Discovering pulsars and transients with intelligent algorithms

Alex Lisboa-Wright
8928493

School of Physics and Astronomy
The University of Manchester

Final Year MPhys Report

January 2017

Project undertaken in collaboration with Lewis Smith

Project supervised by Dr Michael Keith

This is the first semester report of a full year project

Abstract

1 Introduction

1.1 Millisecond pulsars

Pulsars are rapidly-rotating neutron stars formed during core-collapse supernovae (SNe), in which a star with a mass greater than circa ** solar masses runs out of hydrogen, becomes a supergiant and eventually collapses in on its plasma core. The pressure of the outer layers falling on the core causes it to contract rapidly and heat up further, allowing the protons and electrons in the core to merge via inverse beta-decay to form neutrons and neutrinos. As core contracts, conservation of angular momentum causes it to rotate much faster. As the neutrinos escape the core, a small fraction of them interact with the outer layers, transferring a huge quantity of energy which expels the layers violently - this is the supernova.

If the core is below the Chandrasekhar mass limit of about 1.4 solar masses, neutron degeneracy pressure prevents it collapsing. The core is now a rapidly-rotating, extremely dense stellar remnant made of neutrons - a neutron star. If the core's mass is greater than the Chandrasekhar limit, the gravitational pressure is stronger than the neutron degeneracy pressure, causing the core to collapse further into a stellar black hole.

Pulsars are characterized by a highly-stable period of rotation and pulse profile within each period, although there are significant differences between the profiles of different pulsars.

Millisecond pulsars (MSPs) are pulsars with a particularly high rotation frequency (low period), with the upper period limit for MSPs typically in the range of 10-30 milliseconds (ms). MSPs are of particular interest because their rotation rates are too high for isolated objects(ref **). Therefore, these pulsars are thought to undergo so-called "spin-up" processes after the supernova event in which they were created.

1.2 The High Time Resolution Universe (HTRU) Survey

HTRU is a survey conducted using the Parkes Radio Telescope in Australia for the southern hemisphere and the Effelsberg Radio Telescope in Germany for the northern hemisphere from 20** to 20** (ref Keith HTRU **), with the purpose of scanning the entire plane of the sky for pulsar and transient signals at the highest resolution used so far in the field.

The survey's classification of raw data relies on the Stuttgart Neural Network Simulator (SNNS), an artificial neural net (ANN) created at the University of Stuttgart which uses 22 different features (***), including the period and dispersion measure as well as how the pulse profile shape matches to various curves, such as Gaussians.

1.3 Motivation

The aim of this project is to improve the performance of machine learning algorithms on classification of pulsars in general and millisecond pulsars in particular, given their properties and the potential implications of those properties.

Currently, candidate classification is carried out by processing the data into graphical form (see Figure ***), which can be analysed by eye as having characteristics of a pulsar (ref Morello SPINN). For large quantities of data, this is

time-consuming and consequently requires that the data be stored until it is analysed, which for the rate of data production of surveys such as the SKA is infeasible. Therefore, the objective of using machine learning techniques is to separate data for potential astrophysical sources from the vast majority of received data, which is noise or radio frequency interference (RFI), which refers to Earth-based radio sources at observing frequencies, such as microwave ovens (ref **), in real time. This avoids the need to store impossibly large quantities of uninteresting data while, ideally, retaining all of the significant observations for further study using existing analysis techniques.

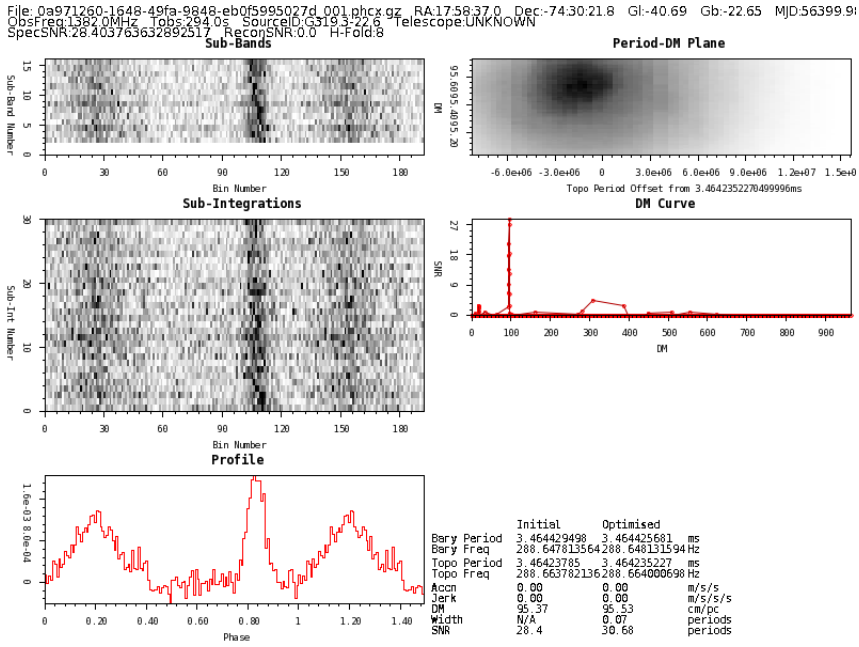

Figure 1: psrsoft image output for a simulated pulsar data file

2 Machine learning

2.1 Theory

ML theory here. Cross validation and training sets/test set.

Fitting/overfitting, decision trees vs ANN.

The (information) entropy of a feature X, $H(X)$, is defined as:

$$H(X) = -\sum_{x \subset X} P(x) \log_2 P(x) \tag{1}$$

where $P(x)$ is the probability of x occurring. The condition entropy, of a feature X given another feature Y, is similarly defined as:

$$H(X|Y) = -\sum_{y \subset Y} P(y) \sum_{x \subset X} P(x|y) \log_2 P(x|y) \tag{2}$$

where $P(y)$ is the probability of y occurring and $P(x|y)$ is the probability of x given y. The mutual information (MI), $I(X;Y)$, between X and Y is defined in terms of these entropies by:

$$I(X;Y) = H(X) - H(X|Y) \tag{3}$$

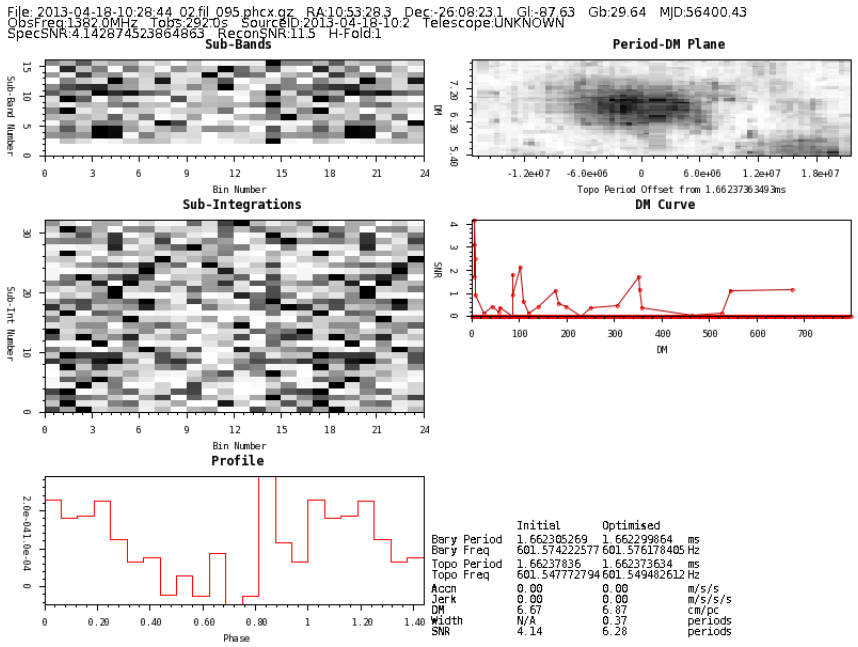

Figure 2: psrsoft image output for a non-pulsar (noise) data file

The results of binary classification come in the form of the true positive (TP), false positive (FP), true negative (TN) and false negative (FN) totals, which denote the number of each class (positive or negative) that were classified correctly or incorrectly, respectively, by the learning algorithm. These can be visualised concisely as the so-called confusion matrix, C:

$$C = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

These numbers are important in defining the following measures of accuracy, which were used in this project. The recall, R, is the fraction of all true positives that were classified as being positive:

$$R = \frac{TP}{TP + FN} \tag{4}$$

The precision, P, is the fraction of the classified positives which are actually positives:

$$P = \frac{TP}{TP + FP} \tag{5}$$

The specificity, S, is the negative analogue to the recall, as it is the fraction of true negatives that are correctly classified:

$$S = \frac{TN}{TN + FP} \tag{6}$$

The false positive rate, FPR, is the fractional remainder of classified negatives, i.e. the fraction of positives incorrectly classified as negatives:

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

The G-mean, G, is the **..... and is defined as:

$$G = \sqrt{R \times S} \tag{8}$$

The F-score, F, is a measure that accounts for both precision and recall:

$$F = 2 \times \frac{P \times R}{P + R} \tag{9}$$

Finally, the overall accuracy, A, is the fraction of all of the data points which are correctly classified:

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

For datasets with a large imbalance between classes, such as data points from pulsar surveys, which are by a large majority non-pulsars, the accuracy became much less sensitive towards the algorithm's performance on the smaller class, i.e. pulsars in the previous example, which is problematic if, as in this project, the performance on the smaller class has more bearing on the final result. Therefore, the recall was of greater interest than the accuracy for this project.

2.2 Algorithms

The algorithms used in this project covered a wide range of properties and behaviours, since the aim of the project was to improve, and therefore prioritise, overall classifier performance in a general scenario, rather than focus on particular properties of certain algorithms. The algorithms used were:

- CART_tree:
- (MLP):
- NaiveBayes: Uses a simple approach in which the classification parameters are assumed to be conditionally independent, given the class, and then applies Bayes' theorem to calculate to probability of each class given the parameters. (** ref James, Langley)
- Support Vector Machine (SVM):
- RandomForest: An ensemble algorithm that compares a group of decision tree to optimise the training set.
- AdaBoost:

Additionally, the Gaussian-Hellenger Very Fast Decision Tree (GH-VFDT), as detailed by Lyon et al (ref **), was used to confirm the results of Table 11 in the same paper, but was then discarded.

3 Experimental background

3.1 Astrophysical measurements

There are a number of important astrophysical measurements that are outputs of the telescope data. For instance, the average pulse profile is a graph of the average received amplitude as a function of the phase during an average pulse period. This profile is shown in the bottom left panel of Figures 1 and 2 as a histogram. By comparing the panels of the two figures, it is clear that a signal from a pulsar will have a more obvious curve (in red) and a clear peak or peaks. Weltevrede et al. (ref **) demonstrate that there are a wide variety of pulse profile shapes, including multiple-peaked and heavily asymmetric profiles.

The signal-to-noise ratio (SNR) is simply the ratio, in a given reading, of the power from the source itself (the signal) versus that from other (background) objects, such as the cosmic microwave background (CMB), the atmosphere, the interstellar medium (ISM) or other origins (depending on the observing frequency). Readings with a higher SNR have a smaller relative error via **Poisson statistics or route-N errors**, so are more likely to represent real source emission. The SNR can, however, be rendered less effective as a classification parameter by RFI (ref Zhu et al. image pattern).

The dispersion measure (DM) is defined in terms of the electron number density, n_e, along the line of sight to the source (ref indian ** guys):

$$DM = n_e dl \tag{11}$$

The units of the DM are usually quoted as pc cm^{-3}, i.e., the DM can be interpreted as an electron column density along the line of sight. The DM is important because an ionized medium (such as the ISM) situated in between the observer and the source can cause the radiation from the source to become dispersed by refraction or absorption and re-radiation, depending on the relative magnitude of the light's wavelength and the typical (one-dimensional) separation between the electrons in the medium. This causes a wavelength-dependent (and hence frequency-dependent) time delay in the reception of the signal. For two different frequencies, f_1 and f_2, the delay is given in SI units by:

$$\delta t = \frac{e^2}{2\pi mc}(f_1^{-2} - f_2^{-2})DM \tag{12}$$

3.2 Applications of machine learning to radio astronomy

This project uses data and the PulsarFeatureLab processing pipeline from Lyon et al. The main application of machine learning to radio astronomy will be for simultaneous recording and classification of data from large surveys such as the Square Kilometre Array (SKA), based at the Jodrell Bank Centre for Astrophysics (JBCA) in Manchester and with observing instruments in South Africa and western Australia, which is the largest radio instrument in history in terms of data collection **. Radio observations require much more data to be recorded than is the case for other frequency instruments, such as optical telescopes. The SKA, during its predicated observing lifespan, is expected to produce quantity of data equivalent to 100 times the estimated total current global data storage capacity (ref **).

4 Data and pipelines

Pulsar Feature Lab was used to process the original and simulated data to extract the desired features. It is a processing pipeline created by Lyon et al. (**) designed to be compatible with various data file formats and has multiple output file type options. In addition, its feature extractor program can output any of a selection of feature groups, and users can add their own feature list to this selection.

The datasets available for the chosen parameters were labelled as follows:

- HTRU1: Dataset from HTRU processed by ** et al. Contained 74 classified MSPs, 1122 other classified pulsars and 89996 noise instances.
- HTRU2: Dataset from HTRU processed by Lyon et al. using PulsarFeatureLab. This was the principle test dataset in the first semester of the project. Contained 71 classified MSPs, 1568 other classified pulsars and 16259 noise instances.
- LOTAAS: A dataset from the LOFAR Tied-Array All-Sky Survey (LOTAAS). This did not contain any classified MSPs and was therefore not useful in this project.

In this project, MSPs and other pulsars were distinguished by defining a threshold pulse period, below which classified pulsars were treated as MSPs and above which they were treated as non-MSPs. The definition described by Lorimer (2008)(ref **Binary and Millisecond Pulsars) of P_{MSP} **$<\approx$ 30 ms was used as a guideline. The MSPs in the datasets, in order to include a few pulsars with periods slightly above 30ms, were defined as P_{MSP} **$<\approx$ 31 ms. This was done to maximise the number of MSPs upon which to test the classifiers, although even with this there were relatively few MSPs in the datasets, as detailed above.

For this project, the output files were chosen to be .arff (A* R* F* F*) files in order to be compatible with the Waikato Environment Knowledge Analysis (WEKA) machine learning tool, which was used to visualise and count the different data types and thus find potentially useful relationships between parameters to optimize simulated data.

5 Experimental procedure

All new codes for this computer-based project were written in Python 2.7.12. The algorithms detailed in Table 11 of Lyon et al. (2016) were run on all three datasets to test the reproducibility of the results. Then, with the exception of the GH-VFDT, they (or their ScikitLearn equivalents) were used as classifiers for the remainder of the experiment, together with the AdaBoost and RandomForest ensemble algorithms.

To produce the best results, the data features had to be selected to give the best distinction between pulsars and non-pulsars while remaining easily processable. Feature selection was achieved using mutual information and then using ranked joint mutual information (JMI) to list the best features and therefore select the better of the feature groups. Two groups were compared:

- Lyon features: Detailed by Lyon et al.(2015)**, this is a set of 8 features consisting of 4 statistical measures - the mean, standard deviation, skewness

and kurtosis, which are the first, second, third and fourth statistical moments, respectively - applied to the pulse profile and DM-SNR curve.

- Thornton features: Detailed by Thornton (2013)**, this is the set of 22 features originally implemented for the SNNS (see Section 1.2), including the period, SNR, DM and fittings to various hypothetic pulse profiles.

To create the simulated pulsars, pulses were created with a uniform distribution of periods in the range **5-20 milliseconds in order to coincide with the expected area of interest for future MSP discoveries after consultation with the project supervisor. The simulated pulse profiles were injected into real noise data files to make the curves more realistic than using the simulated profiles alone, as these would be far too smooth and therefore too easy to distinguish from real data. To generate the final simulated data the injection was performed on a multiple-core GPU and left to run, due to the large processing power required. This produced directories each containing one simulated MSP data file and several noise files.

To extract the pulsar file, a script called "find_fake_pulsar.py" was created to allocate each file a score which was a sum of the fractional deviations of the pulse period and the DM value from the original injected data values. Therefore, the pulsar file would ideally have a score of zero. However, the fitting during the process ** allowed the parameters to be adjusted such that there were variations in them. The program compared the scores within each directory and took the lowest score forward. The relevant data file was then tested to determine whether its period and DM were within a strict tolerance level of the original injected values. Only if they passed this condition were they named as being the pulsar files, to ensure that the files which were produced were likely to be the correct ones. If no file met the requirements, a message to that effect was produced instead of a file name.

These file names were then grouped together

A test of significance was carried out on the HTRU2 results generated before and after added the simulated MSPs to the training data, in the form of a student's t-test, to determine the so-called p-value. This generates a result, known as the p-value, by comparing the two results and using their respective standard deviations **. This is then used to assess the validity, in this situation, of the null hypothesis, which states that any difference between two sets of results is purely due to random errors in the results and thus the results are equivalent**. (ref **). If the significance test generates a p-value less than p_{thres}, the null hypothesis is rejected. If it generate a value greater than p_{thres}, the null hypothesis is accepted. The threshold p-value was chosen as $p_{thres} = 0.05$, which is a typical value and indicates that **.

Having used HTRU2 as a dataset to optimize the simulated data, the simulated data was added to HTRU1 and the algorithms were then applied to the resulting dataset.

As an extension, a program, "plot_classifier_eval.py", was written that would use the combined HTRU2 and simulated training dataset to determine the effect of varying the MSP cutoff period, P_{MSP}, on the MSP recall. This was carried out by entering a quantity and range of P_{MSP} values as arguments, then plotting the mean recalls against P_{MSP}, with error bars, for each classifier. This was carried out for both the newly-defined MSPs and all pulsars, as a reference. The results are shown in Figure 4.

6 Results

The results detailed in Table 11 of Lyon et al. (2016) were successfully reproduced, demonstrating their accuracy. The mutual information and joint mutual information (JMI) were calculated.

tables graphs

Classifier	R **for HTRU2 data**	R **for HTRU2 + simulated data**
CART_tree	0.301 ± 0.042	0.594 ± 0.023
MLP	0.383 ± 0.027	0.603 ± 0.012
Naive_Bayes	0.380 ± 0.000	0.465 ± 0.000
SVM	0.487 ± 0.013	0.682 ± 0.008
Random_Forest	0.293 ± 0.058	0.563 ± 0.039
AdaBoost	0.358 ± 0.016	0.558 ± 0.008

Table 1: Recall on HTRU2 MSPs (class 1) using the HTRU2 noise (class 0) and non-MSPs (class 1) as training data

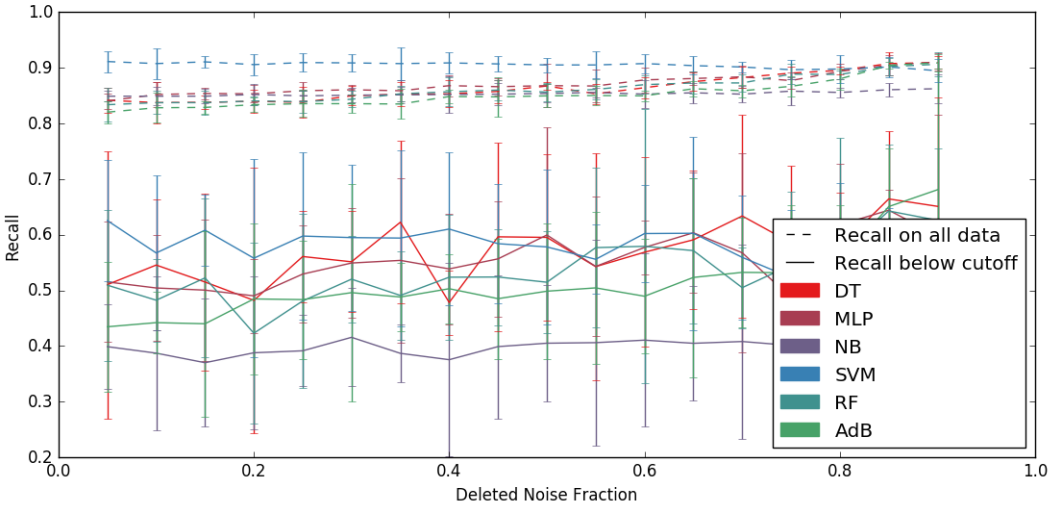

Figure 3: Recall

Varying P_{MSP}, surprisingly, had no significant effect on the recall of MSPs. The SVM classifier again proved to be the superior classifier, this time for all P_{MSP},

Classifier	R **for HTRU1 data**	R **for HTRU1 + simulated data**
CART_tree	0.819 +/- 0.040	0.859 +/- 0.025
MLP	0.816 +/- 0.012	0.854 +/- 0.034
Naive_Bayes	0.811 +/- 0.000	0.824 +/- 0.000
SVM	0.981 +/- 0.007	0.986 +/- 0.000
Random_Forest	0.822 +/- 0.015	0.846 +/- 0.039
AdaBoost	0.789 +/- 0.028	0.792 +/- 0.008

Table 2: Recall on HTRU1 MSPs (class 1) using the HTRU1 noise (class 0) and non-MSPs (class 1) as training data

Classifier	p-value for HTRU2 data	p-value for HTRU1 data
CART_tree	0.000785	0.267
MLP	0.000372	0.189
Naive_Bayes	0.000230	0.115
SVM	3.94E-05	0.340
Random_Forest	0.00430	0.273
AdaBoost	7.38E-05	0.924

Table 3: p-values from a student's t-test of significance for adding simulated data. The threshold p-value for both is 0.05.

Classifier	R for HTRU2 data	R for HTRU2 with noise reduction
CART_tree	0.301 ± 0.042	0.479 ± 0.023
MLP	0.383 ± 0.027	0.428 ± 0.016
Naive_Bayes	0.380 ± 0.000	0.372 ± 0.008
SVM	0.487 ± 0.013	0.490 ± 0.006
Random_Forest	0.293 ± 0.058	0.417 ± 0.068
AdaBoost	0.358 ± 0.016	0.420 ± 0.027

Table 4: Recall on HTRU2 MSPs (class 1) using the HTRU2 noise (class 0) and non-MSPs (class 1) as training data

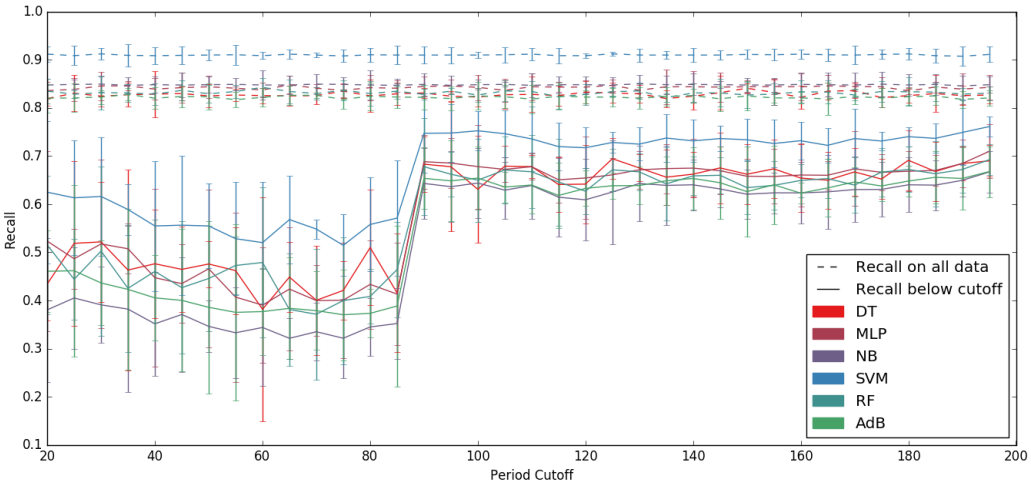

Figure 4: Recall on MSPs for variation in the MSP cutoff period

noise reduction

7 Discussion

The addition of simulated data clearly and significantly improves the performance of the classification algorithms on the HTRU2 data. This is not only due to the realistic appearance of the data in the parameter space, but also to the impact of adding a large number of MSPs to the data population, as it counters the dominance of non-millisecond pulsars within the dataset of classified pulsars and noise within

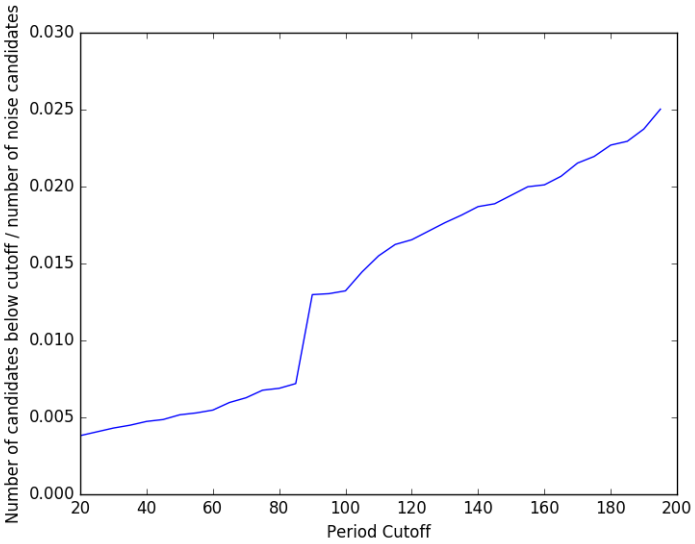

Figure 5: Recall

the full dataset.

The performance of the Naive_Bayes classifier was the poorest for any given dataset in HTRU2. During repeated runs, even as the other classifiers' results fluctuated slightly, the Naive_Bayes results did not change at all, and the standard deviation was so small as to be effectively zero, somtimes being quoted by the program as exactly zero. This clearly indicates that this classifier is not useful in the context of the parameters chosen. This problem can be explained by the clear relationships between some of the parameters used - these relationships were, in fact, used to optimize the simulated data to give the final, improved classification results. As detailed earlier, the Naive_Bayes classifier assumes the parameters it uses to be independent of one another, which is clearly contradicted by the aforementioned trends in the data.

HTRU1 was not improved by any level of significance by the addition of the final simulated data. However, the classifiers performed very well without adding the simulated data, and adding it did not make the results worse - had the simulated data been unrealistic for HTRU1, it would have been expected that the results would have deteriorated.

It must be noted, however, that HTRU1 and HTRU2 were processed using different pipelines methods and codes, which may explain how two sets of data from the same survey can produce such different performance levels from the same classifier algorithms and for every one of those algorithms.

8 Conclusion

For HTRU2, using only simulated MSPs or real non-MSPs as the class 1 objects in the training set produced low recall numbers on the real MSPs for the classifiers. However, adding them both to the training data caused a highly significant increase in the recall for all of the (diverse) classifier algorithms.

For HTRU1, this was not the case. However, using the real non-MSPs alone as the class 1 training data produced a high recall across all classifiers anyway, leaving

little room for improvement after adding simulated MSPs. Doing so did not change the results significantly, not even detrimentally.

Given the different behaviour of the same classifiers across all scenarios between the two datasets and given the fact that those sets were each processed by different groups using different computing pipelines, it is highly probable that the two facts are causally connected, i.e. that the HTRU1 data has been "cleaned" to a greater extent than the HTRU2 data, hence the pulsar data points (most of which are for non-MSPs) are more easily classified correctly and it is difficult to introduce simulated MSP data which conforms to the same behaviour as the existing data.

For HTRU2, the recall on MSPs appears to be unaffected by the position of P_{MSP}, which is surprising.

9 Direction for future work

This project, as its title implies, is aimed at improving classification of both pulsars and transients such as fast radio bursts (FRBs). We hope to apply the techniques and experience gained with pulsars during this semester to FRBs next semester. Given the rarity of FRBs (only ** comfirmed so far) and the substantial similarity between their signals and RFI, as detailed in ** et al. (20**), it will require fine-tuning. The features used will have to be evaluated again - the Lyon features would appear to be a favourable starting point due to the statistical (as opposed to physical) nature, which could allow these features to be more easily generalised to non-pulsar target data. However, this is not guaranteed.

References

10 Appendix

10.1 Source code repository

**GIT repo URL here!!!!

10.2 Risk assessment

1. Hazard identification:

(i) High computer usage and the subsequent risk of sight problems due to over-exposure to screens.

(ii) Electrocution

2. People at risk:

(i) MPhys participants

(ii) MPhys participants

3. Risk evaluation:

(i) The risk is adequately controlled due to industry standards and user guidance regarding computer screen usage.

(ii) The risk is adequately controlled due to grounding and insulation of wires.

4. Actions to be taken:

(i) Take regular breaks away from computer screens.

(ii) Follow standard guidelines for use of electrical equipment.

Last reviewed: 15th December 2016
Carried out by: Alex Lisboa-Wright, Lewis Smith. Supervisor: Michael Keith.