International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03Issue 10
June2016

# Towards Effective Bug Triage with Software Data Decline Techniques

[1]Mr. B.KIRAN KUMAR, [2] Mr. V.N.VENU GOPAL

[1] M.Tech(CSE) from Jagruti Institute of Engineering and Technology
[2] Associate Professor, Department of Computer Science and Engineering, Jagruti Institute of Engineering and Technology, Telangana State, India.

***Abstract:*** *Bug triage is the process of fixing bug is bug triage, whose main objective is to correctly allocate a developer to a new bug further handling. To reduce time and cost of bug triaging, many software companies spend their most of cost in dealing with these bugs. we present an automatic approach to predict a developer with relevant experience to solve the new coming report. In proposed approach we are doing data reduction on bug data set which will reduce the scale of the data as well as increase the quality of the data. We are using instance selection and feature selection simultaneously with historical bug data. We have added a new module here which will describe the status of the bug like whether it assigned to any developer or not and it is rectified or not.*

***Key Words:*** *Bug triage, data reduction, Instance selection, Feature selection, Data Mining, Mining software repositories.*

## 1. INTRODUCTION

A bug repository plays an important role in managing software bugs. Many open source software projects have an open bug repository that allows both developers and users to submit defects or issues in the software, suggest possible enhancements, and comment on existing bug reports. For open source large-scale software projects, the number of daily bugs is so large which makes the triaging process very difficult and challenging [2]. Software companies spend over 45 percent of cost in fixing bugs .There are two challenges related to bug data that may affect the effective use of bug repositories in software development tasks, namely the large scale and the low quality. In a bug repository, a bug is maintained as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing [1]. Primary contribution of this paper is as follow: Here in this paper we are using feature selection and instance selection with historical data for reducing the bug data in bug repository so that we get quality data as well as low scale data. We are also adding a graph module for representing the bug report's.

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 3. System Study 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.

## 3. SYSTEM ARCHITECTURE

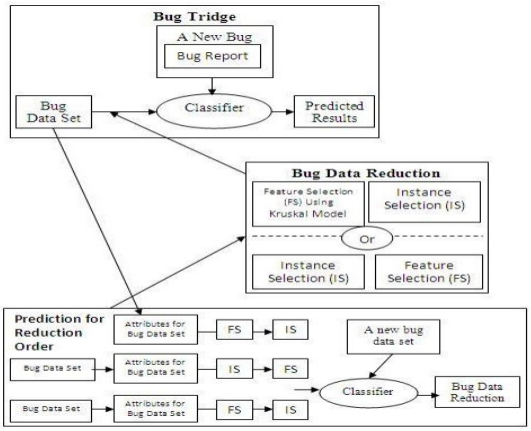


Figure 1: System Architecture

Aim 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. He will give the status related to bug whether it is rectified or not [1]. b) Data Reduction Here we are reducing the bug data by using instance and feature selection so that we get low scale as well as quality data.

## 4. INSTANCE SELECTION

- Instance selection methods associated with data mining tasks such as classification and clustering
  - it's a nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Choosing a subset of data to achieve the original purpose of a data mining application as if the whole data is used.
  - The ideal outcome of instance selection is model independent.

P Ms ≅ P Mw

**Evaluation measures:**

**Direct Measure:** Keep as much resemblance as possible between the selected data and the original data.

- Ex) Entropy, moments, and histograms.

**Indirect Measure**

- For example, a classifier can be used to check whether instance selection results in better, equal, or worse predictive accuracy.
- Conventional evaluation methods in sampling, classification, and clustering can be used in assessing the performance of instance selection.
- Ex) Precision, recall.

**3.3 Historical data**:

This is also used for reducing the bug data. Here we will enter the date for the the last accessed bug's and the data which we get we will use it for data reduction [3].

## 4. CONCLUSION

In this paper we have focused on reducing bug data set in order to have less scale of data and quality data. For that we have used feature selection and instance selection techniques of data mining as well as we have used historical data. Our experimental results showed that this data reduction technique will give quality data as well as it will reduce the data scale. We have added new module in this paper than the earlier which will give various details related to the bugs to administrator in graphical format. In future work, we plan on improving the results of data reduction more in bug triage to explore how to prepare a high quality bug data set.

## REFERENCES

[1]Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu," Towards Effective Bug Triage with Software Data Reduction Techniques" ieee transactions on knowledge and data engineering, vol. 27, no. 1, january 2015

[2] Mamdouh Alenezi and Kenneth Magel, Shadi Banitaan "Efficient Bug Triaging Using Text Mining" © 2013 academy publisher

[3]Francisco Servant "Supporting Bug Investigation using History Analysis" 978-1-4799-0215-6/13 c 2013 IEEE

[4]Pamela Bhattacharya, Iulian Neamtiu, Christian R. Shelton, "Automated, Highly- Accurate, Bug Assignment Using Machine Learning and Tossing Graphs", May 2, 2012

[5] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, Aug. 2006, pp. 43–50.

[6] P. S. Bishnu and V. Bhattacherjee, "Software fault prediction using quad tree-based k-means clustering algorithm," IEEE Trans. Knowl. Data Eng., vol. 24, no. 6, pp. 1146–1150, Jun. 2012.

[7] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," Data Mining Knowl. Discovery, vol. 6, no. 2, pp. 153–172, Apr. 2002

## ABOUT THE AUTHORS

**Mr. B.KIRAN KUMAR** is pursuing M.Tech degree in, Computer Science and Engineering from Jagruti Institute of Engineering and Technology, Telangana State, India.

**Mr. V.N.VENU GOPAL,** presently working as Associate Professor in, Department of computer science and engineering, Telangana State,India.