Volume 9, No. 1, January-February 2018



International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SOFTWARE FAULT PREDICTION USING CASE-BASED REASONING: A COMPARATIVE ANALYSIS

Dr. Ekbal Rashid Associate Professor Deptt. Of Computer Science & Engineering Aurora's Technological and Research Institute,Uppal, Hyderabad Mr. Madhup Kumar M.Tech Research Scholar Jharkhand Rai University, Ranchi,India

Abstract: Software fault estimation is important to increase the software reliability. Therefore, increasing the software reliability tends to increase the software quality. For testing the quality of software module, I have used four established similarity functions namely Euclidean method, Canberra method, Exponential method and a Manhattan method. The selection of a particular similarity measure may affect the performance precision of a CBR-Based fault prediction. It has been observed that, all the distance functions perform nearly the equal for the same data set which indicates efficiency of indigenous tool.

Keywords: Fault prediction, CBR, Distance functions, Machine learning

1. INTRODUCTION

Nowadays software fault prediction became crucial for increasing the software reliability. More software reliability provides better software quality. Faults are defects that results in software failure and unnecessary increase the testing costs. Faults or defects in software modules are becoming biggest challenge and it needs to be resolved. Software quality assurance is major concerns in modern era. Many Software companies are accepting that with faults or defects lack quality. Therefore, company needs a methodology which can remove the faults at the early stage of software development process which reduces the testing cost and development cost as well. Various machine learning techniques have been applied for software fault prediction such as support vector machine, neural network, and genetic algorithm and many more. We have used casebased reasoning as a method for finding the errors or fault in software module and this is the novelty of our paper.

2. BACKGROUND AND RELATED WORKS

Besides the machine learning methods discussed so far, many techniques have been proposed for software fault prediction. I especially observed that most of the models reported in the survey used as previous fault data [16]. Many researchers have used AI-Based approach like Case-Based Reasoning (CBR), Genetic Algorithm (GA), Neural Network (NN), and many more. Khan et al. [6] mentioned that, when software quality was predicted, the main objective was to predict reliability and stability of the software. Becker et al. [7] was predicted performance of the software. Zhong et. al in [4] has used unsupervised Learning techniques to build a software quality estimation system. Case-based reasoning has also been used by Kadoda et. al in [1]. Myrtveit et al in [2] and Ganesan et. al in [3] have also studied CBR was applied to software quality modeling of a family of complete industrial software

systems and the accuracy is measured better than a corresponding multiple linear regression model in predicting the number of design faults. Aamodt and Plaza are given the case-based reasoning cycle [9]. Rashid et. al emphasized on the importance of software quality prediction and accuracy of case-based estimation model [5] [8] [14][15].

3. SOME IMPORTANT CATEGORIES OF MACHINE LEARNING TECHNIQUES

In this section I have discussed about different learning methods such as Artificial Neural Network (ANN), Genetic Algorithm (GA) and Case-Based Reasoning (CBR). Applying machine learning techniques always useful for improving the efficiency of CBR systems.

A. Artificial Neural Network (ANN): Artificial Neural Network can be applied for enhancing the efficiency of case-based reasoning system. In general ANN produces the test cases for software testing. [11]. Therefore, recommendations of using neural networks are very much application-dependent.

B. Genetic Algorithm (GA): Genetic algorithm is searchedbased algorithm and it is also based of biological pattern or symbol. Genetic algorithm has been developed by John Holland and his colleagues for the goals of their research [12]. The view is learning as a competition among a population of evolving candidate problem solutions. A robustness function evaluates each solution to decide whether it will supply to the subsequent generation of solutions. Then, through operations analogous to gene transfer in sexual imitation, the algorithm creates a new population of candidate solutions [13].

C. Case-Based Reasoning (CBR): Case-based reasoning (CBR) was first dignified in the year 1980s from the effort of Schank and others on memory [10].Case-Based reasoning

is one of the most popular machine learning techniques. CBR is a problem solving paradigm that is fundamentally different from other major AI approaches, in that instead of relying solely on general knowledge of a problem domain it uses specific cases [1]. As we know case-based reasoning works on past cases or experiences. The objective is to find the group of known cases that matches the new case at best and retaining the new experience by adding in it into the existing knowledge-base (case-base) for future solution. As indicated in the figure 1.



Figure 1. Working model of case-based reasoning (Adapted from Aamodt and Plaza)

I. APPLICATION OF DISTANCE FUNCTIONS

In this section, I have applied four distance functions for finding the error level or faults in a software module by using the case-based reasoning methodology. All four distance functions are weighted. As indicated in table 1.

Table 1. Distance Functions



4. METHODOLOGY

The primary data used in this paper which is collected from B.tech students of computer science and engineering from the college campus. All the students got the program in the form of assignments which has been written in high level language. Based on students data I calculated error with six parameters using case-based reasoning. All six metrics which is used in this paper given below.

- Lines of Code (LOC)
- \bullet Number of functions or procedures (NOF).
- Difficulty level of Software (DLS) (Scale 1-3 where 1=Low, 2=Average, 3=High.
- Experience of Programmer in Year (PEX)
- Development Time (DT)
- Number of variables (NOV)

VI. RESULTS

Four established distance function namely Euclidean, Manhattan, Canberra and Exponential have been used in this paper. The prediction of faults in a software module is evaluated by using the magnitude of relative error (MRE) defined as:

As I continue making the model, through this research paper one can easily identify the distance functions which are more efficient in fault prediction. (See figure 1to 9). On the other side entire module such as program have been tested and calculated the fault level by using case-based reasoning methodology. It can be seen as a **Figure** 10 and 11.

C:\WINDOWS\system32\cmd	.exe - tc	- 🗆 ×
	SOFTWARE QUALITY PREDICTION	
1.Add Record		
2.Predict 3.Show Records		
4.Quit 2		
C-last Durdisting Mathed		
1.Euclidean		
2.Manhattan 3.Canberra		
4.Exponential		

figure 1: Tools for software quality prediction

Dr. Ekbal Rashid et al, International Journal of Advanced Research in Computer Science, 9 (1), Jan-Feb 2018,582-586



Figure 2: Selection of prediction method (Euclidean Method)



Figure 3 Level of accurecy by Euclidean Method



Figure 4: Selection of prediction method (Manhattan Method)



Figure 5: Level of Accuracy by Manhattan Method

C:\WINDOWS\system32\cmd.ex	æ - tc	- 🗆 X
	SOFTWARE QUALITY PREDICTION	
1.Add Record 2.Predict 3.Show Records 4.Quit 2		
Select Prediction Method 1.Euclidean 2.Manhattan 3.Canberra 4.Exponential 3_		

Figure 6: Selection of prediction method (Canberra Method)

C: \	\WINDOWS\system32\cmd.exe - tc	_ 🗆 X			
En	Lines Of Code(1-100) & Weight(0.0-1.0) : 55 1				
En	Number Of Functions(0-45) & Weight(0.0-1.0): 2 1				
En	Difficulty Level(1/2/3) & Weight(0.0-1.0): 2 1				
En	number of Variables : 7 1				
Enter Actual Developement Time : 15					
Enter Programmer's Experience & Weight: 2 1					
t- L	+ +	!			
+-	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	< INPUT < RETRIEVED			
MRE W.R.T. Dev_Time=0.00% Record No. : 12 Error Prediction EL(Predicted from KBS)=0.0000000 E2(Input from User)=0.000000 Do you want to contine ?(y/n)n_					

Figure 7: Level of Accuracy by Canberra Method



Figure 8: Selection of prediction method (Exponential Method)



Figure 9: Level of Accuracy by Exponential method



Figure 10: Quality Level of software with no faults 100%



Figure 11: Quality level of software with 6 errors 99.38%

5. CONCLUSIONS

While applying machine learning method especially casebased reasoning is often useful for improving the results. Because it applies to all knowledge containers. In this research paper I have used four similarity measures function for comparing the results while applying the case-based reasoning method. In contrast to other studies which emphasized on NN. Except for the exponential method, all the distance function performs nearly the same. This shows the importance of distance functions while calculating the fault in a software module. Prediction of faults or errors in software using case-based reasoning by indigenous tool is the novelty of the paper and this is the need of the research scholar.

REFERENCES

- G. Kadoda, M Cartwright, L Chen, and M. Shepperd. (2000), "Experiences Using Case- Based Reasoning to Predict Software Project Effort", In Proceeding of EASE, p. 23-28, Keele, UK.
- [2] I. Myrtveit and E. Stensrud. (1999), "A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models", IEEE transactions on software Engineering, Vol 25, no. 4, pp. 510-525.
- [3] K. Ganeasn, T.M. Khoshgoftaar, and E. Allen. (2000), "Casebased Software Quality Prediction", International journal of Software Engineering and Knowledge Engineering, 10 (2), pp. 139-152.
- [4] Shi Zhong, Taghi M.Khoshgoftaar and Naeem Selvia "Unsupervised Learning for Expert-Based Software Quality Estimation". Proceeding of the Eighth IEEE International Symposium on High Assurance Systems Engineering (HASE'04).
- [5] Ekbal Rashid, Srikanta Patnaik, Vandana Bhattacherjee "A Survey in the Area of Machine Learning and Its Application for Software Quality Prediction" has been published in ACM SigSoft ISSN 0163-5948, volume 37, number 5, September 2012, http://doi.acm.org/10.1145/2347696.2347709 New York, NY, USA.
- [6] M. J. Khan, S. Shamail, M. M Awais, and T. Hussain, " Comparative study of various artificial intelligence techniques

to predict software quality" in proceedings of the 10th IEEE multitopic conference, 2006, INMIC 06, PP 173-177, Dec 2006.

- [7] S. Becker, L. Grunske, R. Mirandola, and S. Overhage, " Performance prediction of component-based systems a survey from an engineering perspective", In architechture systems with Trust-worthy components, Vol 3938 of LNCS, Springer, 2006.
- [8] Ekbal Rashid, Srikanta Patnaik, Vandana Bhattacherjee "Enhancing the accuracy of case-based estimation model through Early Prediction of Error Patterns" proceedings published by the IEEE Computer Society 10662 Los Vaqueros Circle Los Alamitos, CA, in International Symposium on Computational and Business Intelligence (ISCBI 2013), New Delhi, 24~26 Aug 2013 ISBN 978-07695-5066-4/13 IEEE, DOI 10.1109/ISCBI.2013
- [9] Aamodt, A. and E. Plaza, Case-based reasoning: foundational issues, methodical variations and system approaches. AI Communications 7(1), 1994.
- [10] Venkata U.B.Challagulla et al "A Unified Framework for Defect data analysis using the MBR technique". Proceeding

of the 18th IEEE International Conference on Tools with Artificial Intelligence(ICTAI'06).

- [11] Tom M. Mitchell, "Machine LearningSection 4.1.1; page 82, McGraw Hill Companies, Inc. (1997.
- [12] David E. Goldberg "Genetic Algorithms in search, Optimization and Machine learning" Pearson Education, Inc.
- [13] Luger, George F. Artificial Intelligence, Structures and Strategies for Complex Problem Solving, Fourth Edition, atpage 471 2002. Harlow, England: Addison-Wesley.
- [14] Ekbal. Rashid "R4 Model for Case-Based Reasoning and Its Application for Software Fault Prediction," International Journal of Software Science and Computational Intelligence (IJSSCI) 8 (2016): 3, doi:10.4018/IJSSCI.2016070102.
- [15] Ekbal Rashid "Improvisation of Case-Based Reasoning and Its Application for Software Fault Prediction" has been published in International Journal of Services Technology and Management (IJSTM).ISSN online: 1741-525X ISSN print: 1460-6720, Vol.21, No.4/5/6, pp.214 – 227, DOI: http://dx.doi.org/10.1504/IJSTM.2015.073921, Inderscience Publisher.
- [16] Catal C. Software mining and fault prediction. WIREs Data Mining Knowl Discov 2012; 2: 420-426.