A Knowledge Discovery Case Study of Software Quality Prediction Based on Classification Models：ISBSG Database

Xiao Wang[1], Lingling Zhang [1,2,*],Yuehua Zhang[2,3], Ying Liu[2],Yong Shi[2]

1 School of Management, Graduate University of Chinese Academy of Sciences, Beijing (100190), China
2 Research Center on Fictitious Economy and Data Science, CAS, Beijing (100190), China
3 School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing (100190), China

mumpop@163.com, zhangll@gucas.ac.cn, zyh4530@163.com, yingliu@gucas.ac.cn, yshi@gucas.ac.cn
*Corresponding author: Phn +86-13661125028,E-mail:zll933@163.com, zhangll@gucas.ac.cn

Keywords: software trustworthy, classification, ISBSG database

Abstract. With the development of information technology, software has played an indispensable role in modern society. However, the quality of software is influenced by many un-trustworthy factors. This paper applies C5.0, SVM and neutral network on ISBSG database to predict the quality of software. The experimental result shows that the quality level of software can be well predicted by C5.0 model. Besides, several useful conclusions have been drawn from the experimental result.

1 Introduction

With the development of information technology, software has played an indispensable role in modern society, especially in the large-scale distributed network systems including military, aerospace, financial systems. However, software may crash and bring variety of failures and faults which lead to huge loss.

In late 80's and early 90's, software security has been a main issue in the international software research community. In 1994, Trusted Software Methodology (TSM) Project, which has been participated by a number of government and commercial organizations in the United States, identified that trustworthiness was highly dependent on management decision, technology decision, and established requirement sets（Amoroso，1994）.

As the US Department of Commerce National Institute of Standards and Technology commissioned in 2002, software bugs, or errors, are so prevalent and so detrimental that they cost the US economy an estimated $59 billion annually, or about 0.6% of the gross domestic product. As we can see, software quality has been an issue of concern not only in theory but also in reality.

2 Research Background

2.1 Related Study of Software Trustworthiness

The existing research on software worthiness is not so much, but there exist many relevant studies. Since the 1980s, a great deal of research and development works have been done on the definition of trustworthiness and relevant methods to improve software trustworthiness. Before that, people had different ideas on the definitions of software trustworthiness. The evaluation guidelines for the credibility of the computer system, which was proposed by the National Computer Security Center (NCSC) of United States, only set focus on security（Department of Defense，1985）. Trusted Software Methodology (TSM) Project, which has been participated by a number of government and commercial organizations in the United States, set forth that trustworthiness was highly dependent on management decision, technology decision, and established requirement sets（Amoroso，1994）.

In fact, the implementation of information system is extremely complex, which is the combination of information technology and management. Although automated tools provide a lot of aid, the design and production of software are mainly done by people. Many researchers believe that human is a key factor to ensure software trustworthiness. Ewusi-Mensah and Przasanyskid’s research on the failure of software projects showed that the impact of management on R&D is global and the affect of technology is only partial（Ewusi-Mensah，1994）. Doherty, King and others found that in the process of large software development, organization is even more important than technical issues（Doherty，1998）. Weinberg believed that software programming is not only social activities but also individual activities. Individual programming activities include intelligence, problem-solving ability, motivation, experience and others. Social programming activities include the composition and communication of programming groups, team and project（Weinberg，1971）. Boehm proposed "software productivity improvement opportunity tree" which includes personnel and organizational factors as the primary opportunities for improving productivity. The factors are composed of personnel arrangements, equipment and management.（Weinberg，1971；Wheeler，1990）

2.2 Knowledge Discovery Method Based on Classification Models

Classification is suitable way to handle evaluation of software worthiness and there exist a loft of mature classification algorithms. Due to the fast training speed and well-established default parameter setting（Opitz，1999）, decision tree is one of the most widely used techniques for classification and prediction. Its accuracy is competitive with other learning methods, and it is very efficient（Opitz，2007）.

Based on information gain, ID3 is the most influential decision tree algorithm（Opitz，2007）. However, information gain measure is biased towards attributes with a large number of values. As a successor of ID3, C4.5 uses gain ratio to overcome this shortcoming.

C5.0 is based on the same decision tree building algorithm and includes all the features of C4.5, while introducing many new technologies, reducing the complexity and improving efficiency. Therefore, in practice C4.5 can be completely replaced by C5.0.

Neural networks and SVMs can also be used for prediction and classification. Although the training time of them may be long, they are highly accurate and extensively applied across numerous domains（Jiawei Han, 2006）.

3 Knowledge Discovery Model of Software Quality Prediction

From the current study above, a lot of attention has been paid to the importance of software and the other factors such as humans and management have been focused on. However, the existing software trustworthiness evaluation is still not mature enough in terms of evaluation modeling and methods. Because the analysis of a certain individual attribute is often separated with another. Compared with current one-sided analysis, an overall evaluation is needed by programmers and managers so as to detect software defects and predict software quality.

Because the existing researches are about one unique aspect of factors and there is still lack of an effective model which can reveal the relation between the software quality and the overall development attributes, we try to reveal the relation between software quality and its overall development features. This paper applies several classification algorithms on ISBSG database to classify and predict the quality level of software.

The remained of this paper is organized as follows. In Section 2, we introduce current research about software trustworthiness and classification models used in the experiment. In Section 3, we give an overview of ISBSG database and the criteria for choosing attributes in software development process. Section 4 presents the experimental process and result. The last section is the summary of this paper.

4 Data Description and Preprocessing: ISBSG Database

The data set used in our investigation represents software projects from the ISBSG database Release 10, which refers to the International Software Benchmarking Standards Group repository. The projects in the ISBSG repository cover a broad cross-section of the software industry. The data can be used for estimating, benchmarking, trend awareness, comparison of platforms/languages/tools or assisting software process improvement program.Release 10 (released in January 2007) consists of 4,017 projects from over twenty-five countries around the world, with 60% of the projects being less than 7 years old.

In order to make our analysis meaningful we had to remove projects according to the following criteria:

• Remove projects whose total defeccts delivered are missing. This would mean that it is impossible to distinguish whether the software project is successful or not.

• As in [11]（Mendes，2005）, remove projects if they were not assigned a high data quality rating (A or B) by ISBSG.

Set class label for each project. The software project is classified into “high quality” as long as one of the following conditions is satisfied：the extreme defects exist , the number of major defects is more than 1 or the number of minor defects is more than 10 , and the others are classified into “low quality”. We analyzed 746 projects: 508 “high quality” and 238 “low quality”.

The ISBSG repository provides 106 variables. This subset was further reduced based on the exclusion criteria shown below:

• Variables that had more than 60% of their values missing were excluded.

• Variables that were obtained in different methods were excluded, in case the different evaluation standards caused deviation.

The data needs to be processed further in view of the following principles:

• Set dummy variables (0/1) for the categorical variables. Specially, the dummy variables set for variables that contained many categories are based on the combined categories rather than actual categories, in order to express relevant situation effectively.

• For continuous variables, replace the missing values with mean if there is not big deviation between values; otherwise, replace the missing ones with median. For category variable, classify the variables into most classes.

• Transform development time into time period measured in years.

At last, 53 variables are left. The basic descriptions of these variables are presented in Table1.

Table 1 Statistical description of 53 variables left

	Location			Variability			
	Mean	**Median**	**Mode**	**Std Deviation**	**Variance**	**Range**	**Interquartile Range**
Data Quality Rating	0.624665	1	1	0.48453	0.23477	1	1
UFP rating	0.675603	1	1	0.46846	0.21946	1	1
	0.260054	0	0	0.43896	0.19268	1	1
	0.064343	0	0	0.24553	0.06028	1	0
Count Approach	0.75067	1	1	0.43292	0.18742	1	0
Normalised Level 1 PDR (ufp)	16.22589	13.05	16.22589	26.03172	677.65058	330.4	10.62589
Normalised PDR (ufp)	17.36753	12.1	17.36753	27.03379	730.826	330.4	11.36753
Pre 2002 PDR (afp)	13.58643	10.35	13.58643	19.86334	394.55214	327.3	8.5
Project Elapsed Time	9.785286	7	9.785286	17.80841	317.13936	320.9	6
Effort Build	3073.276	3073.276	3073.276	4030	16239143	63848	1733
Effort Test	1368.697	1368.697	1368.697	2123	4506038	37332	778.69653
Effort unphased	2659.172	499.5	0	6104	37262703	61398	2239
Development Type	0.568365	1	1	0.49564	0.24566	1	1
Organisation type	0.150134	0	0	0.35744	0.12777	1	0
	0.849866	1	1	0.35744	0.12777	1	0
Business Area Type	0.857909	1	1	0.34938	0.12206	1	0
	0.142091	0	0	0.34938	0.12206	1	0
Application Type	0.351206	0	0	0.47767	0.22817	1	1
	0.648794	1	1	0.47767	0.22817	1	1
Architecture	0.215818	0	0	0.41166	0.16947	1	0
	0.117962	0	0	0.32278	0.10419	1	0
	0.66622	1	1	0.47188	0.22267	1	1
Client Server?	0.741287	1	1	0.43822	0.19204	1	1
Functional Sizing Technique	0.687668	1	1	0.46376	0.21507	1	1
Development Platform	0.257373	0	0	0.43748	0.19139	1	1
	0.190349	0	0	0.39284	0.15432	1	0
	0.072386	0	0	0.2593	0.06724	1	0
	0.479893	0	0	0.49993	0.24993	1	1
Language Type	0.63807	1	1	0.48088	0.23125	1	1
	0.296247	0	0	0.45691	0.20876	1	1
	0.20876	0	0	0.24789	0.06145	1	0
Primary Programming Language	0.158177	0	0	0.36515	0.13334	1	0
	0.209115	0	0	0.40695	0.16561	1	0
	0.146113	0	0	0.35346	0.12493	1	0
	0.486595	0	0	0.50016	0.25016	1	1
1st Hardware	0.037534	0	0	0.19019	0.03617	1	0
1st Operating	0.245308	0	0	0.43056	0.18538	1	0

System							
1st Data Base System	0.071046	0	0	0.25707	0.06609	1	0
	0.100536	0	0	0.30092	0.09055	1	0
	0.198391	0	0	0.39906	0.15925	1	0
	0.630027	1	1	0.48312	0.23341	1	1
CASE Tool Used	0.776139	1	1	0.41711	0.17398	1	0
Used Methodology	0.906166	1	1	0.29179	0.08514	1	0
How Methodology Acquired	0.743968	1	1	0.43673	0.19074	1	1
	0.087131	0	0	0.28222	0.07965	1	0
	0.168901	0	0	0.37492	0.14056	1	0
Intended Market	0.115282	0	0	0.31958	0.10213	1	0
	0.703753	1	1	0.45691	0.20876	1	1
	0.095174	0	0	0.29365	0.08623	1	0
	0.085791	0	0	0.28024	0.07854	1	0
Resource Level	1.879357	1	1	1.3377	1.78945	3	3
Max Team Size	8.943767	7	7	13.56192	183.92563	308	3
2010-year	9.55764	10	6	3.86136	14.9101	15	6

5 Model Application in Software Quality Prediction

After preprocessing the original database, we finally get 746 records, including 238 first-class records and 508 second-class ones. According to prior experience, unbalanced data has a great impact on the classification accuracy. So we adapt under-sampling method to choose training dataset and testing dataset so as to eliminate the influence.

We introduce C5.0 model to analyze and predict the relation between software quality and its development attributes. As a comparison, neutral network and SVM are also used for classification. The classification rates of these models are shown in Table 2.

Table2. Classification rates of C5.0, SVM and neutral network on ISBSG

	accuracy
C5.0	77.88%
SVM	69.17%
Neutral Network	70.43%

From the result, we can observe that the accuracy of C5.0 is the highest of the three models: 77.88%. And neutral network performs a little better than SVM on ISBSG database. This means that when we deal with software projects, C5.0 is the most effective and we should adapt this method to classify software quality. On the basis of our research, it is reasonable to choose this model while facing relevant problems.

After getting the classification model of ISBSG database, we can analyze the quality of software depending on the known model and predict software trustworthiness. This provides a useful tool for managers to distinguish whether a certain software project would be successful or not before it is implemented. Undoubtedly, it can significantly reduce the potential cost of project failure.

Besides, the result will also be influenced by parameters. One of them is the threshold value we set to classify the quality level of projects. Result may vary when the value changes. Robustness test has been carried out to eliminate the effect to the greatest extent.

However, the overall accuracy is not satisfactory to a certain extent. It may be due to the specific feature of the dataset. We will try the other classifiers on this dataset in our future research to testify our assumption.

6 Conclusions

In this paper, we carried out a case study that applied C5.0, SVM and neutral network model to predict the quality of software. The data we used is ISBSG database, which contains 4,017 records and 106 attributes. After preprocessing, 746 records and 53 attributes remained.

On the one hand, based on the experimental result, the quality level of software was well predicted and C5.0 model performed best on testing dataset compared with SVM and neutral networks.

On the other hand, the model obtained by C5.0 model can be used to predict the quality of software as a

reference for project managers.

Acknowledgment

This work was partially supported by the President Fund of Graduate University of Chinese Academy of Sciences（GUCAS)（A）(Grant No.085102HN00), National Natural Science Foundation of China (Grant No.71071151,70921061).

References

[1] E. Amoroso, C. Taylor, J. Watson, J. Weiss, 1994, *A process-oriented methodology for assessing and improving software trustworthiness,*Proceedings of the 2nd ACM Conference on Computer and communications security, Virginia, USA

[2] Department of Defense, National Computer Security Center,1985, *Trusted Computer System Evaluation Criteria*, DOD 5200.28 STD

[3] Ewusi-Mensah K, Przasanyski Z. , 1994, “Factors contributing to the abandonment of information system development project”, *Journal Of Information Technology,* 9, 185~201

[4] Doherty N F, King M. ,1998, “The consideration of organizational issues during the system development process: an empirical analysis”, *Behavior & Information technology,* 17, 1, 41~51

[5] Weinberg, G M. , 1971, *Psychology of Computer programming*, Van Nostrand Reinhold, New York

[6] Boehm B W. , 1987, “Improving Software Productivity”, *IEEE Computer*,20, 9, 43~57

[7] Wheeler, D.J. and Chambers, D.S., 1990, *Understanding Statistical Process Control*, Addison-Wesley, BOSTON, MA

[8] D. Opitz, R. Maclin, 1999, “Popular ensemble methods: An empirical study”, *Journal of Artificial Intelligence Research,* 11, 169-198

[9] Bing Liu, 2007, *Web data ming: exploring hyperlinks contents and usage,* Springer,Verlag Berlin Heidelberg

[10]Jiawei Han, Micheline Kamber, 2006, *Data mining: concepts and techniques(second edition),* Morgan Kaufmann, San Francisco, CA

[11] Mendes Emilia, Lokan Chris, Harrison Robert, Triggs Christopher, 2005, “A Replicated Comparison of Cross-Company and Within-Company Effort Estimation Models Using the ISBSG Database”. *11th IEEE International Software Metrics Symposium (METRICS 2005)*, 36