

Building Decision Tree Software Quality Classification Models Using Genetic Programming

Yi Liu and Taghi M. Khoshgoftaar

Florida Atlantic University
Boca Raton, Florida, USA
{yliu,taghi}@cse.fau.edu

Abstract. Predicting the quality of software modules prior to testing or system operations allows a focused software quality improvement endeavor. Decision trees are very attractive for classification problems, because of their comprehensibility and white box modeling features. However, optimizing the classification accuracy and the tree size is a difficult problem, and to our knowledge very few studies have addressed the issue. This paper presents an automated and simplified genetic programming (GP) based decision tree modeling technique for calibrating software quality classification models. The proposed technique is based on multi-objective optimization using strongly typed GP. Two fitness functions are used to optimize the classification accuracy and tree size of the classification models calibrated for a real-world high-assurance software system. The performances of the classification models are compared with those obtained by standard GP. It is shown that the GP-based decision tree technique yielded better classification models.

A timely quality prediction of the software system can be useful in improving the overall reliability and quality of the software product. Software quality classification (SQC) models which classify program modules as either fault-prone (*fp*) or not fault-prone (*nfp*) can be used to target the software quality improvement resources toward modules that are of low quality. With the aid of such models, software project managers can deliver a high-quality software product on time and within the allocated budget. Usually software quality estimation models are based on software product and process measurements, which have shown to be effective indicators of software quality. Therefore, a given SQC technique aims to model the underlying relationship between the available software metrics data, i.e., independent variables, and the software quality factor, i.e., dependent variable.

Among the commonly used SQC models, the decision tree (DT) based modeling approach is very attractive due to the model's white box feature. A DT-based SQC model is usually a binary tree with two types of nodes, i.e., query nodes and leaf nodes. A query node is a logical equation which returns either true or false, whereas a leaf node is a terminal which assigns a class label. Each query node in the tree can be seen as a classifier which partitions the data set into subsets.

All the leaf nodes are labelled as one of the class types, such as *fp* or *nfp*. An observation or case in the given data set will “trickle” down, according to its attribute values, from the root node of the decision tree to one of its leaf nodes.

This study focuses on calibrating GP-based decision tree models for predicting the quality of software modules as either *fp* or *nfp*. In the context of what constitutes a good DT-based model, two factors are usually considered: classification accuracy and model simplicity. Classification accuracy is often measured in terms of the misclassification error rates, whereas model simplicity of decision trees is often expressed in terms of the number of nodes. Therefore, an optimal DT is one that has low misclassification error rates and has a (relatively) few number of nodes. GP is a logical solution to the problems that require a multi-objective optimization, primarily because it is based on the process of natural evolution which involves the simultaneous optimization of several factors.

Very few studies have investigated GP-based decision tree models, and among those none has investigated the GP-based decision trees for the SQC problem. Previous works related to GP-based classification models have focused on the standard GP process, which requires that the function and terminal sets have the *closure property*. This property implies that all the functions in the function set must accept any of the values and data types defined in the terminal set as arguments. However, since each decision tree has at least two different types of nodes, i.e., query nodes and leaf nodes, the closure property requirement of standard GP does not guarantee the generation of a valid individual(s).

Strongly Typed Genetic Programming (STGP) has been used to alleviate the closure property requirement (Montana [1]), by allowing each function to define the different kinds of data types it can accept. Moreover, each function, variable, and terminal are specified by certain types in STGP. When the STGP process generates an individual or performs a genetic operation during the simulated evolution process, it considers additional criteria that are not part of standard GP. For example, in our study of calibrating SQC models, a function in the function set can only be in query nodes, while the terminal variables such as *fp* and *nfp* can only be in the leaf nodes. Therefore, in our study we use STGP to build decision trees.

In this study we investigated a simplified GP-based multi-objective optimization method for automatically building optimal SQC decision trees that have a high classification accuracy rate and a relatively small tree size. The GP-based decision trees were built to optimize two fitness functions: the average weighted cost of misclassification, and the tree size which is expressed in terms of the number of tree nodes. Moreover, the relative classification performances of the GP-based DT model and the that of standard GP were evaluated in the context of a real-world industrial software project. It was observed that the proposed approach yielded useful decision trees.

References

1. Montana, D.J.: Strongly typed genetic programming. *Evolutionary Computation* (1995) 199–230