Information Sciences 259 (2014) 555–570

Contents lists available at ScienceDirect

Information Sciences

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

# Software quality assessment using a multi-strategy classifier

Taghi M. Khoshgoftaar [a,*], Yudong Xiao [a], Kehan Gao [b]

[a] Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, United States
[b] Department of Mathematics and Computer Science, Eastern Connecticut State University, Willimantic, CT 06226, United States

ARTICLE INFO

*Article history:*
Available online 3 December 2010

*Keywords:*
Rule-based model
Case-based learning
Genetic algorithm
Multi-strategy classifier
Software quality classification

ABSTRACT

Classifying program modules as fault-prone or not fault-prone is a valuable technique for guiding the software development process, so that resources can be allocated to components most likely to have faults. The rule-based classification and the case-based learning techniques are commonly used in software quality classification problems. However, studies show that these two techniques share some complementary strengths and weaknesses. Therefore, in this paper we propose a new multi-strategy classification model, RB2CBL, which integrates a rule-based (RB) model with two case-based learning (CBL) models. RB2CBL possesses the merits of both the RB model and CBL model and restrains their drawbacks. In the RB2CBL model, the parameter optimization of the CBL models is critical and an embedded genetic algorithm optimizer is used. Two case studies were carried out to validate the proposed method. The results show that, by suitably choosing the accuracy of the RB model, the RB2CBL model outperforms the RB model alone without overfitting.



## 1. Introduction

Software reliability is an important attribute of high-assurance and mission-critical systems. Identifying which software modules, during the software development process, are likely to be faulty is a challenging but valuable technique for improving software quality. Software quality assurance efforts can be focused on those program modules that are either high-risk or likely to have a high number of faults. A variety of classification and prediction approaches have been proposed in recent years [3,4,15–18,23,27,35]. Among them, rule-based classification and case-based reasoning techniques are commonly used in software quality classification problems. However, studies [2,8,14] show that these two techniques share some complementary strengths and weaknesses. In the field of data mining and machine learning, there exists a methodology that combines two or more basic learning approaches into an algorithm that may behave more appropriately than any basic method alone. This methodology is termed as an "empirical multi-strategy learning technique" [28,29]. KBNGE [37] and RISE [8] systems are examples of this type.

In Wettschereck [37], a hybrid method that combines the nearest-hyperrectangle algorithm and the *k*-Nearest Neighbor algorithm, called KBNGE, was introduced for improved classification accuracy. Results from eleven domains showed that KBNGE achieved generalization accuracies similar to the *k*-Nearest Neighbor algorithm at improved classification speed. KBNGE is a fast and easy to use inductive learning algorithm that gives very accurate predictions in a variety of domains and represents the learned knowledge in a manner that can be easily interpreted by the user. In Domingos [8], a unification

* Corresponding author. Tel.: +1 561 297 3994; fax: +1 561 297 2800.
*E-mail addresses:* taghi@cse.fau.edu (T.M. Khoshgoftaar), yudongxiao@gmail.com (Y. Xiao), gaok@easternct.edu (K. Gao).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2010.11.028

of two widely-used empirical approaches, rule induction and instance-based learning, was described. Theoretical analysis showed this approach to be efficient. It was implemented in the RISE 3.1 system. In an extensive empirical study, RISE consistently achieves higher accuracies than state-of-the-art representatives of both its parent approaches (PEBLS and CN2), as well as a decision tree learner (*C4.5*). Lesion studies show that each of RISE's components is essential to this performance. Most significantly, in 14 of the 30 domains studied, RISE is more accurate than the best of PEBLS and CN2, showing that a significant synergy can be obtained by combining multiple empirical methods.

A multi-strategy system can consist of a global procedure which calls different algorithms as sub-procedures, or a simple algorithm that can behave and adapt properly to different situations. Many combination or unification schemes are possible. In this paper, we present a novel classification model, RB2CBL, which cascades a rule-based (RB) model with two case-based learning (CBL) models. RB2CBL integrates the advantages of both RB and CBL models and restrains their drawbacks. Next, we will simply introduce the background and characteristics of the RB model (*C4.5*) and the CBL model as well as the possibility of the collaboration between the two types of the models.

Rule-based classification methods extract classification rules from a training dataset and use them to predict the category of the dependent variable for the incoming unlabelled instances. Once the rules are obtained, RB models work fast. Among RB models, decision trees [6,13,16,35] are especially attractive in the data mining and software quality modeling fields for several reasons. First, due to their intuitive representation, the result is easy to understand by humans [6,26]. Second, decision trees do not require many parameter settings from the user and thus are especially suited for exploratory knowledge discovery. Third, decision trees can be constructed relatively quickly [10,33]. In addition, the accuracy of decision trees is comparable or superior to other classification models [24]. In fact, the *C4.5* [30] decision tree model has become a very popular machine learning method because it produces an accurate and fast classifier.

In a case-based reasoning (CBR) system [17,22], by calculating the *distance* or *similarity* between the test data and the fit (training) data points, an unlabelled data point in the test dataset is predicted to be the same class as the most similar training data points are. The CBR model uses a lazy evaluation method, i.e., it classifies a data point only when immediately needed. Accordingly the CBR model has no overhead and new training data can be added in at any time. It is very flexible and, theoretically speaking, suited for dynamically changing data.

CBR systems have been used to solve real-world problems in many fields [3,5,12,17,19,25,36]. However, CBR algorithms have several deficiencies [2]:

- They are computationally expensive because they save and compute similarities to all the training cases.
- They are intolerant of noise.
- They are intolerant of irrelevant features.
- They are sensitive to the choice of the similarity function.

For CBR systems, as learning progresses, the learner's knowledge about certain parts of the input space increases, and examples in the "well-understood" portion of the space become less useful.

In the *Probably Approximately Correct* (PAC) model [9,32,34], learning algorithms need to approximately double the number of seen examples in order to halve their error rate. However, for conservative algorithms, since the number of examples actually used for learning is proportional to the error rate, the number of new examples used by the algorithm each time it wishes to halve its error rate remains (approximately) constant. Thus, the number of examples actually used to achieve some error rate is logarithmic rather than linear.

Along this line, Aha et al. described a framework and methodology of case-based learning (CBL) [1,2] that generates class predictions using only specific instances. This approach extends the nearest neighbor algorithm, which has large storage requirements. It described how storage requirements can be significantly reduced with, at most, minor sacrifices in learning rate and classification accuracy. While the storage-reducing algorithms perform well on some real-world databases, its performance degrades rapidly with increase of the level of attribute noise in training instances. Therefore, Aha et al. extended it with a significance test to distinguish noisy instances. The extended algorithm's performance degrades gracefully with increasing noise levels and compares favorably with a noise-tolerant decision tree algorithm.

Two induction paradigms with largely complementary strengths and weaknesses are rule-based induction and the case-based learning algorithm. Rule-based induction systems often succeed in identifying small sets of highly predictive features and making effective use of statistical measures to combat noise [8]. However, they can only form axis-parallel frontiers in the instance space, and they have trouble recognizing exceptions, or in general small, low-frequency sections of the space [14]; this is known as the small disjuncts problem. Furthermore, their general-to-specific (separate and conquer) search strategy causes them to suffer from the splintering problem: as induction progresses, the amount of data left for further learning dwindles rapidly, leading to wrong decisions or insufficient specialization due to lack of adequate statistical support. On the other hand, the CBL method can form complex, non-axis-parallel frontiers, and be well-suited to handling clustered distributed datasets and exceptions, but can be very vulnerable to noise and irrelevant features. Therefore, we create a new multi-strategy classification model which attempts to overcome the limitations of the RB and CBL models while maintaining the benefits of both models.

The rest of the paper continues with Section 2 in which the proposed multi-strategy classifier is described. Sections 3 and 4 present two case studies and finally we draw our conclusion in Section 5, including suggestions for future work.