Information Sciences 259 (2014) 571–595

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An empirical study of the classification performance of learners on imbalanced and noisy software quality data

Chris Seiffert, Taghi M. Khoshgoftaar *, Jason Van Hulse, Andres Folleco

Florida Atlantic University, Boca Raton, FL 33431, USA

ARTICLE INFO

Article history:
Available online 9 January 2011

Keywords:
Imbalance
Class noise
Sampling
Binary classification

ABSTRACT

Data mining techniques are commonly used to construct models for identifying software modules that are most likely to contain faults. In doing so, an organization's limited resources can be intelligently allocated with the goal of detecting and correcting the greatest number of faults. However, there are two characteristics of software quality datasets that can negatively impact the effectiveness of these models: class imbalance and class noise. Software quality datasets are, by their nature, imbalanced. That is, most of a software system's faults can be found in a small percentage of software modules. Therefore, the number of fault-prone, *fp*, examples (program modules) in a software project dataset is much smaller than the number of not fault-prone, *nfp*, examples. Data sampling techniques attempt to alleviate the problem of class imbalance by altering a training dataset's distribution. A program module contains class noise if it is incorrectly labeled. While several studies have been performed to evaluate data sampling methods, the impact of class noise on these techniques has not been adequately addressed. This work presents a systematic set of experiments designed to investigate the impact of both class noise and class imbalance on classification models constructed to identify fault-prone program modules. We analyze the impact of class noise and class imbalance on 11 different learning algorithms (learners) as well as 7 different data sampling techniques. We identify which learners and which data sampling techniques are most robust when confronted with noisy and imbalanced data.

1. Introduction

One of the main goals of any software engineering project is to deliver a product which is of the highest possible quality given limited time and resources. Fault detection and correction is an essential step in delivering a high quality software product, but this process can be very expensive if resources are not intelligently allocated. Data mining techniques can be used to identify software modules that are most likely to contain faults [25,23]. By distinguishing between *fault-prone* (*fp*) and *not fault-prone* (*nfp*) modules, resources can be intelligently allocated with the goal of detecting and correcting as many faults as possible. However, there are two characteristics common among software quality datasets that can hinder our ability to distinguish between *fp* and *nfp* (class variable) modules: class imbalance and class noise.

In this study, we consider only the binary classification problem. That is, we consider only two possible classes of software modules: *fp* or *nfp*. In a binary classification problem, a dataset is considered imbalanced if one class appears more frequently than the other. Class imbalance can make it significantly more difficult to detect examples (program modules) belonging to

* Corresponding author. Address: Data Mining and Machine Learning Laboratory, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA. Tel.: +1 561 297 3994; fax: +1 561 297 2800.
E-mail address: taghi@cse.fau.edu (T.M. Khoshgoftaar).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.12.016

the underrepresented class. This imbalance can be severe in the case of software quality data, since a small percentage of software modules often contain a large percentage of the faults. For example, in the dataset on which this study is based, 16% of the modules contain 90% of the faults. The underrepresented class is called the minority class, while the more frequently occurring class is called the majority class. In the domain of software quality, the minority class (*fp*) is also called the *positive* class since it is the class we wish to detect, while the majority class (*nfp*) is called the *negative* class.

A related problem is that the class of interest often has a higher misclassification cost than that of the majority class — in other words, incorrectly predicting an instance that actually belongs to the minority class as a majority class example is more costly than incorrectly predicting a majority class example as belonging to the minority class. This holds true in the domain of software quality where misclassifying an *fp* module as *nfp* is more expensive than the reverse, since this type of misclassification is likely to cause faults to go undetected.

Several *data sampling* (or resampling) techniques have been proposed to alleviate the problem of class imbalance. These techniques attempt to reduce the severity of imbalance within the data by removing examples from the majority class, or adding examples to the minority class. Some techniques do so randomly, while others attempt to "intelligently" augment or subtract from the dataset in a clever way to benefit the classifier.

Incorrect values in a dataset are called noise. Noise, especially when it occurs in the class attribute (class noise) can have a negative impact on classification performance [27,7]. An instance contains class noise if the true class of the instance is different from the recorded class. The most basic strategy for dealing with noise is to use a classification algorithm that is robust [17].

In this study, we examine the impact of noise and imbalance on a variety of classification algorithms including decision trees, nearest neighbors, neural networks and Bayesian learners. The objective of this work is to analyze the relationship between classification performance, data sampling, learner selection, class imbalance and class noise. From a real-world software quality dataset, we derive 12 datasets with different levels of noise and imbalance. We apply 11 classification algorithms to these datasets combined with seven different sampling techniques, varying sampling technique parameters when applicable. We examine the interaction between the choice of classifier and sampling technique on each of the 12 derived datasets. Each classifier/sampling technique combination was evaluated using 10 runs of 10-fold cross validation. In total, 1,267,200 classifiers were built to produce the results presented in this work.

In this work, we address the following research questions:

- What is the relative impact of class noise vs. class imbalance? Which has a more severe impact on the performance of the different classification algorithms and sampling techniques?
- How do different classification algorithms react to the application of different sampling techniques? Are some classifiers more significantly improved by the use of sampling techniques? Do certain sampling techniques work better when used in conjunction with specific classification algorithms?
- What benefit do sampling techniques provide at different levels of class imbalance and noise? When the data is highly imbalanced or very noisy, do certain sampling techniques perform better than others?
- How do classification algorithms perform at different levels of class imbalance and noise after sampling techniques have been applied to the data? When the data is highly imbalanced or very noisy, do certain classification algorithms perform better than others?

The results of our experiments show two sampling techniques, Wilson's editing and random undersampling, perform very well. Some learners are impacted by the use of sampling, while others are unaffected. Generally speaking, class noise is also shown to have a more significant impact on learners than imbalance. Section 4 presents much more detailed analysis, along with the supporting results from our experimentation.

2. Related work

Sampling techniques have received significant attention in recent research, however most of this research does not take the quality of data into consideration. Drummond and Holte [13] found that majority undersampling is more effective at dealing with the imbalance problem using C4.5, but Maloof's research [30] shows that undersampling and oversampling produce roughly equivalent classifiers using Naive Bayes and C5.0 (the commercial successor to C4.5). Weiss and Provost [39] examine the impact of class distribution, finding that the ideal class distribution is dependent on domain, though in general the natural distribution yields the best overall accuracy, while a balanced distribution results in the highest AUC. Barandela et al. [3] and Han et al. [18] examine the performance of more "intelligent" data sampling techniques such as SMOTE, Borderline SMOTE, and Wilson's Editing. Weiss and Provost [39], Japkowicz and Stephan [21], Elkan [14] and Kolcz et al. [28] study the impact of sampling on different classifiers finding that some can benefit greatly by using data sampling techniques while other classifiers are relatively unaffected by sampling. Other methods have also been proposed to handle class imbalance [10].

In addition, numerous studies have been performed to analyze the impact of noise and countless techniques have been proposed to cope with it. Zhu and Wu [43] show that noise, especially class noise, can negatively impact classification performance. Three types of techniques have been proposed to alleviate the problem of noise in data. The first is to use a robust