ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

Effective Bug Detection Using Data Reduction Techniques

Javyant Devare[1], Divya Prakash[2], Chandrakant Tiwari[2], Shashi Bhushan[2]

Assistant Professor, Dept. of Computer Engineering MIT Academy of Engineering, Alandi, Pune, India[1]

BE final year Student, Dept. of Computer Engineering MIT Academy of Engineering, Alandi, Pune, India[2]

ABSTRACT: In today's world Bug issues are a major problem which concerns the software company. The companies spent huge chunks of currency and time in overcoming these problems , we extract attributes from historical bug data sets and build a predictive model for a new bug data set and which can detect all different bugs and can list them or prioritize them. These can prove to be a great help to these companies because this way it is easy for them to list and minimize these bugs. Using mining techniques we create a model by leveraging data mining techniques, mining software repositories can uncover interesting information in software repositories. Maintain these repositories, we can detect the bug efficiently.

KEYWORDS: Bug Detection, Data Mining, Bug Tracking System, Data Reduction,Entropys priorities

I. INTRODUCTION

A software bug is an fault or failure in a computer program that proves to generate an incorrect or ambiguous result, or to behave in unexpected ways. There are many feasible ways to find bugs in a software. Various Dynamic techniques, such as testing and assertions, depends on the runtime behavior of a program.The most efficient and nice static technique for terminating bugs is a formal evidence of correctness. Bug Patterns are error-free coding trails that arise from the use of erroneous design patterns, misunderstanding of language semantics, or simple and common mistakes. As developers, we many times believe that any bugs in our code must be subtle, unique and require sophisticated tools to uncover. All of the bug pattern detectors are done using BCEL ,which is an open source byte code analysis and instrumentation library. The detectors are executed using the Visitor design pattern; every detector checks every class of the analyzed library or the application. Data mining (the analysis step of the Knowledge Discovery in Databases process,an interdisciplinary subfield of computer science, is the computational process of finding patterns in huge data sets which includes methods at the intersection of machine intelligence and machine learning, statistics, and database systems. The Mining Software Repositories (MSR) analyzes the rich data in software repositories, such as version, mailing list archives, bug tracking systems, control repositories , issue tracking systems etc. to uncover eye catching and function-able information about the software systems, projects and software engineering. By using various data mining techniques, mining software repositories can provide a solution to these problems. A bug repository provides a data based platform to support many types of tasks on bugs, e.g., fault prediction bug localization and reopened bug analysis . In this paper, bug reports in a bug repository are called bug data. bug triage is very time taking method . it includes handling software bugs , which assigns a right developer to a new bug coming In the experiments,we evaluate the data reduction techniques for bug triage on the bug reports of two large open source projects, such as Eclipse. Experimental output shows that by using the instance selection technique to the data set can reduce bug reports but the accuracy of bug triage may be decreased; applying the feature selection technique can reduce words in the bug data and the accuracy can be increased.

II. RELATED WORK

In this section, we review existing work on modeling bug data, bug triage, and the quality of bug data with defect prediction.To investigate the relationships in bug data, Sandusky et al. form a bug report network to examine the dependency among bug reports. Besides studying relationships among bug reports, Hong et al. build a developer social network to examine the collaboration among developers based on the bug data in Mozilla project. This developer social network is helpful to understand the developer community and the project evolution. By mapping bug priorities to developers, Xuan et al. identify the developer prioritization in open source bug repositories. The developer

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

prioritization can distinguish developers and assist tasks in software maintenance. In our work, we address the problem of data reduction for bug triage. To our knowledge, no existing work has investigated the bug data sets for bug triage. In a related problem, defect prediction, some work has focused on the data quality of software defects. In contrast to multiple-class classification in bug triage, defect prediction is a binary class classification problem, which aims to predict whether a software artifact (e.g., a source code , a class, or a module) contains faults according to the extracted features of the artifact. In software engineering, defect prediction is a kind of work on software metrics. To improve the data quality, Khoshgoftaar et al. and Gao et al. examine the techniques on feature selection to handle imbalanced defect data. Shivaji et al. proposes a framework to examine multiple feature selection algorithms and remove noise features in classificationbased defect prediction. Besides feature selection in defect prediction, Kim et al. present how to measure the noise resistance in defect prediction and how to detect noise data. Moreover, Bishnu and Bhattacherjee process the defect data with quad tree based k-means clustering to assist defect prediction. In this paper, in contrast to the above work, we address the problem of data reduction for bug triage. Our work can be viewed as an extension of software metrics. In our work, we predict a value for a set of software artifacts while existing work in software metrics predict a value for an individual software artifact.

III. PROJECT IDEA

Develop a web based application which can be installed on the system and performs following operations: 1. Add New Bugs The application will have a testers module which will allow a tester to add new bugs to system. 2. Data Reduction The application will perform data reduction on bug data to get relevant information that will be used for bug triage. 3. Bug Triage The application will perform bug triage in form of suggestions and recommendations to developers

IV.SYSTEM ARCHITECTURE

V. OPEN SOURCE SOFTWARE DETAILS

SOFTWARE REQUIREMENTS: 1. Operating System : Linux 2. Technology : Java and J2EE 3. Web Technologies : Html, JavaScript, CSS 4. IDE : Eclipse Juno 5. Web Server : Tomcat 6. Database : My SQL 7. Java Version : J2SDK1.7

HARDWARE REQUIREMENTS: 1. Hardware : Pentium Dual Core 2. Speed : 2.80 GHz 3. RAM : 1GB 4. Hard Disk : 20 GB

VI. PROPOSED ALGORITHM

INPUT : training set T with n words and m bug reports , Reduction order FS-¿IS Final number Nf of words Final number Mi of bug reports

OUTPUT : reduced data set Tf for bug triage

1) apply FS to n words of T and calculate objective values for all the words;
2) select the top nF words of T and generate a training set Tf;
3) apply IS to Mi bug reports of Tf;
4) terminate IS when the number of bug reports is equal to or less than Mi and generate the final training set

Tf Feature Selection is a preprocessing technique for selecting a reduced set of features for large-scale data sets. Four well-performed algorithms in text data and software data Information Gain (IG) , x2 statistic(CH) Symmetrical Uncertainty attribute evaluation (SU) Relief-F Attribute selection (RF) Instance selection is a technique to reduce the number of instances by removing noisy and redundant instances . An instance selection algorithm can provide a reduced data set by removing non-representative instances . Four instance selection algorithms Iterative Case Filter (ICF) ,(max likelihood inference, stochastic perturbation parameter) Learning Vectors Quantization (LVQ)(supervised classification) Decremented Reduction Optimization Procedure (DROP) and Patterns by Ordered Projections (POP)

VII. PSEUDO CODE

step 1) apply FS to n words of T and calculate objective values for all the words;
step 2) select the top nF words of T and generate a training set Tf;
step 3) apply IS to Mi bug reports of Tf;

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

step 4) terminate IS when the number of bug reports is equal to or less than Mi and generate the final training set

VIII. CONCLUSION AND FUTURE WORK

Bug triage aims to assign an appropriate developer to fix a new bug, i.e., to determine who should fix a bug. Cubranic and Murphy first propose the problem of automatic bug triage to reduce the cost of manual bug triage. They apply text classification techniques to predict related developers. In our work, we address the problem of data reduction for bug triage. To our knowledge, no existing work has investigated the bug data sets for bug triage. In a related problem, defect prediction, some work has focused on the data quality of software defects.

REFERENCES

[1] C. C. Aggarwal and P. Zhao, Towards graphical models for text processing, Knowl. Inform. Syst., vol. 36, no. 1, pp. 1?21, 2013
[2] N. E. Fenton and S. L. Peeger, Software Metrics: A Rigorous and Practical Approach, 2nd ed. Boston, MA, USA: PWS Publishing, 1998.
[3] Eclipse. (2014). [Online]. Available: http://eclipse.org/
[4] K. Gao, T. M. Khoshgoftaar, and A. Napolitano, Impact of data sampling on stability of feature selection for software measurement data,in Proc. 23rd IEEE Int. Conf. Tools Artif. Intell., Nov. 2011, pp. 10041011.
[5] T. M. Khoshgoftaar, K. Gao, and N. Seliya, Attribute selection and imbalanced data: Problems in software defect prediction, in Proc. 22nd IEEE Int. Conf. Tools Artif. Intell., Oct. 2010, pp. 137?144
[6] Mozilla. (2014). [Online]. Available: http://mozilla.org/
[7] D. Lo, J. Li, L. Wong, and S. C. Khoo, Mining iterative generators and representative rules for software speci?cation discovery,IEEE Trans. Knowl. Data Eng., vol. 23, no. 2, pp. 282?296, Feb. 2011.
[8] S. Kim, H. Zhang, R. Wu, and L. Gong, Dealing with noise in defect prediction,? in Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng., May 2010, pp. 481?490.
[9] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.
[10] K. Gao, T. M. Khoshgoftaar, and A. Napolitano, Impact of data sampling on stability of feature selection for software measurement data,? in Proc. 23rd IEEE Int. Conf. Tools Artif. Intell., Nov. 2011, pp. 1004?1011.
[11] J. Anvik, L. Hiew, and G C. Murphy, Who should x this bug? in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361370.
[12] X. Zhu and X. Wu, Cost-constrained data acquisition for intelligent data preparation, IEEE Trans. Knowl. Data Eng., vol. 17, no. 11, pp. 1542?1556, Nov. 2005

BIOGRAPHY

Mr Jayavant Devare is a assistant professor in Computer Department of MIT AOE Pune University
Mr Shashi Bhushan kumar is a BE final year student in MIT AOE Pune University
Chandra kant Tiwari is a BE final year student in MIT AOE Pune University
Divya Prakash Singh is a BE final year student in MIT AOE Pune University