Predicting Software Defects using Multiple Criteria Linear Programming

Xiuxiang Zhao¹, Ying Liu¹, and Shi Yong ²

¹ Research center on Fictitious Economy and Data Science, CAS, Beijing 100080, China
Email: azhao416@163.com yingliu @163.com
² College of Inform. Science & Technology, Univ. of Nebraska at Omaha, Nebraska, USA
Email: yshi@gucas.ac.cn

Abstract—Software defect is the main factor influencing software reliability. So detect the software defect in software testing is the most effective way to improve software reliability. In this thesis, we will trying to use MCLP method to predict software defect and do some comparison with some other typical classification algorithm. And the results of our experiments indicate that MCLP is a competitive classification method in the software defect prediction area.

Index Terms—software defect, Multiple Criteria Linear Programming (MCLP)

I. INTRODUCTION

With the developing of IT technology, software plays a more and more important and key role in socio-economic sphere, people therefore have a high expectation for software reliability. Generally speaking, software defect is the main factor influencing software reliability. So detect the software defect in software testing is the most effective way to improve software reliability. There are many factors which can lead to software defects in its software lift cycle, such as Software Requirements, software design, software coding and software testing and so on. However, software defects which are produced during the stages of Software Requirements, Software Design and Software Testing, will finally expressed in software source codes. Therefore, detecting software defect by software source codes is the most popular method to predict software defect.

The most usually way the researchers used to detect the software defect in software source codes is Size and Complexity Metrics software defect Detecting method. Size and complexity metrics is an abstract expression of software source codes complexity, such as line of code, cyclomatic complexity and design complexity and so on. So far, the most frequently used size and complexity metrics is McCabe and Halstead complexity metrics.

In this thesis, we will trying to use MCLP method to predict software defect and do some comparison with some other typical classification algorithm.

II. RELATED WORK

Many scholars try to predict software defects to improve software quality, and most defect predictors are based on size and complexity metrics. Generally, defect predictors based on size and complexity metrics have two different models: (1) predicting the number of defects in the system based on regression method; (2) predicting whether the system have defects or not based on classification methods.

Complexity and size metrics have been used to predict the number of defects for a long time. The most earliest researcher seems to be Akiyama’s [1], his research was based on a software developed at Japan, his study showed that the total number of software defects have linear relationship with the lines of code (LOC) and linear model of size and complexity metrics could predict the total number of defects. Another scholar Ferdinand thought that the total number of defects increases with the number of code segments which is a sequence of executable statements [2]. And some other scholars argued that the number of defects have non-linear relationship with the lines of code.

At the same time, some researchers tried to predict software defect in the system by using classification method. Tim Menzies made use of ROCKY , model tree and decision tree classification methods to predict software defect and found that some simple learners (ROCKY) performed as well as more sophisticated methods for predicting detects [3]. After sometime, T. Menzies came out three conclusions in his another paper named "Assessing Predictors of Software Defects". The three conclusions are as follows: (1) Naive Bayes out-performs J48 for defect detection; (2) When learning on more and more data, little improvement was seen after processing 300 examples; (3) accuracy is a surprisingly uninformative measure of success for a defect detector. Two detectors with the same accuracy can have widely varying PDs (probability of detection) and PFs (probability of false alarm) [4].

III. MULTIPLE CRITERIA LINEAR PROGRAMMING (MCLP) CLASSIFICATION MODEL

Multiple Criteria Linear Programming (MCLP) classification model was proposed by Y. Shi et al, and in recent years, MCLP classification model developed rapidly in the theory and applied widely in many fields. Some new models which were based on MCLP model were proposed by researchers, such as MCQP [5], RS-MCLP [6] and RMCLP [7]. At the same time, MCLP classification method have been applied in many realistic problems, In 2001, Shi et al applied MCLP method in credit card portfolio management [8]. In 2004, He et al use fuzzy linear programming method and Multiple
Criteria Non-linear Programming method to classify credit cardholder behavior [9, 10]. In 2006, Kou detected network intrusion by using Multi-group Mathematical Programming based Classifier [11]. Following is the MCLP model:

\[
\text{Min } w_a \sum_{i=1}^{n} a_i - w_b \sum_{i=1}^{n} \beta_i \\
\text{Sub :} \quad a_i x_i - a_i + \beta_i - b = 0, \forall A_i \in G_1 \]

\[
A_i x_i + a_i - \beta_i - b = 0, \forall A_i \in G_2 \\
\text{Where } A_i \text{ is given, } x \text{ and } b \text{ are unrestricted, } a_i \text{ and } \beta_i \geq 0, G_1 \text{ is the first group, } G_2 \text{ is the second group} [12].

IV. EXPERIMENTS

Software defect prediction dataset CM1, KC1 and JM1 used in this study come from NASA’s Metrics Data Program (MDP) at http://mdp.ivv.nasa.gov. This data contains static code measures coming from McCabe and Halstead features. These features were defined in the 70s in an attempt to objectively characterize code features that are associated with software quality. The dataset now are publicly download for researchers at http://promise.site.uottawa.ca/SERepository/ [13].

This paper will assess defect detectors depending on their accuracy (ACC), probability of detection (PD), and probability of false alarm (PF). Table.IV is definitions of these measures:

In this paper, we will apply several different methods for generating defect detectors, including C5.0 Decision Tree, Support Vector Machine (SVM), Neural Network (NN) and Multiple Criteria Linear Programming (MCLP). Table.I to Table.III is the results of these defect detectors:

<table>
<thead>
<tr>
<th>TABLE I. RESULT ON CM1 DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5.0</td>
</tr>
<tr>
<td>ACC</td>
</tr>
<tr>
<td>PD</td>
</tr>
<tr>
<td>PF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II. RESULT ON KC1 DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5.0</td>
</tr>
<tr>
<td>ACC</td>
</tr>
<tr>
<td>PD</td>
</tr>
<tr>
<td>PF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III. RESULT ON JM1 DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5.0</td>
</tr>
<tr>
<td>ACC</td>
</tr>
<tr>
<td>PD</td>
</tr>
<tr>
<td>PF</td>
</tr>
</tbody>
</table>

From these result tables we can see that ACC can not be taken as the only criterion to evaluate defect predictor. That’s because in software defect dataset, the scale of positive class samples (samples with defects) and negative class samples (samples without defects) is in a greatly imbalanced situation. Some are even high to 1:9 (in CMI Dataset). Therefore, we must consider PD and PF both, so that we can have a much more objective evaluation to defect predictor. After considered the evaluation criterion of ACC, PD and PF, MCLP classification method shows the most best in our experiments. It indicates that MCLP is a competitive classification method in the software defect prediction area.

V. CONCLUSIONS AND FUTURE WORK

Where do software defects come from? By mapping software source code into size and complexity metrics, we can try to solve this problem using data mining. The experiments in our paper displayed that size and complexity metrics is useful to find software defect, classification model based on size and complexity metrics can predict software defect effectively.

In the future researches we will apply MCLP classification model to other software data sets involving other software quality attributes in addition to defects. And there are some other questions we need to solve: Are there much better indicators for software defects than size and complexity metrics? How do we use our prediction models within the software development process?

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (#90718042).

REFERENCES


