Imbalanced Classification Using Genetically Optimized Random Forests

Todd Perry
School of Computing
University of Portsmouth
Portsmouth PO1 3HE, UK

Mohamed Bader-El-Den
School of Computing
University of Portsmouth
Portsmouth PO1 3HE, UK

ABSTRACT

Class imbalance is a problem that commonly affects 'real-world' classification datasets, and has been shown to hinder the performance of classifiers. A dataset suffers from class imbalance when the number of instances belonging to one class outnumbers the number of instance belonging to another class. Two ways of dealing with class imbalance are modifying the dataset to reduce the number of instances belonging to the majority class(es) (known as *resampling*), or allowing the classifier to penalize misclassifying the minority class(es) more than the majority class(es), this can be done by implementing a *cost matrix.* This paper attempts to improve the classification performance of the *Random Forest* classifier on imbalanced datasets by exploiting these two techniques, to do this a genetic algorithm is employed to find optimal parameters. Results are compared to commonly used classification algorithms.

Categories and Subject Descriptors

I.2.1 [**I. Computing Methodologies**]: ARTIFICIAL INTELLIGENCEApplications and Expert Systems[Industrial automation]

Keywords

Random Forest; Genetic Algorithms; Classification; Cost-Sensitive Classification; Cost Matrix

1. INTRODUCTION

In recent years, machine learning has quickly made it's way from a purely academic field to a powerful decision making tool. Classification is an example of machine learning, classification algorithms learn to recognize different categories, or *classes*, using the dataset. Once a classification algorithm, or *classifier*, has been trained, it can be used to predict what class an unseen observation belongs to. Datasets where the number of observations belonging to one class greatly out-weights that of anothers are known as imbalanced. This *class imbalance* problem affects many real world problems, and has shown to be detrimental to classifier performance [5].

This paper presents a Genetic Algorithm (GA) based for improving the performance of the Random Forest (RA) algorithm, specifically on imbalanced datasets. The proposed algorithm uses genetic algorithm to automatically optimize the cost-matrix conjunction with the RA parameters. The rest of the paper is structured as follows: Section 2 proposes a solution that uses a genetic algorithm to optimize classification performance. Section 3 is a report of the investigation into the proposed solution and the paper is concluded in section 4.

2. PROPOSED ALGORITHM

The *Random Forest* [2] classifier is used as a base for the algorithm proposed in this paper. RF belongs to Ensemble classification which an established approach in machine learning, RF aims to boost the performance of classification techniques by building a number of classifiers, and then collectively use them all to identify unlabelled instances.Each individual in the *Random Forest* ensemble is known as a *Random Tree* and acts as a regular decision tree except for the fact that not all attributes are taken into account at each node - instead a number of attributes are randomly selected, and then the splitting criterion is calculated for only those selected attributes [2].

The two main methods of dealing with imbalanced data are *Cost Matrices* and *Resampling.* The implementation of a *Cost Matrix* allows classifiers to penalize misclassifying the minority class more, which has been shown to improve performance. *Resampling* either removes some majority records from the dataset (*Undersampling*) or adds minority class records (*Oversampling*). Both resampling techniques make the data more balanced, but also destroy the quality of the data.

The algorithm proposed in this paper uses both *Cost Matrices* and *Undersampling* in an attempt to improve classification performance. As well as this, the algorithm optimizes several of the *Random Forest's* parameters. The genome can be seen below:

$$\langle C_{12}, ..., C_{N(N-1)}, T, K, U\rangle \tag{1}$$

Where C_{ij} are cost matrix parameters, T is the number of trees in the *Random Forest* ensemble, K is the number of attributes considered at each node by members of the *Random Forest* ensemble and U is an undersampling rate

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764712

parameter. U controls exactly how much of the majority class is removed in the training data.

Each value in the genome is represented as a double, however T and K are required to be integers, so the *floor* operator will be used for integer conversion.

F-Measure is used as a fitness function as takes both *Sensitivity* and *Precision* into account. *Sensitivity* corresponds to the number of minority class records correctly classified, and *Precision* corresponds to the inverse of the false positive rate, ie. a low *Precision* implies a high false positive rate. *Accuracy* is not used as a metric as is gives a poor idea of performance when the dataset is imbalanced. Consider a dataset with 100 records, 95 belonging to class A and 5 belonging to class B. An algorithm could classify all examples as A, and despite no members of class B being correctly classified, the accuracy would be 95%.

The algorithm in this paper used both one point crossover and random mutation as genetic operators. Individuals were selected using tournament selection with a tournament size of 2.

3. INVESTIGATION

3.1 Experimental Setup

The proposed solution is tested on five publicly available datasets from the KEEL repository [1]. A breakdown can be seen in table 1. IR corresponds to the imbalance ratio

Table 1: Sensitivity Results Breakdown

Dataset	#Ins	#Features	#Classes	IR
abalone19	4174	8	2	128.44
coil2000	9822	85	2	15.76
magic	19020	10	2	1.84
page-blocks0	5472	10	2	8.78
thyroid	7200	21	3	40.16

Four classifiers are investigated, *CART* [3], *Random Forests* [2], Bayesian Networks and the proposed algorithm. All experiments use a 60/40 training/test set split. Sensitivity and precision are recorded and discussed.

Genetic algorithms were run with a population size of 30, a generation size of 30, a crossover rate of 0.8 and a mutation rate of 0.2.

All experiments were implemented using the *Weka Data Mining Software* [4], and all classifier parameters are default.

3.2 Experimental Results

The breakdown of the sensitivity results can be seen in table 3 and the breakdown of the precision results can be seen in table 2. Results shown are the mean of 30 runs. The genetic algorithm shows an improvement over the unoptimized classifiers when comparing sensitivity, the genetic algorithm that uses undersampling appears to perform better on datasets with a large amount of class imbalance (Seagate Datasets, *Abalone*). The proposed algorithm still shows improvement over the unoptimized algorithms on the more balanced datasets. When comparing precision, the best performing algorithm is the unoptimized Random Forest. It appears the proposed algorithm takes advantage of the tradeoff between sensitivity and precision. Despite this, the proposed algorithm's precision is still competitive.

Table 2: Sensitivity Results Breakdown

Dataset	BN	CART	Proposed	RF
abalone	0.000	0.000	**0.092**	0.000
coil	0.094	0.000	**0.365**	0.056
magic	0.654	0.715	**0.785**	0.755
blocks	0.818	0.840	**0.885**	0.856
thyroid	0.932	**0.988**	0.981	0.983

Table 3: Precision Results Breakdown

Dataset	BN	CART	Proposed	RF
abalone	0.000	0.000	**0.030**	0.000
coil	0.261	0.000	0.135	**0.175**
magic	0.807	0.835	0.824	**0.881**
blocks	0.726	0.859	0.856	**0.901**
thyroid	0.924	0.941	0.945	**0.947**

These results show the proposed algorithm outperforming other classifiers in terms of the sensitivity metric, at the cost of some precision. This implies the proposed algorithm will correctly classify more of the minority class, but the majority class performance will suffer slightly.

4. CONCLUSION

In this paper an algorithm has been proposed that has been shown to improve the sensitivity of imbalanced classification. The algorithm was tested against 3 other classification algorithms on 5 datasets and in 4/5 of the cases outperformed all other classifiers in terms of sensitivity. The proposed algorithm also managed to classify the *abalone* dataset with non-zero sensitivity/precision metrics, which is something no other classifier achieved. Future work could focus on attempting to find a point in the sensitivity/precision tradeoff that is less harmful to the precision metric, this could potentially be done by using a multiobjective genetic algorithm with pareto-front.

5. REFERENCES

[1] J Alcalá, A Fernández, J Luengo, J Derrac, S García, L Sánchez, and F Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287, 2010.

[2] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees.* CRC press, 1984.

[4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[5] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.