International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Survey on Bug Triage with Software Data Reduction Techniques

Jayashri Gholap[1], N. P. Karlekar[2]

[1]ME Computer (Engineering), Dept. of Computer Engineering, SIT,Lonavala, Savitribai Phule, Pune University, Pune, India

[2]Assistant Professor, Computer Engineering, Dept. of Computer Engineering, SIT, Lonavala, Savitribai Phule, Pune University, Pune, India

Abstract: ***Bug triage is an essential step in the process of bug fixing. Bug triage is the process of fixing bug whose main objective is to correctly allocate a developer to a new bug further handling. Many software companies spend their most of cost in dealing with these bugs. To decrease the time cost in manual work and to enhance the working of automatic bug triage, two techniques are applied namely text classification and binary classification. In literature various papers address the problem of data reduction for bug triage, i.e., how to reduce the scale and improve the quality of bug data. By combining the instance selection and the feature selection algorithms to simultaneously reduce the data scale and enhance the accuracy of the bug reports in the bug triage. This survey focused on various data reduction technique for bug triage. As per literature, need to develop a effective model for doing data reduction on bug data set which will reduce the scale of the data as well as increase the quality of the data., by reducing the time and cost.***

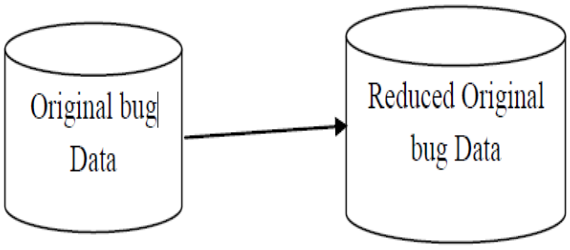
Keywords: Bug triage, data reduction, Instance selection, Feature selection, Data Mining, Mining software repositories.

1. Introduction

In modern software development, software repositories are like large-scale databases for storing the output of software development. Initial step in bug repository is to manage software bugs. Bug fixing is an important and time-consuming process in software maintenance. Open source development projects typically incorporate an open bug repository to which both software developers and users can report bugs or defects or issues in the software, suggest possible enhancements, and comment on existing bug reports. The advantage of an open bug repository is that it may allow more bugs to be identified and solved, improving the quality of the software product.

Bug triage is most vital step for bug fixing, is to allocate a new bug to a relevant developer for further handling. For open source software projects, large number of bugs are produced daily which makes the triaging process very difficult and challenging. Main objective of bug triage is to assign a developer for bug fixing. Once a developer is assigned to a new bug report he will fix the bug or try to rectify it. The motivation of this work is to reduce the large scale of the training set and to remove the noisy and redundant bug reports for bug triage.

Data reduction is the process of reducing the bug data by using two techniques namely, instance selection and feature selection which intends to get low scale as well as quality data.



2. Related Work

J. Anvik, L. Hiew, and G. C. Murphy, [2] author present a semi-automated approach intended to simplify one part of this process, the assignment of reports to a developer for further handling. Bug triage aims to allocate an appropriate developer to fix a new bug that is to determine who should fix a bug. Author first proposes the problem of automatic bug triage to reduce the cost of manual bug triage.

Presented approach is based on a supervised machine learning algorithm that is applied to information available in the bug repository. When a new report arrives, the classifier produced by the supervised machine learning technique offered a small number of developers suitable to resolve the report.

C. C. Aggarwal and P. Zhao [3], author introduced a new paradigm for text representation and processing called distance graph representations. Distance graph representations keep information about the relative ordering as well as distance among the words in the graphs and provide a much more affluent representation in terms of sentence structure of the provided data.
Knowledge discovery from text is possible with help of distance graph representation which is not possible with the use of a pure vector-space representation. Use of the distance graph representation provides significant advantages from an effectiveness perspective.

S. Kim, H. Zhang, R. Wu, and L. Gong [4], author proposes two schemes to deal with the noise present in defect data. Author introduced a method to measure noise conflict in software defect prediction and also proposed a new method called CLNI for identifying noisy instances in defect data. Noise detection and elimination algorithms are proposed to address this problem.
Proposed algorithm can identify noisy data with accuracy. In addition, after eliminating the noises using proposed algorithm, defect prediction accuracy is improved. For the machine learners that do not have strong noise resistant

ability, the noise-eliminated training sets produced by CLNI can improve the defect prediction performance.

G. Jeong, S. Kim, and T. Zimmermann [5], author proposed bug tossing graph model can be easily incorporated into existing bug triaging systems. Find out that over 37 percent of bug reports have been reassigned in manual bug triage to other developers specifically in case of Mozilla and Eclipse. Proposed the model increased the prediction accuracy as compared to traditional bug triaging approaches.

Main objective of proposed method is to reduce reassignment in bug triage.

Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis [6], bug tossing is the same as ticket routing that is transferring a problem ticket among various expert groups in search of the right resolver to the ticket, which is a well-known problem in the machine learning literature. Most approaches use various statistical models to mine workflow from activity logs.

Author design a search algorithm, called Variable-order Multiple active State search (VMS), that generates ticket transfer recommendations. Author addressed the possibility of improving ticket routing efficiency without accessing the ticket content by using Markov models. In this paper, they transferred that idea to Eclipse and Mozilla tossing events using a modified graph search algorithm. Like their work, proposed approach also reduced the length of tossing paths. Furthermore, they combined their content-less approach with a content-based approach to locate an initial developer using a traditional bug assignment algorithm.

C. Sun, D. Lo, S. C. Khoo, and J. Jiang [7], in this paper author propose a retrieval function (REP) to identify such duplicates accurately between two bug reports. Proposed approach is twofold, first BM25F is an effective textual duplicates measure that is designed for short unstructured queries and seconds a new retrieval function REP fully utilizing text and other information available in reports such as product.

A. Srisawat, T. Phienthrakul, and B. Kijsirikul, [8], paper proposed SV-kNNC approach for data reduction to enhance performance of kNN. Proposed algorithm is three fold approaches, first support vector machines (SVMs) are applied to select some important training data then weights are allocate to each training instance based on k-mean clustering and finally classify the query instances by kNN classification process.

Advantages of SV-kNNC has the ability to reduce data, the classification time required is less and provides best performance because training data are evaluated twice before classification process.

Table 1: Survey Table

Sr.no	Paper	Proposed	Advantages	Disadvantage
1	Dealing with noise in defect prediction [4].	Author introduced a method to measure noise conflict in software defect prediction and also proposed a new method called CLNI for identifying noisy instances in defect data.	Performance and accuracy is improved by using proposed approach.	The limitation of their method is that mislabeled instances are often not outliers.
2	Improving bug triage with tossing graphs [5].	Author proposed bug tossing graph model can be easily incorporated into existing bug triaging systems.	Proposed the model increased the prediction accuracy by up to 23 percentage points compared to traditional bug triaging approaches. Proposed method is to reduce reassignment in bug triage.	Current model is based on regular Markov chains and thus only use the current state for prediction
3	Efficient Ticket Routing by Resolution Sequence Mining [6].	Author design a search algorithm, called Variable-order Multiple active State search (VMS), that generates ticket transfer recommendations.	Proposed approach is robust to the size, time-variability and also reduced the length of tossing paths.	Need to extend current model to various mining techniques for better performance.
4.	Towards more accurate retrieval of duplicate bug reports [7].	Proposed approach is twofold, first BM25F is an effective textual duplicates measure that is designed for short unstructured queries and seconds a new retrieval function REP fully utilizing text and other information available in reports such as product.	Improved the accuracy of duplicate bug retrieval.	Need to speed up the retrieval process.
5.	SV-kNNC: An algorithm for improving the efficiency of k-nearest neighbor [8]	Paper proposed SV-kNNC approach for data reduction to enhance performance of kNN.	Advantages of SV-kNNC has the ability to reduce data, the classification time required is less and provides best performance because training data are evaluated twice before classification process.	Not all noisy and redundant data is removed by proposed algorithm.

3. Architectural View

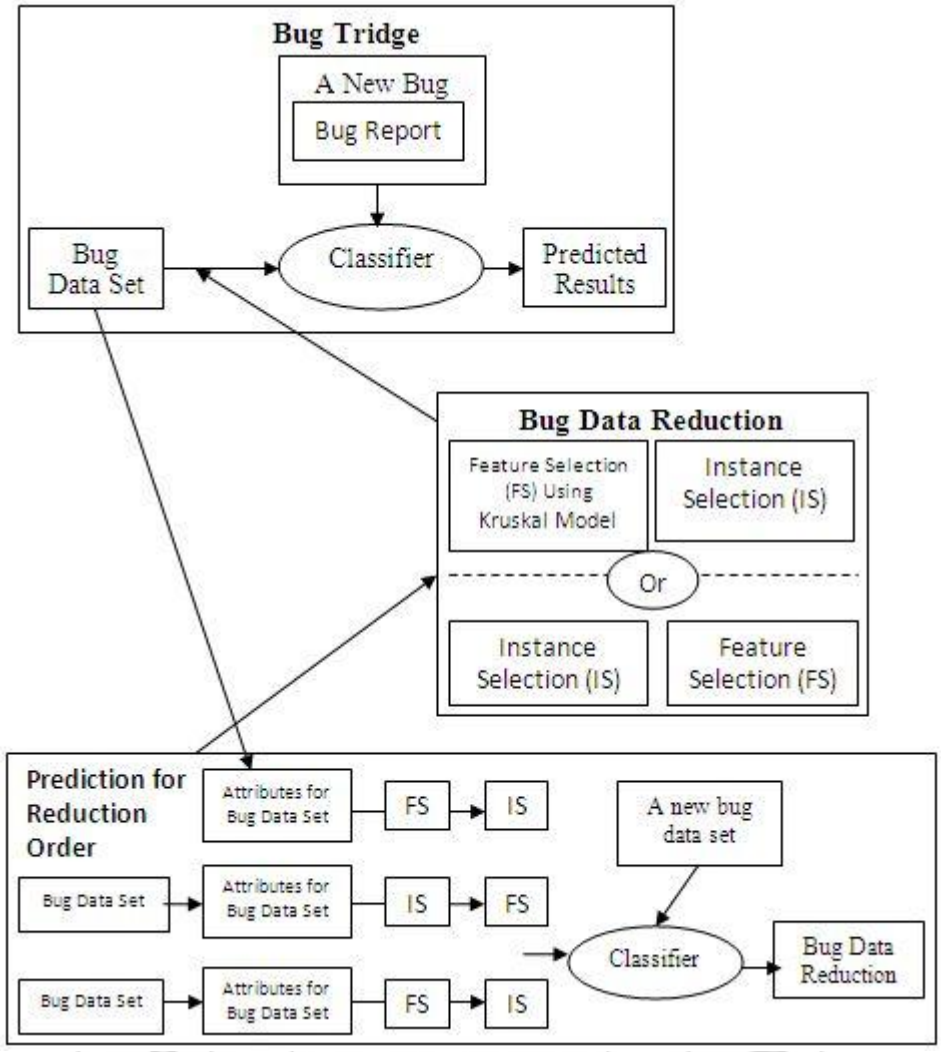


Figure 1: System Architecture

4. Conclusion

This paper presented an all-inclusive survey on the data reduction technique for bug triage. The main features, the advantages and disadvantages of each technique are described. As Bug triage is a vital step of software maintenance in both labor cost and time cost. The goal is to correctly assign a developer to a new bug for further handling. Many software companies spend their most of cost in dealing with these bugs. The motivation of this work is to reduce the large scale of the training set and to remove the noisy and redundant bug reports for bug triage.

As per survey, there is strong need to focus on reducing bug data set in order to have less scale of data and quality data. Propose the improved feature selection method by using kruskal model for addressing the problem of data reduction.

References

[1] Jifeng Xuan, He Jiang, Member, IEEE, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, Fellow, IEEE, "Towards Effective Bug Triage with Software Data Reduction Techniques", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 1, JANUARY 2015.

[2] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.

[3] C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.

[4] 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..

[5] G. Jeong, S. Kim, and T. Zimmermann, "Improving bug triage with tossing graphs," in Proc. Joint Meeting 12th Eur. Softw. Eng. Conf. 17th ACM SIGSOFT Symp. Found. Softw. Eng., Aug. 2009, pp. 111–120.

[6] Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis, "Efficient ticket routing by resolution sequence mining," in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2008, pp. 605–613.

[7] C. Sun, D. Lo, S. C. Khoo, and J. Jiang, "Towards more accurate retrieval of duplicate bug reports," in Proc. 26th IEEE/ACM Int. Conf. Automated Softw. Eng., 2011, pp. 253–262.

[8] A. Srisawat, T. Phienthrakul, and B. Kijsirikul, "SV-kNNC: An algorithm for improving the efficiency of k-nearest neighbor," in Proc. 9th Pacific Rim Int. Conf. Artif. Intell., Aug. 2006, pp. 975–979.
[9] S. Artzi, A. Kie_zun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D. Ernst, "Finding bugs in web applications using dynamic test generation and explicit-state model checking," IEEE Softw., vol. 36, no. 4, pp. 474–494, Jul./Aug. 2010.
[10] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.