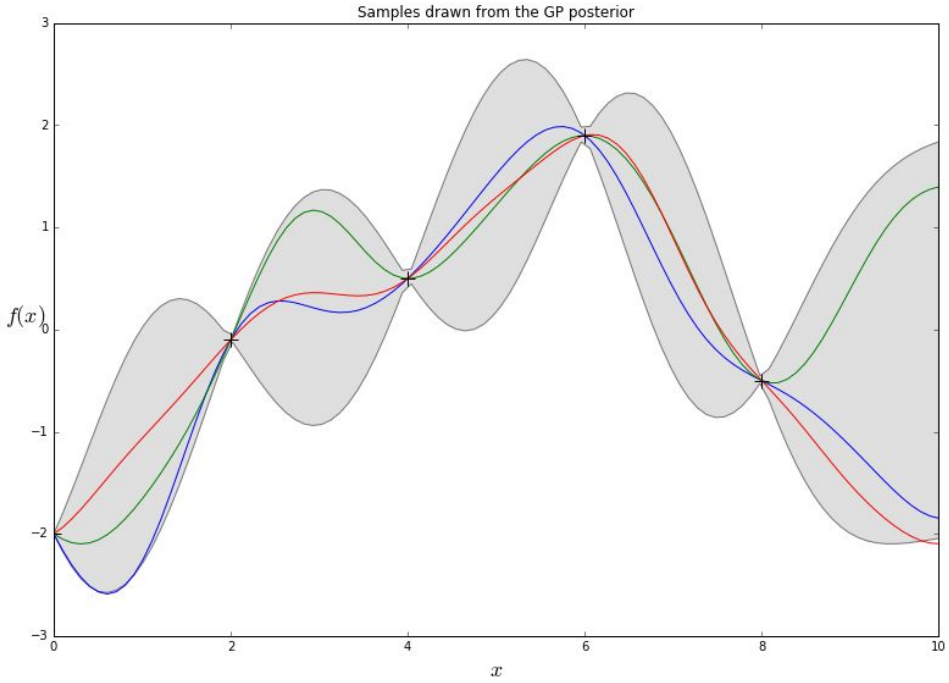

FIGURE 2.4. Gaussian Process posterior distribution. Three sampled functions from the GP posterior conditioned on five observations. The standard deviation is smaller close to the observations and gets bigger as one moves away.

2.3.2. Gaussian Process Bootstrap

Because the Gaussian process is a probabilistic model, it can be used in other ways rather than just a simple regressor. Kirk & Stumpf (2009), shows an example of how one can apply gaussian process regression to form a parametric time series bootstrap. The technique is pretty straight forward. A GP is adjusted over the time series of interest and the posterior distribution that best explains the behavior of the data is obtained. Finally several other possible time series can be sampled from the distribution, until a sample set of the desired size is formed.

This has many advantages over more traditional bootstrap approaches. First, it takes into account the relation different observations have on each other, and their relative position in the curve. In other words, if an observation is being sampled from an isolated fragment of a series, the value will vary considerably across different samples. While samples that have actual observations near them, will have similar values to the points around them. Second, the way observations influence each other can be controlled depending on the kernel function that is chosen. If periodic relations are expected or seen in the data, a periodic term can be added to the kernel for example. Third, depending on the kernel that is being used it allows to take into consideration the error in the values of the data that one posses. Fortunately, in the case of photometric lightcurves, catalogs posses the measurement error for every observation. Therefore this information can be added to the model in order to increase its accuracy, because the model knows beforehand which data points are more reliable than the rest. Finally, it uses all the observations available to create the sampled curves, whereas other bootstrap techniques work by dividing the data into subsets where valuable information may be lost.

2.3.3. Bagging

Bagging stands for bootstrap aggregating and is a machine learning ensemble strategy first introduced by Breiman (1996). It allows to combine the strength of multiple models in order to increase the overall predicting accuracy. The idea behind bagging is to generate many versions of the same predictor, where each version is trained on a different bootstrap sample of the original training set. Then, in the case of objects classification, a plurality

vote is performed by the group of models and the most voted class is regarded as the final output. Bagging not only improves the predictive power of the models, but also, by taking the voting distribution, it gives a a confidence measure of the prediction it makes.

Bagging is specially effective when the predictive method presents a high instability. Büchlmann & Yu (2002) formalize the notion of instability and shows how this technique helps to overcomes the effects it has in classification performance. The formal mathematical definition escapes the scope of this thesis, but the general idea is that instability is bigger when the model being adjusted does not converge to a definite value after a certain amount of data. In other words, if small changes in the data considered to train, or new observations of the same, produce differences in the final model. This is precisely the case shown in Figure 1.1. If the value of the features is highly unstable due to the small amount of observations, then the learned model will suffer the same problem, and the predictions it realizes will not be reliable.

3. RELATED WORK

3.1. Supervised Classification in Astronomy

Automatic classification of lightcurves success depends on two important and separated aspects of the process. First is the type of classifier being used. There are many different supervised classification algorithms in machine learning theory, each with its own advantages and limitations (Breiman, 2001; Cortes & Vapnik, 1995; Cox, 1958; J. R. Quinlan, 1986). But no matter which classifier is used, none of them will be successful if the features used for representation are not informative enough and therefore able to distinguish different kind of objects. This is one of the reasons why a lot of the research, regarding automatic classification of variable stars, has focused on the way lightcurves are represented rather than the classifiers they are fed to.

The second aspect is precisely that, how the objects are represented. Lightcurves, being composed of several hundreds of observations, which are hardly ever the same size, unevenly sampled and at different times, are not suited to be introduced to a classifier as input. To address this inconvenient, lightcurves are converted to vectors of numerical values. Great investigations efforts have been made to address this topic, Richards et al. (2011) introduced features that measure different statistical characteristics of time series like: standard deviation, skewness, kurtosis, slopes, and period. Kim et al. (2011) used features that capture the period, color, amplitude and the autocorrelation value of light curves in order to accurately identify quasars from the MACHO Large Magellanic Cloud database. Also, Pichara et al. (2012) proposed new features based on the parameters of

an adjusted continuous autoregressive model of the lightcurves, which generated an improvement in the accuracy of quasar detection methods. Nun et al. (2015) developed a software library which aims to facilitate the feature extraction process. The library includes a very complete compendium of the most important features in recent literature. And because it is open sourced is possible for the whole academic community to ensure that their implementation is correct and to contribute if new descriptors are designed in the future. Mackenzie et al. (2016) took a step in a different direction and proposed an unsupervised feature learning algorithm for variable stars classification. In other words, it devised an algorithm that automatically learns how to represent features without the help of any expert, or using any feature previously designed.

3.2. Bootstrapping

Bootstrap methods are a family of techniques in statistics, that rely on sampling with replacement in order to do inference (Efron & Tibshirani, 1994). They were first introduced by Efron (Efron, 1979), and have become increasingly popular since, because they allow to obtain measures of accuracy (such as the standard error) of a sampling statistic for small samples of data. Their only limitation is that they are computer intensive, because they require to repeat the calculation of the statistics of interest over many bootstrap samples, but the advance that computer power has shown recently makes this easy to overcome and implement.

Because of the previous reasons, and the fact that they can assign deviations measures to almost any statistics, they are perfectly suitable to obtain confidence intervals for the

values of complex astronomical features. Nevertheless, the case of lightcurve features is more complicated than usual. Because this descriptors work on time series, which are order dependent data, the manner in which to resample the data is not evident.

Bootstrapping time series, or order dependent data, is not an obvious task, and many different approaches have been proposed along the years. Special considerations must be made, because the data cannot necessarily be changed of order without changing the values of the estimators one wants to calculate. The Block Bootstrap (Kunsch, 1989), attempts to solve this issue by dividing the time series observation in adjacent blocks of length ℓ. Then the resampling is made by drawing this blocks uniformly, thus preserving the original time series structure within each block. Although the choice of ℓ is not obvious, Block Bootstrap has been shown to work for general stationary data generating processes (Bühlmann, 2002). Kreiss & Franke (1992) introduces a different kind of approach based on autoregressive models and sieve approximation (Grenander, 1981). Finally, Kreiss et al. (1998) proposes the so called local bootstrap, which aims to model the dependency that each of observation has on the previous ones. This models proves to be effective only when the observations are generated by a short-range dependent process (Paparoditis & Politis, 2000).

Although all of these methods prove to be effective in specific cases, they all make different assumptions over the time series where they are going to be applied, in order to deliver good results (Bühlmann, 2002). This, together with the fact that photometric lightcurves do not obey many consistency requirements, make necessary to look for more flexible ways of obtaining bootstrap samples.

4. METHODOLOGY

As demonstrated before in Chapter 1, when lightcurves are composed of only a few points the value of the features that describes them becomes disperse. This because, as there are little observations the values of each one becomes more important, and tiny variations on their values, or the presence of new ones, affects the estimation considerably. This deteriorates the effectiveness of classifiers as features are not longer able to describe different objects consistently. To overcome this problem we draw from what is proposed in Kirk & Stumpf (2009), to create bootstrapped samples of any feature, together with a bagging approach to combine the different outcomes each set of samples produces. By doing this we diminish the effects feature variance has on classification performance.

Our algorithm consists of four major steps. In the first stage, we adjust a gaussian process regression model to each lightcurve, and sample n time series from the posterior distribution obtained. In the second stage we take a different sample from each of the original objects to form n different sets. Then we calculate a set of descriptors for each of the samples in this so called "sample sets". The third stage consists of training a classifier on each of this sets, thus obtaining n different models. The fourth and final step is to classify the unknown lightcurves. For this we use the same idea, n samples are taken from the adjusted GP on the lightcurve. Finally each of this samples is classified by one of the models, thus obtaining a voting distribution on the object's class. Figures 4.1 to 4.6 show the different stages of the process.

4.1. Time series bootstrapping

The first step of the process is to take every lightcurve in the training set, and take bootstrapped samples from each of them. The idea is that each of this lightcurve presents different behaviors in the sections where the sampling is poor, in other words where not many measurements where made. On the contrary if the lightcurve presents a very good sampling we expect the bootstrapped samples to be very similar.

To obtain bootstrap samples of the lightcurves in the training set we adjust a gaussian process regression model on each of them, and take n samples from the obtained posterior distribution. As described in Section 2.3.1, what defines the shape of a Gaussian Process is the kernel function. In the case of photometric lightcurves, we take from the work done by Faraway et al. (2014), and use a similar gaussian process prior to the one they proposed. Because variable stars do not normally show a noticeable variation on the overall amount of flux, we choose to use a constant mean function equal to the mean value of their signal. Then the prior we use is:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$

where

$$\mu(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

and

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}_p - \mathbf{x}_q)^2\right) + \sigma_n^2 \delta_{pq}$$

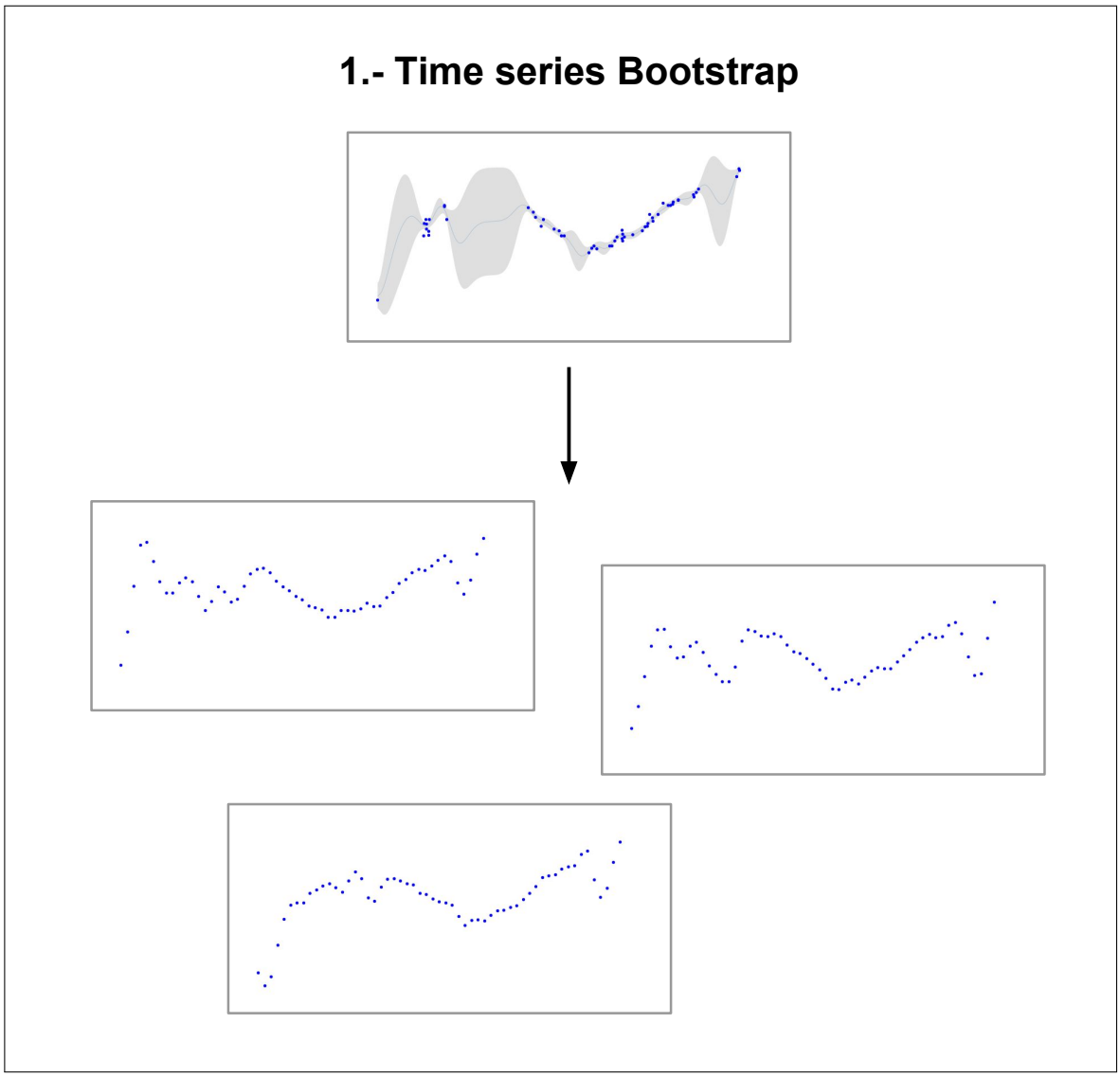

FIGURE 4.1. Illustration of the first stage of the algorithm, the Time series Bootstrapping.

In the equations above, $\mu(\mathbf{x})$ is the mean of the signal, σ_f^2 is the signal variance, l is the length scale, δ_{pq} is the kronecker delta and σ_n^2 is the noise variance. The last term is particularly interesting because for astronomical data, unlike the majority of cases, random error can be measured for each observation.

Figure 4.2, shows the adjusted gaussian process model over a lightcurve from the MACHO catalog.

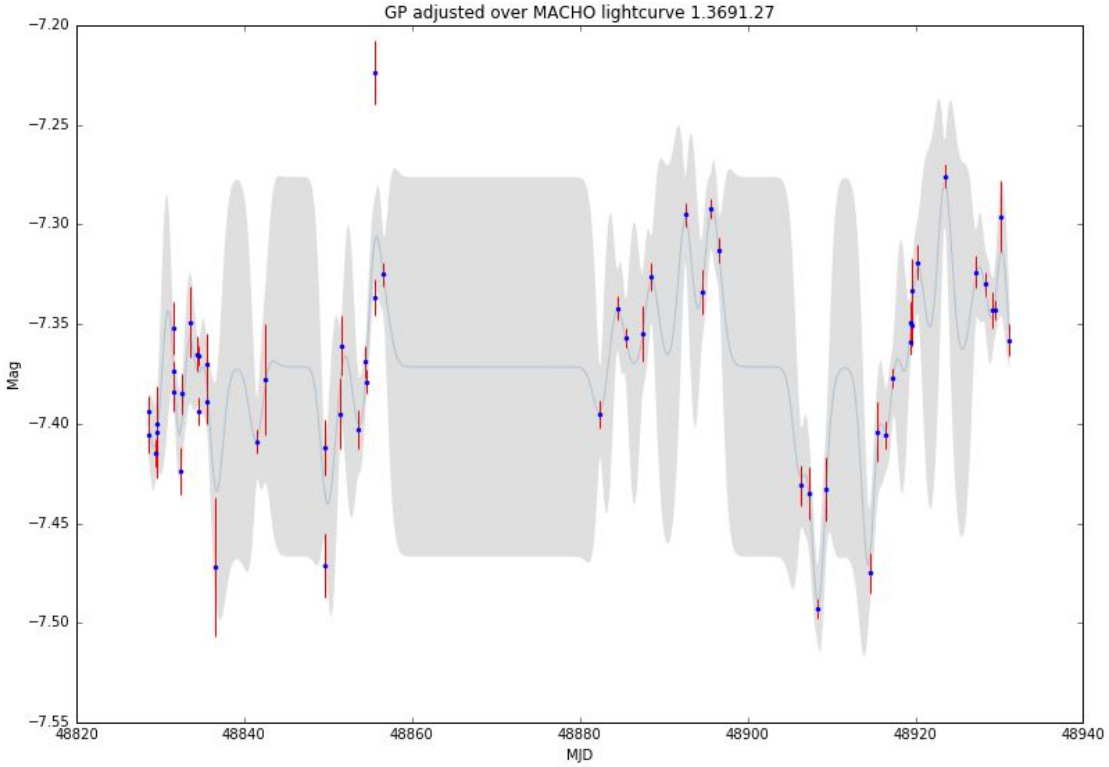

FIGURE 4.2. MACHO lightcurve Gaussian Process fit. The model captures the general form of the time series, and adjusts the deviation according to the observations possessed. The model is less influenced by measurements with greater measurement error.

The number of samples to take is not obvious at first hand and, because it may change in different scenarios, it must be found empirically. There is a trade-off between the accurate representation of the curves distribution and the computational time the method takes. In our experiments we found that 100 samples gave optimal results while still being computationally feasible. Figure 4.3 shows the original light curve on top and two random samples taken from it.

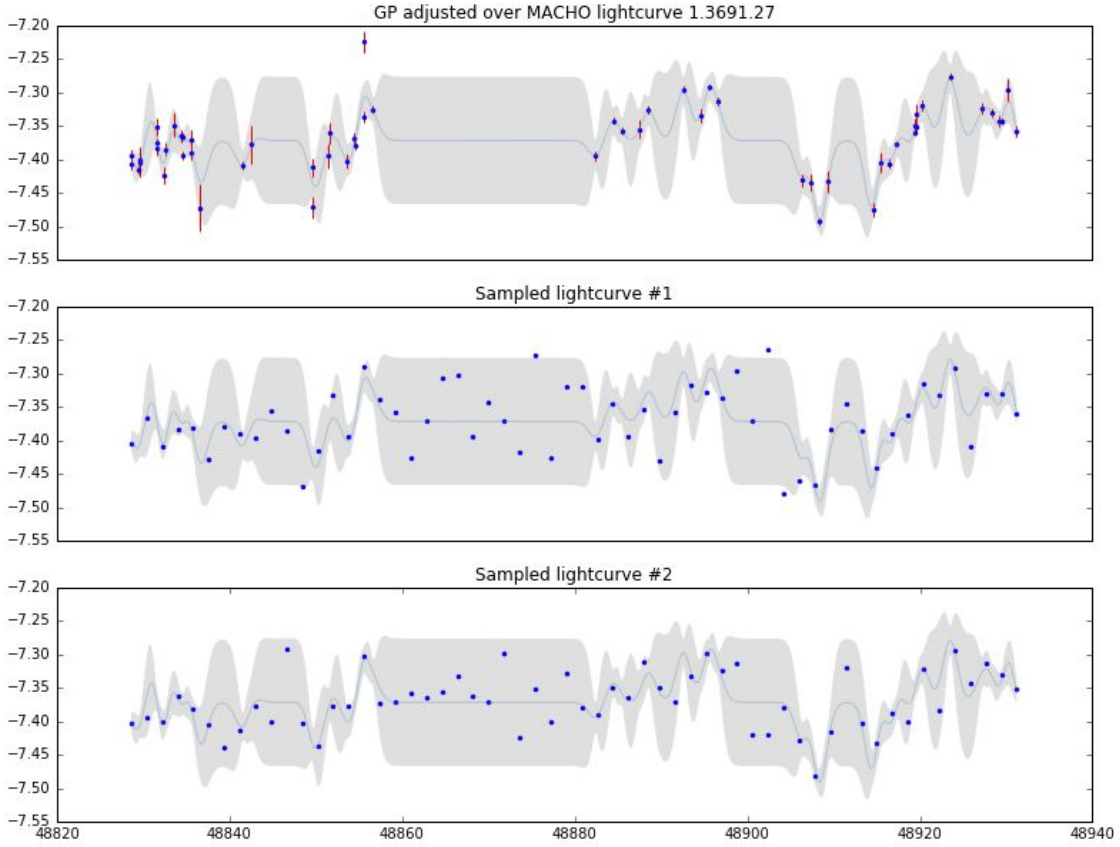

FIGURE 4.3. GP random samples. The GP is adjusted over a lightcurve and two random samples. The samples are taken at uniform times over the span of the measurements. Sampled observations near the original ones have very similar values, while samples taken from empty spaces are more disperse.

4.2. Sample sets

After taking the bootstrap samples, we form n different training sets. Where each set contains a single and different sample for each of the original labeled lightcurves. We refer to these sets as the sampled sets. An illustration of this stage is shown in 4.4.

Then a group of time series features is calculated for each curve of the sampled sets. For this task we use FATS Nun et al. (2015). This open sourced python library allows easy and efficient calculation of the most used lightcurve features existent in literature.

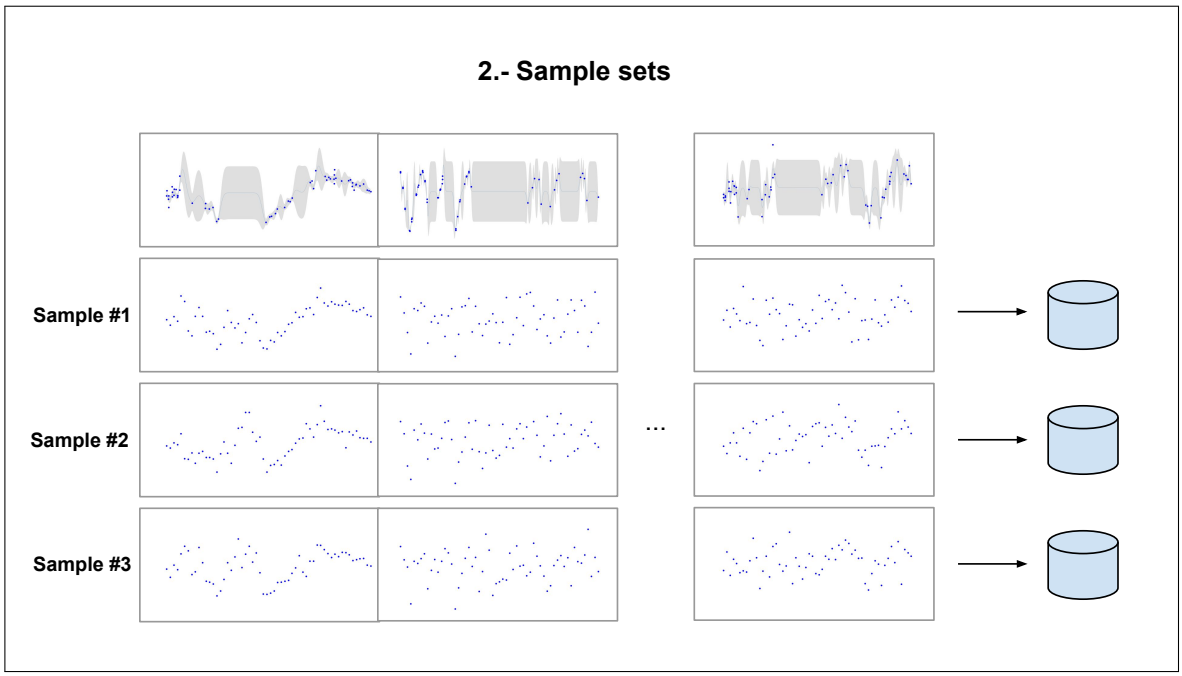

FIGURE 4.4. Illustration of the second stage of the algorithm. The different samples of each lightcurve are separated into different "sample sets". Each of this sets represents a different random scenario of the observed lightcurves.

Although this tool allows to calculate more than 50 different time series features, we restricted our work to a subset of only twenty three features that prove to be effective for classification. We decided to discard all features that need different bands to be calculated, because this adds further complexity to the problem, and including them goes beyond the scope of this investigation.

At this stage, because the features have been calculated for n bootstrapped samples of each lightcurve, we now posses an estimation of the distribution of their values for each object. According to this distributions, features that present a high variability in their values will be less influential on the classification, whereas features that are more consistent

will be taken more into account by the model. Analysis on the features distribution is shown in 5.2.5.

4.3. Training

After we calculate the features for each of the samples sets, we adjust a decision tree classifier (Breiman et al., 1984; J. R. Quinlan, 1986) on each of them. We decided to use decision trees as the classifiers to combine, because this way our model resembles the Random Forest Classifier (Breiman, 2001), which has proven to be one of the most effective classifiers for variable star classification (Carliles et al., 2010; Richards et al., 2011; Pichara et al., 2012; Pichara & Protopapas, 2013). Although instead of combining trees, trained with different subsets of features, we combine trees trained on different random scenarios, where each scenario is a possible uncertain outcome of the values of the original training set. An illustration of this stage is shown in 4.5.

4.4. Classification

The final stage is to predict the class of a new unlabeled object. For this the same logic presented before is used. Because the values of a new lightcurve may be corrupted, the prediction yielded by the classifiers have a greater chance of being incorrect. Therefor, again, n different samples are obtained and their features calculated. Then each of this samples is given to a different trained model for it to cast its vote. Finally the vote of all models is combined and the most popular class is regarded as the final predicted class. An illustration of this stage is shown in 4.6.

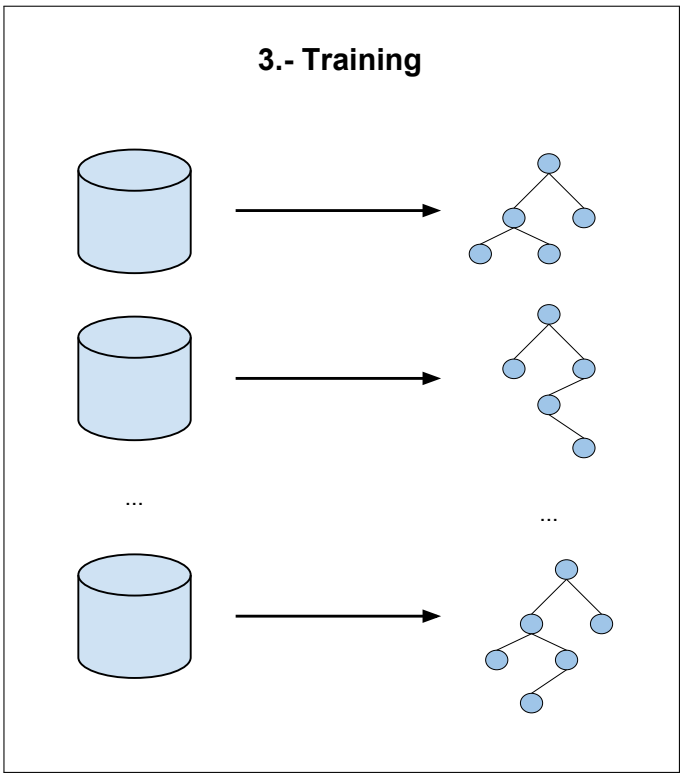

FIGURE 4.5. Illustration of the third stage of the algorithm. Each sampled set of lightcurves, and the subsequent features, es used to train a different decision tree. This way each decision tree will present a different structure depending on the particular set of samples which it was trained with.

It is important to note that because a voting is taken place, the actual prediction of this framework gives a belief of belonging to each of the possible classes. One can take advantage of this quality to discard, or further analyze confusing results, in the case, for example, that many models give different predictions. If a lightcurve presents very little, noisy, or unevenly distributed measurements, the value of its features will change greatly among different samples. Therefor it is likely for different classifiers to be confused and cast contradicting votes. On the contrary, if a lightcurve is well sampled, and therefor very well described, the voting of the different classifiers is likely to be more consistent.

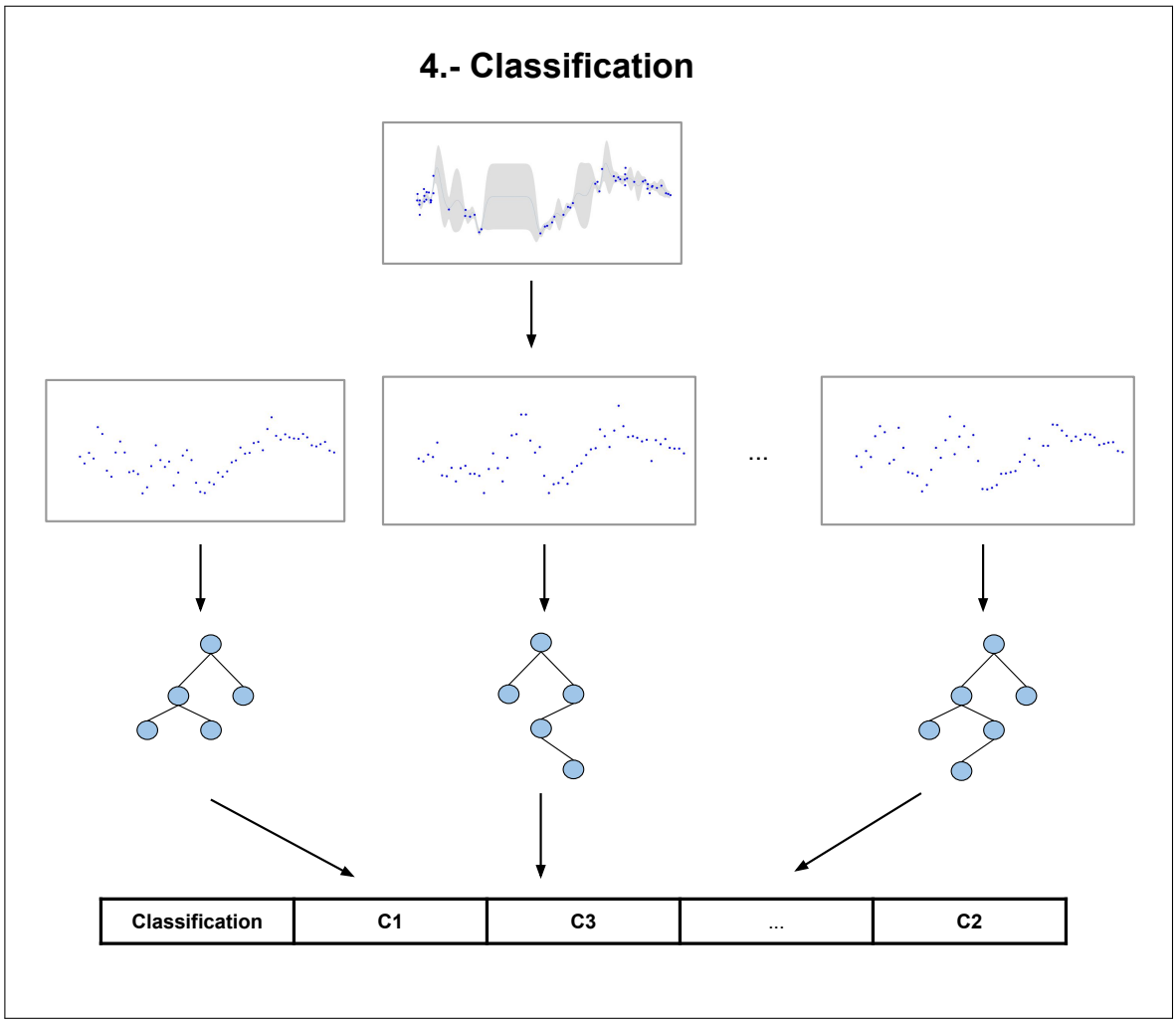

FIGURE 4.6. Illustration of the final stage of the algorithm. When an unknown lightcurve needs to be classified, the same process is realized. Various samples are taken from it, their features calculated, and then given to a different classifier from the ones trained on the previous step.

5. EXPERIMENTAL RESULTS

In this section the experimental results are presented. First, we detail a synthetic experiment based on the Robot navigation dataset. The goal of this example is to show how classification results are affected when the value of the variables are affected by randomness. And then, how this problem can be reduced by using a bagging technique like the one proposed. Then we present the classification results obtained by working with photometric data. In this case we take complete photometric catalogs and, by reducing the number of observations by hand, we simulate the scenario where these surveys were beginning their observation process.

The difference of the real case with the synthetic one (and one of the key contributions of this investigation) is how the method proposed in Section 2.3.2 is used to obtain the bootstrapped samples of noisy lightcurve features.

In both the synthetic and real cases we compare how a bagging scheme classifier improves the classification of standard models. Classifier performance is measured with a 10-fold stratified cross-validation F-Score on each of the classes present in the training sets. We choose the classic Decision Tree (Breiman et al., 1984; J. R. Quinlan, 1986; R. Quinlan, 1993) and the Random Forest (Breiman, 2001) as the classifiers which to compare our model with. We compare with the Decision Tree to validate that the bagging realized in our method improves the results of this simple model. Second, we compare with the Random Forest because this is the classifier of choice in many recent literature (Nun et al., 2014; Kim et al., 2011; Pichara et al., 2012; Pichara & Protopapas, 2013;

Richards et al., 2011) regarding automatic classification of variable stars, and is also the most precise according to our tests. All three models work with the exact same set of features.

5.1. Robot Experiment

5.1.1. Data

The dataset used for this experiment is taken from the UCI machine learning repository(Lichman, 2013). It is called ”wall following robot” dataset (Freire et al., 2009) as it was collected from a mobile robot which tries to navigate along the walls of a room without colliding. The robot was equipped with a belt of 24 ultrasound sensors that measure the proximity of objects in a 360 degree radius at evenly timed steps. Then each entry of the dataset contains the readings of the 24 sensors together with a class, which corresponds to the specific movement the robot must make, from a group of four defined possible movements.

5.1.2. Training Set Composition

The robot training set is composed of 5456 readings, and the class composition is detailed in Table 5.1. We choose to work on this dataset, as a synthetic example, because it does not have any missing values and it also has a similar number of attributes and instances as the photometric datasets we work with.

TABLE 5.1. Robot Training Set Composition

	Class	Number of Objects
1	Move-Forward	2205
2	Sharp-Right-Turn	2097
3	Slight-Right-Turn	826
4	Slight-Left-Turn	328

5.1.3. Results

To evaluate the effects that feature noise has in classification results, the following experiment is made. The robot dataset is taken, and for each feature the amplitude is calculated. That is, the difference between the maximum and the minimum value it takes on the dataset. Then to each feature, of each instance, a white noise kernel is added, with standard deviation equal to a randomly chosen value between zero and a fixed percentage of the amplitude. So, for example, to generate a dataset with a 5 percent of noise, we take a sample from all of those kernels, using 5 percent of the corresponding feature amplitude as the maximum possible standard deviation.

The advantage of doing this, is that it allows us to generate any number of randomly sampled sets, from the same feature distribution. In this way, we can compare, how a classifier that works on a single observed dataset, works against an ensemble of classifiers, each trained on a different random dataset.

We did this test, for various levels of added noise, ranging from 5 to 20 percent. The results obtained are shown in Figure 5.1. It is evident that for all of the classes, the voting scheme classifier gives better results than both, the Decision Tree and the Random Forest, trained over a single observed random set.

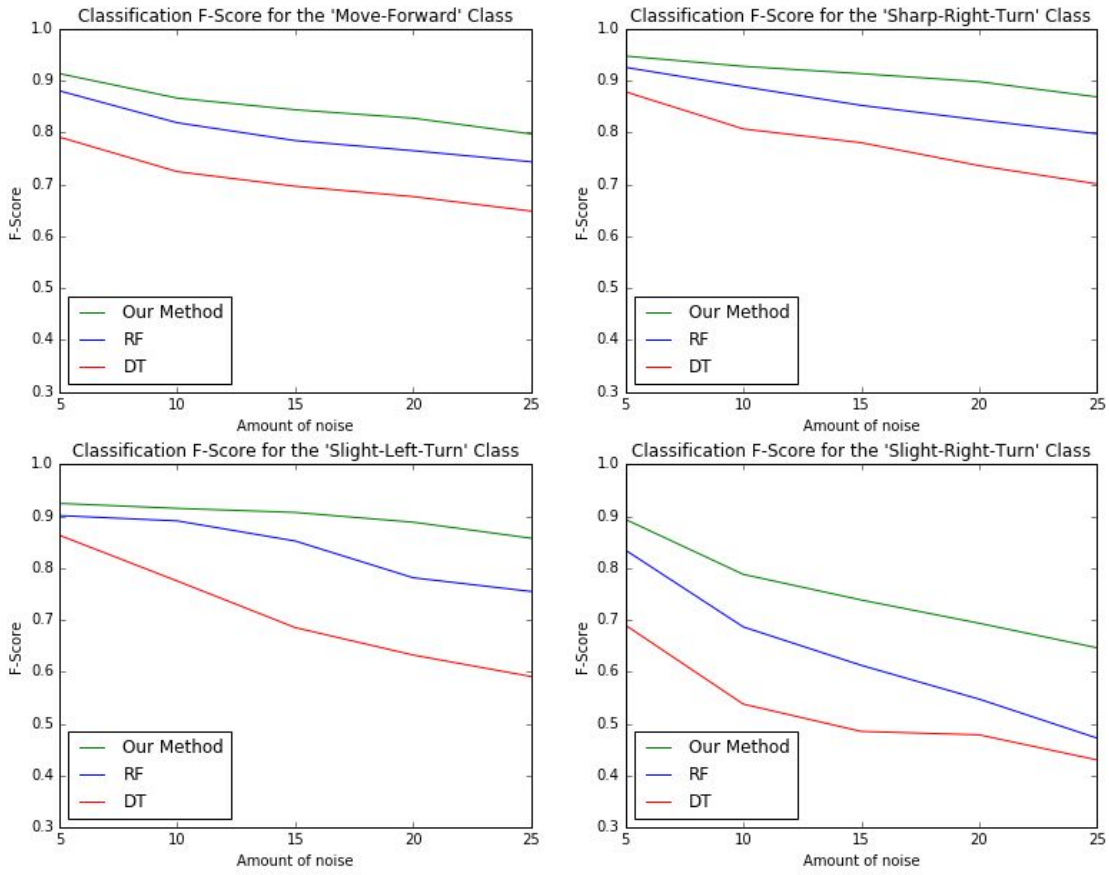

FIGURE 5.1. Classification F-Score for the Robot training set. The results obtained by bagging the predictions of many different classifiers are less affected by noise than both the Decision Tree and the Random Forest.

5.2. Lightcurve Classification

In this section the results obtained working with data obtained from two different astronomical surveys (MACHO and OGLE) are described. First, a brief depiction of the catalogs used in this section is shown, followed by a description of the training sets which we worked with. Then, the results corresponding to the first stage of the classification framework, the time series bootstrapping, are shown. After that, we comment on the values

obtained for the empirical distribution of the time series features. Finally the classification results obtained by working with the proposed framework are shown.

5.2.1. MACHO

The MACHO catalog is the result of a project that aimed to find dark matter in the form of massive compact halo objects (MACHOs). The project made photometric observations of tens of millions of stars, for almost 6 years, in the Large Magellanic Cloud (LMC), Small Magellanic Cloud (SMC) and Galactic bulge (Alcock et al., 2001).

5.2.2. OGLE-III

The OGLE-III catalog of variable stars (Udalski et al., 2008) contains photometric data obtained during the third phase of The Optical Gravitational Lensing Experiment. This wide-field sky survey was designed with the objective of finding dark matter through the microlensing technique. It contains regular measurements of the brightness of more than 200 million objects, from the large and small Magellanic Clouds, the Galactic bulge and the Galactic Disk, taken since 2001.

5.2.3. Training sets

The photometric training sets are labeled subsets of the actual surveys. The MACHO training set is composed of 6627 curves (Kim et al., 2011). The OGLE training set is composed of 4733 labeled curves. Their class composition is detailed in tables 5.2 and 5.3. The OGLE training set is chosen as a subset of the most represented variable star classes in the catalog, and of comparable size to the MACHO dataset. Also, in order

to make the classification more difficult, we choose objects from the Large Magellanic Cloud, Small Magellanic Cloud, and the Galactic Disk.

TABLE 5.2. MACHO Training Set Composition

	Class	Number of Objects
1	Non Variable	4768
2	Quasar	34
3	Be Star	112
4	Cepheid	101
5	RR Lyrae	606
6	Eclipsing Binary	255
7	MicroLensing	393
8	Long Period Variable	358

TABLE 5.3. OGLE-III Training Set Composition.

	Class	Number of Objects
1	Cepheid	724
2	Type 2 Cepheid	575
3	RR Lyrae	998
4	Eclipsing Binary	794
5	Delta Scuti	656
6	Long Period Variable	986

5.2.4. Bootstrapping results

Figure 5.2 shows a Gaussian process model adjusted over a lightcurve from the MACHO catalog. It is important to notice that the model assigns greater uncertainty to regions where no observations are recorded, while regions with better measurements are regarded as more accurate. This is very important, because lightcurves with greater gaps in their measurements will produce bootstrapped samples with greater differences in their values, while better sampled curves will result in more consistent ones.

Figure 5.3 shows the same lightcurve fit and three samples taken randomly from the model. It is evident that all the samples present very similar values on regions with higher density of observations. On the other hand, regions where the original time series has less information, are very different among the samples. This is the behavior expected for this stage of the process.

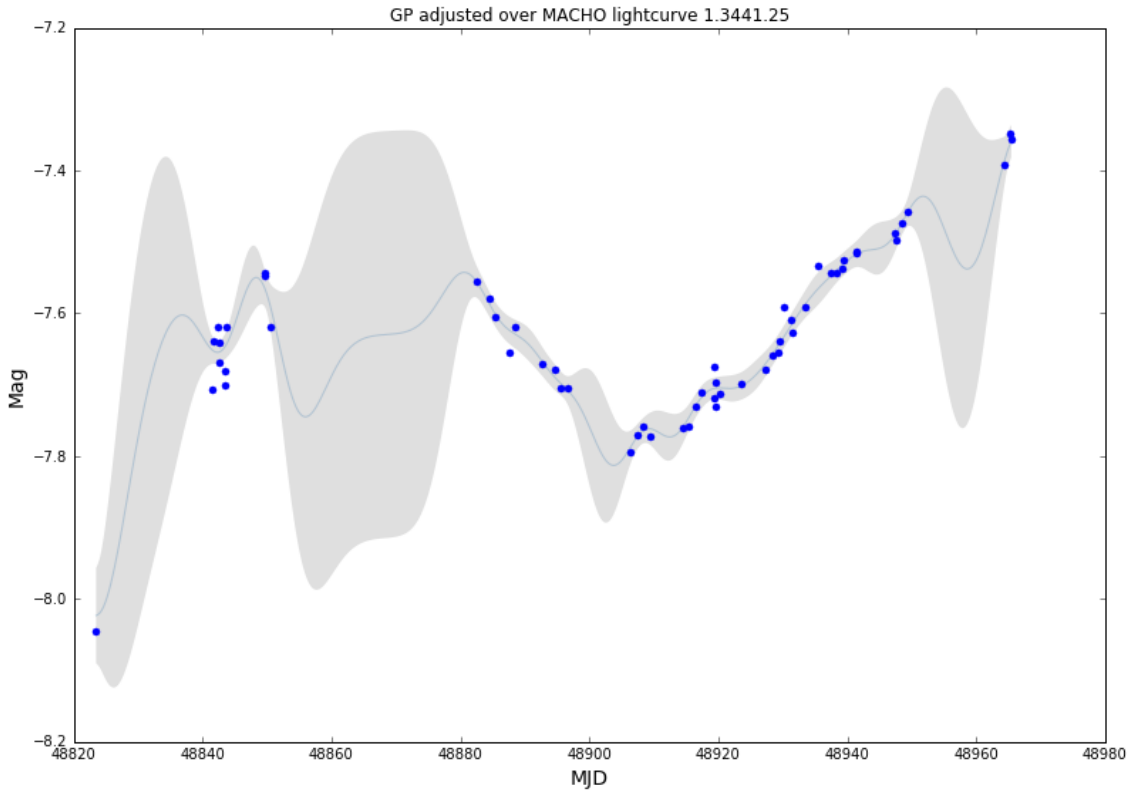

FIGURE 5.2. Gaussian Process regressor adjusted over a lightcurve from the MACHO catalog. The model gives greater uncertainty to regions where no observations are recorded.

5.2.5. Feature distribution

Every statistical estimate has an inevitable degree of error in its estimation. Therefore finding methods to assign measures of accuracy in their values is crucial. Variables

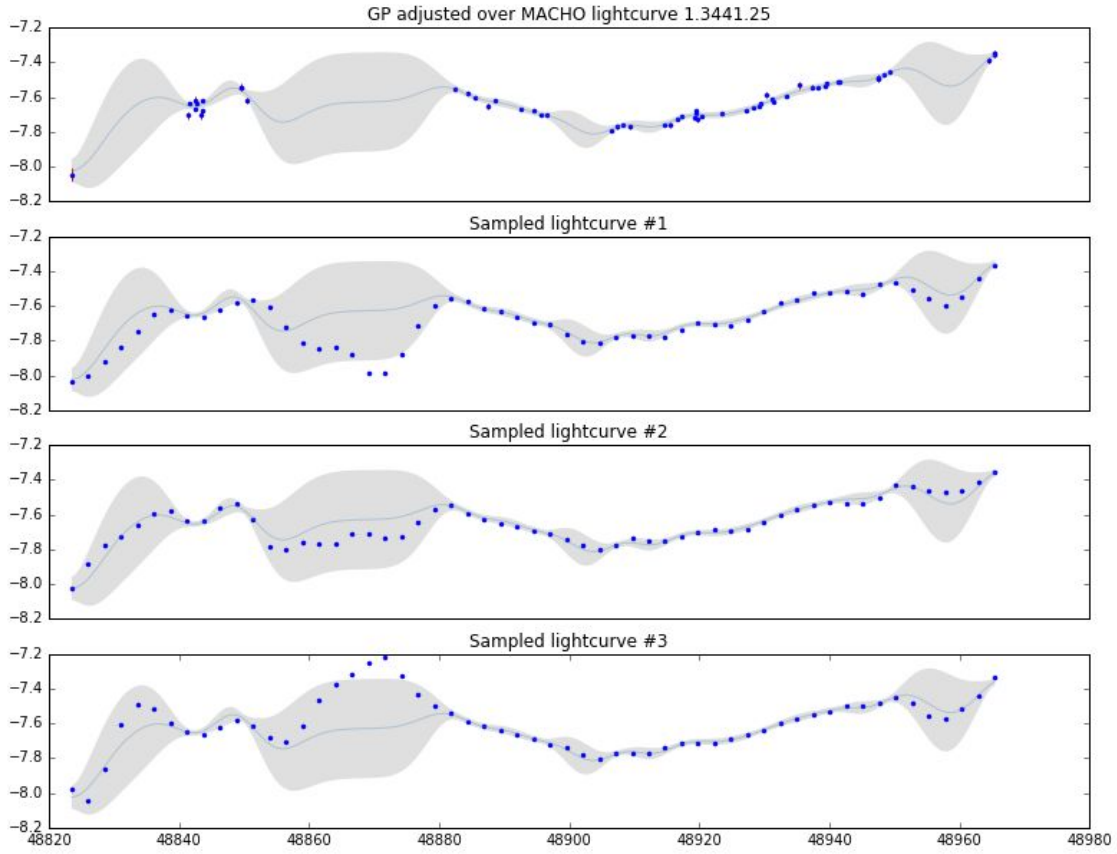

FIGURE 5.3. Gaussian process adjusted over a light curve and three random samples taken from it. Samples taken from empty spaces are more disperse. Therefore lightcurve with poorer sampling, both in total number and uniformity of observations, will present a greater dispersion in the value of their calculated features.

which values present high degrees of error (just as some photometrical measurements) are normally dismissed versus more precise ones when using them for analyses. The bootstrapping technique used in this investigation allows for the same logic to be applied to the time series features used for classification. Figure 5.4 shows a graphical comparison of the distribution of the same feature for two different curves from the OGLE catalog. It is evident that one curve presents much more error in the estimation of the eta variability feature.

The curve that presents more consistency in its values, will be more influential in the classification process than the other one. Because as the values will be given to different classifiers, inconsistent behaviors are dismissed by the voting of the majority, while consistent ones are reinforced.

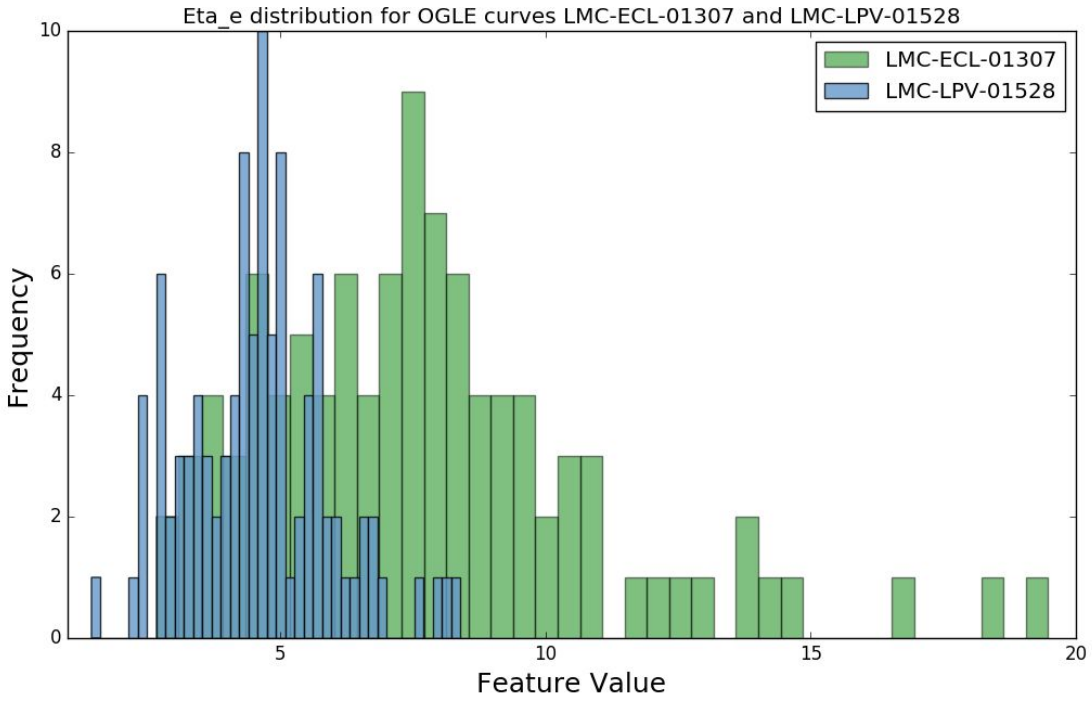

FIGURE 5.4. Distribution for the values of the eta variability measure for two lightcurves from the OGLE catalog. It is evident that the the blue values are much more concentrated and thus present lower variability. On the other hand the green values show many escaped higher values.

5.2.6. Classification Results

Tables 5.4 and 5.5 show the classification results, for each catalog, obtained by the model proposed in this paper, a Random Forest and a Decision Tree. Compared with the Decision Tree, the UF shows better results for all of the classes on the MACHO training set, except for the Cepheids. On the OGLE training set the results are almost the same.

	Class	Random Forest	Our Method	Decision Tree
1	Be Star	0.570	0.546	0.461
2	Cepheid	0.931	0.790	0.870
3	Eclipsing Binary	0.474	0.465	0.392
4	Long Period Variable	0.877	0.856	0.850
5	RR lyrae	0.737	0.762	0.671
6	Microlensing	0.823	0.775	0.690
6	Non Variable	0.930	0.936	0.910
6	Quasar	0.041	0.247	0.130

TABLE 5.4. Classification F-Score on the MACHO training set

	Class	Random Forest	Our Method	Decision Tree
1	Cepheid	0.804	0.833	0.757
2	Delta scuti	0.824	0.825	0.807
3	Eclipsing Binary	0.872	0.728	0.845
4	Long Period Variable	0.974	0.963	0.954
5	RR lyrae	0.832	0.891	0.704
6	Type II Ceph	0.775	0.785	0.694

TABLE 5.5. Classification F-Score on the OGLE training set

Except for the Eclipsing Binaries, all classes see their F-Score improved by our model. These results show that combining the votes of many decision trees, over different samples of the same objects, effectively improves the classification performance.

Compared with the Random Forest, although there are specific differences on the per class performance, both models have similar results on the MACHO training set. The UF gets better results for RR Lyrae and Quasars, while the RF does better at identifying Cepheids and Microlensings. On the OGLE training set, the results are similar, with the difference that the UF gives better results for Cepheids and RRLyrae, which are extremely valuable to find, compared with the rest of the classes.

Although the proposed model does not outperforms the random forest classifier is important to notice the high precision the model presents for some important variability

classes. For example, RR Lyrae stars have an 0.89 f-score which is impressive, because the model is only working with five percent of the available observations. Long period variables are even more impressive with a 0.97 f-score. This results show that astronomers may not need to wait long periods of time to identify this type of objects reliably.

6. CONCLUSIONS

In this work, we present a new way of bootstrapping features for lightcurve classification where, instead of making subsamples of the instances of the training set, we sample the original time series used to estimate them.

A Gaussian Process Regression is used to form a probabilistic model of the values observed for each lightcurve. In bayesian terms, this is called a posterior distribution, because it combines the evidence the data gives, with a prior that reflects the beliefs we have on the behavior of stellar variability. The prior also considers the measurement error each observation presents and adjusts the model accordingly.

Our results show that the regression model correctly describes the behavior of the lightcurves. Because the Gaussian Process is a generative model, it uses all of the observations to form new samples, instead of only considering the information of preceding points. This preserves the long term patterns underlying in the data. The model also assigns greater deviation to the regions where no observations are recorded. Therefor samples taken from empty spaces are more disperse than the ones taken near other observed points.

We have also shown how to obtain an empirical distribution of the value of any feature. Lightcurves with poorer sampling, both in total number or uniformity of observations, present a greater dispersion in the value of their calculated features. This allow for a model to discard the instances which values have higher variability for others with more consistency in their values.

Finally, by comparing our results to the ones of a single decision tree, we show that combining the votes of many different classifiers, over different samples of the same objects, increases the overall classification accuracy. Although it does not outperforms the random forest classifier, both models show that they are are able to recognize some classes with surprising precision, in spite of working with only a fraction of the available information.

We believe this framework constitutes the first attempt to include the error of time series features into the automatic classification process. In this sense, it proves that better results can be obtained by using simple models, like the decision tree, when this issue is taken into account. We hope that this investigation encourages the research community to take more into consideration the error associated with feature calculation, how the measurement process impacts it, and to develop more ways to overcome it.

Acknowledgments

This work is supported by Vicerrectoría de Investigación (VRI) from Pontificia Universidad Católica de Chile, the Institute of Applied Computer Science at Harvard University, and we also acknowledge the support from CONICYT-Chile, through the FONDECYT project number 11140643.

References

Alcock, C., Allsman, R., Alves, D. R., Axelrod, T., Becker, A. C., Bennett, D., et al. (2001). The macho project: microlensing detection efficiency. *The Astrophysical Journal Supplement Series*, *136*(2), 439.

Aubourg, E., Bareyre, P., Brehin, S., Gros, M., Lachieze-Rey, M., Laurent, B., et al.(1993). Evidence for gravitational microlensing by dark objects in the galactic halo. *Nature*, *365*, 623–625.

Bloom, J., & Richards, J.(2011). Data mining and machine-learning in time-domain discovery & classification. *Advances in Machine Learning and Data Mining for Astronomy*.

Bloom, J., Richards, J., Nugent, P., Quimby, R., Kasliwal, M., Starr, D., et al. (2012). Automating discovery and classification of transients and variable stars in the synoptic survey era. *Publications of the Astronomical Society of the Pacific*, *124*(921), 1175.

Breiman, L.(1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L.(2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A.(1984). *Classification and regression trees*. CRC press.

Büchlmann, P., & Yu, B.(2002). Analyzing bagging. *Annals of Statistics*, 927–961.

Bühlmann, P.(2002). Bootstraps for time series. *Statistical Science*, 52–72.

Carliles, S., Budavári, T., Heinis, S., Priebe, C., & Szalay, A. S.(2010). Random forests for photometric redshifts. *The Astrophysical Journal*, *712*(1), 511.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 215–242.

Debosscher, J., Sarro, L., Aerts, C., Cuypers, J., Vandenbussche, B., Garrido, R., et al. (2007). Automated supervised classification of variable stars. I. Methodology. *Astronomy and Astrophysics*, *475*, 1159–1183.

Drake, A., Djorgovski, S., Mahabal, A., Beshore, E., Larson, S., Graham, M., et al. (2009). First results from the catalina real-time transient survey. *The Astrophysical Journal*, *696*(1), 870.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The annals of Statistics*, 1–26.

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

Faraway, J., Mahabal, A., Sun, J., Wang, X., Zhang, L., et al. (2014). Modeling light curves for improved classification. *arXiv preprint arXiv:1401.3211*.

Freire, A. L., Barreto, G. A., Veloso, M., & Varela, A. T. (2009). Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *Robotics symposium (lars), 2009 6th latin american* (pp. 1–6).

Grenander, U. (1981). *Abstract inference*. Wiley New York.

Huijse, P., Estevez, P. A., Protopapas, P., Principe, J. C., & Zegers, P. (2014). Computational intelligence challenges and applications on large-scale astronomical time series databases. *Computational Intelligence Magazine, IEEE*, *9*(3), 27–39.

Jiawei, H., & Kamber, M. (2001). Data mining: concepts and techniques. *San Francisco, CA, itd: Morgan Kaufmann*, *5*.

Kim, D.-W., Protopapas, P., Alcock, C., Byun, Y.-I., & Bianco, F. B. (2009). Detrending time series for astronomical variability surveys. *Monthly Notices of the Royal Astronomical Society*, *397*(1), 558–568.

Kim, D.-W., Protopapas, P., Bailer-Jones, C. A., Byun, Y.-I., Chang, S.-W., Marquette, J.-B., et al. (2014). The epoch project-i. periodic variable stars in the eros-2 lmc database. *Astronomy & Astrophysics*, *566*, A43.

Kim, D.-W., Protopapas, P., Byun, Y.-I., Alcock, C., Khardon, R., & Trichas, M. (2011). Qso selection algorithm using time variability and machine learning: Selection of 1,620 qso candidates from macho lmc database. *arXiv preprint arXiv:1101.3316*.

Kim, D.-W., Protopapas, P., Byun, Y.-I., Alcock, C., Khardon, R., & Trichas, M. (2011). Quasi-stellar object selection algorithm using time variability and machine learning: Selection of 1620 quasi-stellar object candidates from macho large magellanic cloud database. *The Astrophysical Journal*, *735*(2), 68.

Kirk, P. D., & Stumpf, M. P. (2009). Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data. *Bioinformatics*, *25*(10), 1300–1306.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).

Kreiss, J.-P., & Franke, J. (1992). Bootstrapping stationary autoregressive moving-average models. *Journal of Time Series Analysis*, *13*(4), 297–317.

Kreiss, J.-P., Neumann, M. H., & Yao, Q. (1998). Bootstrap tests for simple structures in nonparametric time series regression.

Kunsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 1217–1241.

Lichman, M. (2013). *Uci machine learning repository.*

Mackenzie, C., Pichara, K., & Protopapas, P. (2016). Clustering-based feature learning on variable stars. *The Astrophysical Journal*, *820*(2), 138.

Matter, D. (2007). The LSST Science Requirements Document. *Science*, 1–41.

Nun, I., Pichara, K., Protopapas, P., & Kim, D.-W. (2014). Supervised detection of anomalous light curves in massive astronomical catalogs. *The Astrophysical Journal*, *793*(1), 23.

Nun, I., Protopapas, P., Sim, B., Zhu, M., Dave, R., Castro, N., et al. (2015). Fats: Feature analysis for time series. *arXiv preprint arXiv:1506.00010*.

Paparoditis, E., & Politis, D. N. (2000). The local bootstrap for kernel estimators under general dependence conditions. *Annals of the Institute of Statistical Mathematics*, *52*(1), 139–159.

Percy, J. R. (2007). *Understanding variable stars*. Cambridge University Press.

Pichara, K., & Protopapas, P. (2013). Automatic classification of variable stars in catalogs with missing data. *The Astrophysical Journal*, *777*(2), 83.

Pichara, K., Protopapas, P., Kim, D., Marquette, J., & Tisserand, P. (2012). An improved quasar detection method in EROS-2 and MACHO LMC datasets. *Monthly Notices of the Royal Academy Society*, *18*(September), 1–18.

Pichara, K., Protopapas, P., & Len, D. (2016). Meta-classification for variable stars. *The Astrophysical Journal*, *819*(1), 18.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Quinlan, R. (1993). 4.5: Programs for machine learning morgan kaufmann publishers inc. *San Francisco, USA*.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning (adaptive computation and machine learning)*. The MIT Press.

Richards, J. W., Starr, D. L., Butler, N. R., Bloom, J. S., Brewer, J. M., Crellin-Quick, A., et al. (2011). On Machine-learned Classification of Variable Stars with Sparse and Noisy Time-series Data. *The Astrophysical Journal*, *733*.

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *Is&t/spie's symposium on electronic imaging: Science and technology* (pp. 861–870).

Udalski, A., Soszynski, I., Szymanski, M., Kubiak, M., Pietrzynski, G., Wyrzykowski, L., et al. (2008). The optical gravitational lensing experiment. ogle-iii photometric maps of the large magellanic cloud. *arXiv preprint arXiv:0807.3889*.

Wachman, G., Khardon, R., Protopapas, P., & Alcock, C. R. (2009). Kernels for periodic time series arising in astronomy. In *Machine learning and knowledge discovery in databases* (pp. 489–505). Springer.

Wang, P., Khardon, R., & Protopapas, P. (2010). Machine learning and knowledge discovery in databases. *Lecture Notes in Computer Science*, *6323*, 418.