Imbalanced Ensemble Learning for Enhanced Pulsar Identification

Jakub Holewik[1], Gerald Schaefer[1](✉), and Iakov Korovin[2]

[1] Department of Computer Science, Loughborough University, Loughborough, UK
[2] Southern Federal University, Taganrog, Russia

Abstract. Pulsars can be detected based on their emitted radio waves. Machine learning methods can be employed to support automated screening of a large number of radio signals for pulsars. This is however a challenging task since training these methods is affected by an inherent imbalance in the acquired data with signals relating to actual pulsars being in the minority.

In this paper, we demonstrate that ensemble classification methods that are dedicated to imbalanced classification problems can be successfully employed for pulsar identification. Classifier ensembles combine several individual classifiers to yield more robust and improved classification, while class imbalance can be addressed through careful sampling or through cost-sensitive classification. Experimental results, based on HTRU2 data, show that the investigated ensembles outperform methods that do not consider class balance, and suggest their use for other applications in astrophysics.

Keywords: Pattern classification · Ensemble classifier · Class imbalance · Astrophysics · Pulsar identification

1 Introduction

Pulsars are rare neutron stars detectable through the radio waves they emit [23]. To allow for automated screening of a large number of radio signals, machine learning methods can be adopted. However, this is hampered by a considerable amount of noise as well as radio frequency interference. Interference is so common that the large majority of signals detected turn out to not stem from pulsars. This makes the use of common machine learning algorithms challenging since they are not designed to take into account such a class imbalance.

Class imbalance is a common issue in pattern classification tasks. When collecting data for training, in particular for binary classification (i.e., tasks where patterns are separated into exactly two classes as in “pulsar” and “not pulsar”), the ideal scenario is that the split of patterns across the classes will be approximately equal. Unfortunately, when collecting data for pulsar candidates, only a handful of observations will correspond to true pulsars. Standard classifiers struggle to learn successfully from such imbalanced datasets and in particular

Y. Tan et al. (Eds.): ICSI 2020, LNCS 12145, pp. 515–524, 2020.
https://doi.org/10.1007/978-3-030-53956-6_47

to learn well from the minority class, which for our problem here (real pulsars) is the one of interest.

In this paper, we show that ensemble classification methods, i.e. methods that combine several individual classifiers, that are dedicated to imbalanced classification problems can be successfully employed for pulsar identification. Our results demonstrate that such dedicated ensembles yield better results compared to methods that do not consider class balance, and suggest their use for other applications in the field of astrophysics.

2 Background

Searching for pulsars is conducted through collecting pulsar candidates, that is, sets of statistical information about certain radio emissions captured from space [18]. Search techniques used to isolate these look for periodic broadband signals that appear dispersed. The collected signals are analysed to determine which of them are actual pulsars. Signals that are determined to be likely coming from pulsars are then passed on for further observation. Traditionally, this analysis was conducted manually by human experts. Unfortunately, the majority of captured signals do not come from pulsars, leading to a lot of time dedicated to discarding noisy candidates. In addition, technological advancements have significantly increased the number of candidates being discovered [26], leading to the manual approach becoming infeasible.

Consequently, various automated approaches have been developed for pulsar classification. For example, [12] describes a computer program for candidate selection as early as 1992. However these methods were not intelligent enough, since after initial filtering of candidates, the selected samples still needed to be manually checked. The use of advanced machine learning approaches leads to more reliable detection. Examples include an artificial neural network for pulsar classification [8], PICS, a method which utilises image pattern classification approaches to recognise pulsars from diagnostic plots [30], and SPINN, a high-performance solution that is also based on neural networks [19].

While such machine learning approaches have shown potential to greatly reduce the work needed for identifying pulsars, none of these solutions address the fundamental problem of data imbalance present in the candidate selection task with true pulsars being greatly outnumbered by noisy samples. It is this aspect that we specifically address in this paper.

3 Imbalanced Classification

Many real-life datasets are imbalanced so that patterns of interest (commonly referred as the positive class) are outnumbered (making it the minority class) by “other” patterns (referred to as the negative class and majority class). In our pulsar detection task, there are many more noise samples compared to those that represent a true pulsar. This is challenging as classification algorithms typically try to maximise accuracy over all samples and thus tend to be biased towards the majority class, leading to poor recognition of minority class samples.

A common approach to address class imbalance is sampling which tries to "fix" the dataset. The goal here is to create a new training dataset with equal class distribution. In undersampling approaches, this is achieved by removing patterns of the majority class, at the cost of discarding potentially useful data [29]. On the other hand, in oversampling, the number of patterns of the minority class is increased to match more closely that of the majority class. The key obstacle here is how to obtain useful new minority class samples.

SMOTE (for Synthetic Minority Over-sampling TEchnique) [4] generates new, artificial patterns of the minority class that are designed to be similar to the actual patterns in the dataset. For this, it uses a nearest neighbour approach, creating new patterns by combining features from existing neighbouring patterns. SMOTE has been widely used and is known to help with generalisation in imbalanced classifiers [4,29].

A different approach to address class imbalance is cost-sensitive classification which is based on the idea of assigning a cost to misclassifications [20]. Conventional classifiers try to reduce the number of misclassifications but do not pay attention to which class they belong. Introducing class costs is a common approach to reflect the varying degrees of importance among classes. In most imbalanced classification problems, the minority class is the class of interest and is thus assigned a higher cost so that the resulting classifier will focus more on reducing the error rate of that class.

4 Classifier Ensembles

Traditionally, a single classifier is used in pattern recognition problems. However, classifiers are rarely perfect and designing a classifier that generalises well is a difficult problem. On the other hand, different classifiers can complement each other when it comes to achieving high performance [10]. This observation leads to the development of multiple classifier systems, also known as ensemble classifiers. Using separate classification methods simultaneously, and then combining their outputs, these methods can deal particularly well with noisy inputs and yield more robust classification [11].

In general there are three reasons for why ensemble methods are worth using [7]:

- *Statistical argument:* This is relevant in problems that suffer from data sparsity. With a limited number of patterns, training could produce many different classifiers with similar performance. But some of them will be better than others at generalisation. Combining these classifiers into an ensemble is better than picking one and risking that it will not perform well on unseen data.
- *Computational argument:* This argument applies to methods that use some sort of hill-climbing or random search, for example neural networks with gradient descent or decision trees with greedy splitting rules. These techniques are inherently difficult to optimise and a common issue is the search getting

stuck in a local optimum. This is where an ensemble can be beneficial by utilising multiple classifiers which begin searching in different places, thus improving the likelihood of finding the global optimum.
- *Representational argument:* It may be impossible to obtain an optimal classifier. For example, for a dataset with a non-linear decision boundary, there is no linear classifier that can achieve perfect classification. In a situation like this, there are two solutions. One is to train a classifier of higher complexity, while the other is to combine some imperfect classifiers with the aim of increasing the overall performance.

The most common approaches of ensemble learning are bagging and boosting. In bagging [1], the idea is that each classifier is not trained on the full dataset. Instead, the dataset is randomly sampled multiple times (with replacement), and each classifier trained on a different subset. All classifiers' outputs are then combined. Boosting revolves around "weak learners", i.e. classifiers that are only slightly better than a random guess [24]. These can be effectively transformed, or "boosted", into strong learners by iteratively training ensemble members, with each of them focussing on specific data patterns that were difficult to learn for the previous classifiers.

5 Ensemble Classifiers for Imbalanced Data

In this paper, we investigate a number of ensemble classifiers that specifically address class imbalance. In the following, we briefly describe the algorithms that we use.

5.1 SMOTEBagging

SMOTEBagging combines a bagging ensemble with various sampling strategies [28]. The main idea is to employ a bagging scheme that trains individual classifiers on subsets of the training data so that each class is equally represented. SMOTE is applied to the minority class, and all original samples along with the generated patterns are used alongside a random subset of the majority class when training each classifier. Experimental results in [28] show that SMOTE effectively improves the diversity and performance of a bagging ensemble.

5.2 SMOTEBoost

[5] proposes a method that also utilises SMOTE, but in combination with an AdaBoost classifier. SMOTE is employed to improve the performance on the minority class, while boosting is used make up for the loss of general accuracy. SMOTEBoost is more sophisticated than just simply running SMOTE on a dataset and then training an ensemble on it. SMOTE is instead used separately for each classifier, and all synthetically generated patterns are discarded before training the next classifier. Unlike standard AdaBoost, which treats all

misclassifications equally, misclassified minority class patterns are focussed on. Particularly hard to learn minority patterns will have "similar" synthetic patterns with similar weights added to the training set, thus enabling classifiers to better learn them, while implicitly creating more diversity in the ensemble (since every classifier is trained on a number of exclusive patterns that will be discarded afterwards).

5.3 EasyEnsemble

[16] proposes an effective ensemble which focusses on undersampling rather than oversampling. EasyEnsemble can be seen as a fusion between bagging and boosting but is somewhat unique in that it technically generates an ensemble of ensembles. During each iteration, it uses random undersampling with replacement to generate a subset of the majority class training data. This subset is then used together with the full minority class data as a training set for an AdaBoost ensemble. This way, a set of diverse AdaBoost ensembles is generated, each trained on different majority class data. Finally, the outputs of all classifiers predicting the same class are summed, and the class with higher support is chosen.

5.4 Balanced Random Forest

In a similar fashion to EasyEnsemble, the Balanced Random Forest algorithm [6] adapts standard the Random Forest algorithm [2] for imbalanced data. The algorithm uses undersampling on the majority class, and so each tree receives a subset with an equal spread between the classes.

5.5 AdaC2

AdaC2 utilises AdaBoost with a cost-sensitive approach to address data imbalance [27]. The goal is to adjust the weights so that misclassified minority class patterns are the main focus. The algorithm uses a cost value for each pattern which represents the penalty to the classifier for misclassifying that pattern with minority class costs higher than majority class costs. These costs are then incorporated into the weights as

$$w_i^{k+1} = \frac{w_i^k e^{\beta_i} C_i}{Z}, \tag{1}$$

where w_i^{k+1} is the weight of pattern i in the next classifier, w_i^k is that pattern's weight in the current classifier, Z is a normalisation factor, β is a parameter such that $\beta_i = \alpha_k$ if pattern i was misclassified by classifier k, and $\beta_i = -\alpha_k$ otherwise, with α_k a parameter that is some predefined function of the classifier's error rate.

5.6 AdaCost

AdaC2 and related algorithms are actually simplified variants of another cost-sensitive method, namely AdaCost [9]. In this algorithm, the AdaBoost cost function is

$$w_i^{k+1} = \frac{w_i^k e^{\beta_i D_i}}{Z}. \tag{2}$$

Here, instead of introducing a constant cost for each pattern, a cost adjustment function D_i is used, which is designed to have higher values when the pattern was misclassified. An interesting aspect of this algorithm is that, unlike AdaC2, it does not reduce to AdaBoost when both the majority and minority class are given the same weight [27].

6 Pulsar Classification

In this paper, we perform pulsar classification based on the HTRU2 study which produced a large database of pulsar candidates collected in the dedicated High Time Resolution Universe Survey [13].

The features are extracted from the pulse profile which describes the longitude-resolved version of a the signal, averaged in frequency and time. The DM-SNR curve represents the correlation between the dispersion measure (DM; the integrated density of free electron columns between the pulsar and the point of observation) and the signal-to-noise ratio (SNR) from the given pattern.

Specifically, we use eight attributes that represent the various features of each pulsar candidate [19], namely:

- mean of the integrated pulse profile;
- standard deviation of the integrated pulse profile;
- excess kurtosis of the integrated pulse profile;
- skewness of the integrated pulse profile;
- mean of the DM-SNR curve;
- standard deviation of the DM-SNR curve;
- excess kurtosis of the DM-SNR curve;
- skewness of the DM-SNR curve.

The dataset[1] comprises 16,259 bogus patterns (caused by radio frequency interference and noise) and 1,639 real pulsar patterns which have been manually verified, thus exhibiting significant class imbalance.

7 Experimental Results

In a binary classification problem, there are four basic measures which are used to define various performance metrics [25]:

[1] https://archive.ics.uci.edu/ml/datasets/HTRU2/.

- *True Positives (TP):* Number of patterns from the positive class that are correctly classified;
- *True Negatives (TN):* Number of patterns from the negative class that are correctly classified;
- *False Positives (FP):* Number of patterns from the negative class that are incorrectly classified as the positive class;
- *False Negatives (FN):* Number of patterns from the positive class that are incorrectly classified as the negative class.

From these, we can calculate [25]:

- *Accuracy:* is the overall percentage of correctly classified patterns and is defined as
$$Acc = \frac{TP + TN}{TP + FN + FP + TN}; \tag{3}$$
- *Precision:* is the percentage of patterns classified as positive that are correct and is defined as
$$Prec = \frac{TP}{TP + FP}; \tag{4}$$
- *Recall* or *Sensitivity:* is the percentage of positive patterns classified correctly and is defined as
$$Rec = Sen = \frac{TP}{TP + FN}; \tag{5}$$
- *Specificity:* is the percentage of negative patterns classified correctly and is defined as
$$Spec = \frac{TN}{TN + FP}. \tag{6}$$

For imbalanced classification problems, the following measures are more useful:

- *F-score* [22]*:* is defined as the harmonic mean of precision and recall,
$$F = \frac{2 * Prec * Rec}{Prec + Rec}; \tag{7}$$
- *G-mean* [15]*:* is defined as the geometric mean of sensitivity and specificity,
$$G = \sqrt{Sen * Spec}. \tag{8}$$

As base classifiers, we employ decision trees and support vector machines (SVMs), except for the Balanced Random Forest which is inherently based on tree classifiers. For comparison, we also implement conventional AdaBoost as a standard ensemble classifier.

The obtained results are given in Table 1 for tree classifiers and in Table 2 for SVMs.

From Tables 1 and 2, we can see that AdaBoost gives fairly good performance. This confirms that the extracted features provide a good basis for successful pulsar identification. However, ensembles that are dedicated to imbalanced classification problems do, with the exception of the Balanced Random Forest, give

Table 1. Experimental results using decision trees as base classifiers.

	Accuracy	Precision	Recall	Specificity	G-mean	F-score
AdaBoost	97.96	94.80	81.21	99.57	89.92	87.48
SMOTEBagging	96.66	76.64	89.11	97.39	93.16	82.40
SMOTEBoost	96.63	76.48	88.93	97.37	93.06	82.24
Balanced Random Forest	96.15	73.84	86.96	97.04	91.86	79.86
EasyEnsemble	95.89	71.10	89.60	96.50	92.99	79.29
AdaC2	97.82	90.41	84.08	99.14	91.30	87.13
AdaCost	97.82	90.41	84.08	99.14	91.30	87.13

Table 2. Experimental results using SVMs as base classifiers.

	Accuracy	Precision	Recall	Specificity	G-mean	F-score
AdaBoost	96.82	94.64	67.52	99.63	82.02	78.81
SMOTEBagging	97.33	81.84	89.44	98.09	93.67	85.48
SMOTEBoost	96.97	81.29	84.95	98.12	91.30	83.08
EasyEnsemble	97.22	80.76	89.68	97.95	93.72	84.99
AdaC2	98.04	94.20	82.80	99.51	90.77	88.14
AdaCost	97.96	92.58	83.44	99.36	91.05	87.77

significantly better results, in particular in terms of yielding both high G-mean and F-score results combined with high sensitivity, thus confirming the usefulness of the presented approaches. Looking more closely at the different ensembles, we can see that EasyEnsemble correctly recognises the highest number of true pulsars, though at the trade-off of misclassifying more patterns from the majority class. Overall, SMOTEBagging gives the best balance and provides good classification for both classes, while the use of SVMs as base classifiers is generally superior to decision trees.

8 Conclusions

In this paper, we have shown that ensemble classifiers that address class imbalance represent a useful approach for finding true pulsars among the candidates in the HTRU2 study. Since the investigated methods are essentially agnostic with respect to the application, we expect that they can also be successfully employed for other astrophysical applications such classification of photometric variable stars [21], supernovas [17] or globular clusters [3]. We also currently investigate these ensemble classifiers for imbalanced classification problems in object classification and video analysis [14].

Acknowledgements. This paper is published due to the financial support of the Federal Target Programme of the Ministry of Science and Higher Education of Russian Federation, project unique identifier RFMEFI60819X0281.

References

1. Breiman, L.: Bagging predictors. Mach. Learn. **24**, 123–140 (1996)
2. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
3. Cavuoti, S., et al.: Astrophysical data mining with GPU. A case study: genetic classification of globular clusters. New Astron. **26**, 12–22 (2014)
4. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
5. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39804-2_12
6. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Technical report, UC Berkeley (2004)
7. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1
8. Eatough, R., et al.: Selection of radio pulsar candidates using artificial neural networks. Mon. Not. R. Astron. Soc. **407**, 2443–2450 (2010)
9. Fan, W., Stolfo, S., Zhang, J., Chan, P.: AdaCost: misclassification cost-sensitive boosting. In: 16th International Conference on Machine Learning, vol. 99, pp. 97–105 (1999)
10. Ho, T., Hull, J., Srihari, S.: Combination of structural classifiers. In: IAPR Workshop on Syntactic and Structural Pattern Recognition, pp. 123–136 (1990)
11. Ho, T., Hull, J., Srihari, S.: Decision combination in multiple classifier systems. IEEE Trans. Pattern Anal. Mach. Intell. **16**, 66–75 (1994)
12. Johnston, S., et al.: A high-frequency survey of the southern galactic plane for pulsars. Mon. Not. R. Astron. Soc. **255**, 401–411 (1992)
13. Keith, M., et al.: The high time resolution universe pulsar survey I. System configuration and initial discoveries. Mon. Not. R. Astron. Soc. **409**, 619–627 (2010)
14. Korovin, I.S., Khisamutdinov, M.V., Ivanov, D.Y.: A basic algorithm of a target environment analyzer. In: 2nd International Conference on Advances in Artificial Intelligence, pp. 7–11 (2018)
15. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: 14th International Conference on Machine Learning, pp. 179–186 (1997)
16. Liu, X., Wu, J., Zhou, Z.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man. Cybern. Part B **39**, 539–550 (2009)
17. Lochner, M., McEwen, J., Peiris, H., Lahav, O., Winter, M.: Photometric supernova classification with machine learning. Astrophys. J. Suppl. Ser. **225**, 31 (2016)
18. Lyon, R.J., Stappers, B., Cooper, S., Brooke, J., Knowles, J.: Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. Mon. Not. R. Astron. Soc. **459**, 1104–1123 (2016)
19. Morello, V., Barr, E., Bailes, M., Flynn, C., Keane, E., van Straten, W.: SPINN: a straightforward machine learning solution to the pulsar candidate selection problem. Mon. Not. R. Astron. Soc. **443**, 1651–1662 (2014)

20. Nakashima, T., Yokota, Y., Ishibuchi, H., Schaefer, G., Drastich, A., Zavisek, M.: Constructing cost-sensitive fuzzy rule-based classification systems for pattern classification problems. J. Adv. Comput. Intell. Intell. Inf. **11**, 546–553 (2007)
21. Richards, J., et al.: Active learning to overcome sample selection bias: application to photometric variable star classification. Astrophys. J. **744**, 192 (2011)
22. Rijsbergen, C.J.V.: Information Retrieval, 2nd edn. Butterworth-Heinemann, Oxford (1979)
23. Roberts, N., et al.: Handbook of Pulsar Astronomy. Cambridge Observing Handbooks for Research Astronomers. Cambridge University Press, Cambridge (2005)
24. Schapire, R.E.: The strength of weak learnability. Mach. Learn. **5**, 197–227 (1990)
25. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. **45**(4), 427–437 (2009)
26. Stovall, K., Lorimer, D., Lynch, R.: Searching for millisecond pulsars: surveys, techniques and prospects. Class. Quantum Gravity **30**, 224003 (2013)
27. Sun, Y., Kamel, M., Wong, A., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**, 3358–3378 (2007)
28. Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. In: IEEE Symposium on Computational Intelligence and Data Mining, pp. 324–331 (2009)
29. Weiss, G.: Mining with rarity: a unifying framework. SIGKDD Explor. **6**, 7–19 (2004)
30. Zhu, W., et al.: Searching for pulsars using image pattern recognition. Astrophys. J. **781**, 117 (2014)