

Effective Bug Assortment Using Data Reduction Techniques

Swati Jain¹, Swapna Rose Wilson²

¹Assistant Professor, Department of Computer Science and Engineering, AMCEC, Bangalore, India

²M.Tech. Scholar, Department of Computer Science and Engineering, AMCEC, Bangalore, India

Abstract— Software companies spend over 48% of their total cost to fix the bugs. An effective way to automatically fix the bugs to the correct developer is called Bug Triage or Bug Assortment. Data sets containing the bug reports are collected from two large open source projects like Mozilla and Firefox. These projects consist of open source bug repositories. Bug repositories are large repositories which stores all the details of bugs. The details are stored in the form of a bug report. These bug report are saved as a document and a related developer is mapped to the label of the document. Software companies spend most of their total cost in fixing these bugs. In bug repositories the two main challenges faced is the large quantity of the data set and the low quality. Noise and redundancy are the main cause for the low quality of the data set. However, irrespective of all these difficulties assigning a proper developer to fix the bug is not an easy task without knowing the actual class of the bug. In this paper we propose data reduction technique which reduces the high scale of the data but it retains the quality of the data set. We also propose domain wise bug solution.

Keywords— Bug, Bug Assortment, Dataset, Bug Repositories.

I. INTRODUCTION

Data mining is an interdisciplinary domain in Computer Science which deals with extracting or mining the useful knowledge or information from large data sets. The other terminology for data mining is knowledge discovery as we extract the useful information from the software repositories. Mining software repositories is an interdisciplinary domain that deals to find solutions to all software engineering problems. For managing these software repositories, the bug repositories play a very important role in extraction of the data. Bug repositories contain all the bug reports and these bug reports are mapped as a document and a particular developer is mapped to the label of the document. Developers as well as users can submit their defects through large open source project like Mozilla and Firefox because they contain large bug

repositories to store all the bug reports. The regular occurring bugs are so large that it becomes too difficult to handle the particular issue.

The main objective of the paper is to obtain the bug reports from large data sets. To get the accurate results we are going to get low scale and high quality data sets by removing the bug reports and words which are redundant and which will be non informative. By using this technique we are going to increase the accuracy of the bug reports. We are also set to give the bug reports according to a particular domain, which is irrespective of the domain which a company or user may be using for his project we are going to tackle the results according to that particular domain. Bug reports are produced according to that domain using Top-k pruning algorithm which tackles each report with the help of a ranking system.

In this paper we take the datasets from two large open source projects like Mozilla and Eclipse. The experiments performed on them showed that an average of 35 new bugs are been found in each bug repositories and the bugs are been increasing day by day to tenfold the previously detected bug reports. It is a big challenge for an expert to manually fix this bugs that are so large in number because without knowing the actual class to which the bug belongs an expert cannot assign a correct developer to fix the bug. To overcome all these difficulties we designed an automated bug assortment system which by the help using text classification and reduction techniques automatically assigns a correct developer to the new bug created. In the text classification technique we use the Naïve Bayes approach. We also use an instance selection algorithm which is Iterative case filtering and a feature selection algorithm called Chi-Square. Iterative case filtering reduces the redundancy of the data sets and gives abundant parameter space likewise Chi-Square algorithm increases the accuracy by working on the sum of the squares of the errors or the bug reports created. We have modified the formula by reducing 0.5 from the sum of the difference between the expected value and the observed value.

So by using the above all techniques we are going to increase the accuracy of the data sets.

II. RELATED WORK

In the existing scenario, experts were the people who used to automatically assign a developer to fix a bug. This section provides the related work carried out, which shows the usage of different techniques that were carried out to increase the accuracy of the bug reports.

1. Automatic bug triage using text categorization.

In this work, the experts used the text classification technique which is the Naive Bayes approach which is used to classify the text based on their data sets. But as they have not used any other technique the ideologies in this paper have failed to reach the maximum accuracy that should have been met. Only 25% of the accuracy is been able to meet in the above mentioned paper.

2. A Framework for automatic assignment of bugs using vector space method.

Vector Space Model is a model which contains the history or the experience of all the developers that are been able to fix a bug. In this model which uses the vector method, the histories of the developers are fetched and the bugs were automatically assigned. But this method also failed to meet the accuracy level.

3. Improving bug triage using bug tossing graphs.

In this paper the experts have surveyed over 4,45,678 bug reports from two large open source projects. Studies shows that this method consumes a lot of time as it uses the tossing model which is used to assign the correct developer. Time consumption is the major disadvantage and also it fails to meet the accuracy level as it is of using more techniques to improve the bug reports.

4. A cost aware bug triage algorithm for bug reporting

In this paper the experts have used a cost triaging technique. The main drawback of this paper is that cost is effectively decreased by accuracy is again a question of fact. Here the total cost that is taken to fix a particular bug is taken into account. As the accuracy of the bug reports is not yet met, we cannot take the ideology that is been depicted in this paper.

5. Memories of bug fixes

In this paper, experts find the bug finding tool to find the number of occurrences the particular bug report has arrived. Studies show that 35% of the bugs occur repeatedly. To store the history of the occurrence of all these bug reports and to maintain a backup copy, we use a source code repository. By using the stored information, this paper gives

the complete knowledge about the history of the arrival of bugs to the developers.

6. Bug triage with software data reduction techniques

The above paper gives the complete information to increase the accuracy of the bug reports. Here the usage of instance selection and feature selection algorithm is been mentioned and the order of applying these algorithm is also given. We also prepare a predictive model which automatically predicts the order of applying the algorithm.

III. DATA REDUCTION FOR BUG TRIAGE

In the data reduction process, the main aim is to reduce the data sets which are redundant and non informative. Data reduction is also done based on the order in which we apply the instance and feature selection algorithm. Here the data sets are converted into a matrix form which has two dimensions. The bug and the words that will be present in the data sets are taken as the dimensions of the matrix. Instance selection algorithm is used to reduce the noisy and redundant data sets. Feature selection is used to select the reduced set of features from large data sets. Prediction model is developed to find the perfect set of features from a large set of data.

The reason why we use both instance selection and feature selection algorithm is that if we use only instance selection then we can decrease the bug reports but accuracy is decreased. If we use only the feature selection then we will get increased accuracy but words are reduced. Combination of both instance and feature selection we reduce the bug reports and words at the same time increase the accuracy.

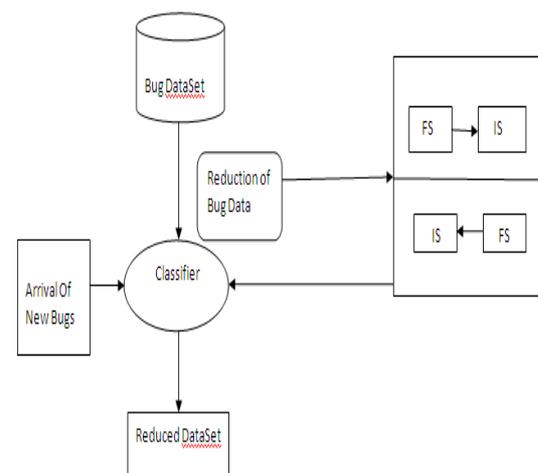


Fig. 1: Predictive Model

IV. PREDICTION FOR REDUCTION ORDERS

The prediction order is the main important concept to be noted to increase the accuracy of the bug reports. The main challenge to face here is to predict the correct order of applying the algorithms. But before applying the reduction orders we need to first check the accuracy of the data sets. Here we convert the reduction orders into binary classifiers. For building this binary classifier, extraction of 18 attributes from the data set is very important. These 18 attributes are divided into bug data set and word data sets.

V. DOMAIN SPECIFIC TASKS

In this domain specific task, the bug reports are given according to the particular domain which the user is using. Top-k pruning algorithm is used to tackle the domain specific tasks. With the goal of giving the accurate bug report, we can also get the bug reports according to the particular domain. Ranking system is used to rank the reports based on their importance and only the top k bug reports are considered and the rest is deleted. By doing so we can get low scale high quality dataset and we can also increase the parameter space.

VI. EXPERIMENTAL RESULTS

To conduct the experiments, we must prepare the dataset for data reduction. We evaluate the bug reports using two large open source projects namely Mozilla and Eclipse. Using Eclipse we can develop multi language software development. It also includes an Integrated Development Environment and also an Extensible Plug-in system. On the other hand, Mozilla is an Internet application suite containing a browser called Firefox and a email client which is Thunderbird. In this paper, the inactive developers who have fixed less than 10 bug reports are automatically removed. For a new bug created, summary and description are the main entity which describes the history of the reports.

Here the summary and the description are converted into vector space model. The vector space model has two steps tokenization and stop word removal. In tokenization, the frequency of the words is counted and non-alphabetic words are removed. Next, stop words like "the", "about", "a" are removed because these words will be of high frequency and it also does not provide any useful information. The accuracy of the bug reports is a very important criterion to get the accurate bug report. The below graph shows the exact way through which we can increase the accuracy using different instance selection and feature selection algorithm.

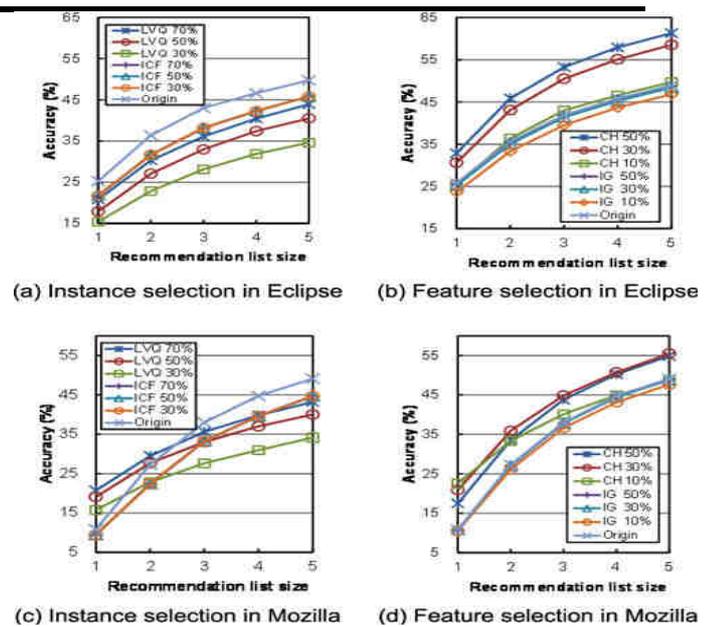


Fig. 2: Accuracy of instance and feature selection algorithms using Eclipse and Mozilla

VII. CONCLUSION

In order to increase the efficiency of the bug reports and also to assign the proper developer to fix the bug created, we use the methodology listed above. Software companies are investing 50% of their total cost in improving the bug reports created and also to assign a proper developer to fix the bugs. We have used a text classification technique through which classification of bug reports based on the size takes place. The orders in which the instance and feature selection algorithms are being applied are given through the predictive models. Bug reports are also given through specific domain that is bug reports are given out pertaining to the specific domain which the user is adapted by the help of ranking algorithm. By using the above said methods we can effectively come across the bugs and give more accurate results to the companies.

REFERENCES

- [1] Jifeng Huang, He Jiang, Yang Hu and Xingdong Hu, "Effective bug triage using software data reduction technique", *IEEE Transactions on Data Mining*, vol.27, no.1, July 2015.
- [2] D.Cubrinic and G.C Murphy, "Bug triage using text classification technique", in *Proc, 16th Int Conf, knowledge engineering*, pp.94-98.

-
- [3] G.Jeong, S.Kim, T.Zimmerman,"Bug Triage with Tossing Graph", 11th Int Conf, Software Engineering, Aug 2009, pp.23-27.
- [4] J.W Park, M.V Lee, J.Kim,"Cost aware triage for bug triage", in Proc,26th Int Conf,Aug 2011,pp 97-103.
- [5] A.E Hassan, "A Road Ahead for mining software repositories",in Proc,7th Int Conf,Sep 2012,pp 67-72.
- [6] J.Anvik, L.Heiw,"Who should fix these bugs?", in Proc, 21st Int Conf, Software Engineering, May 2007, pp.21-25.
- [7] G.Lang, Q.Li,"Matrix simplification with attribute dependency", Knowl.Info, Syst, vol.37,no.3,pp.611-638, 2014.
- [8] J.Xuan, H.Jiang,"Solving large scale release problem", iee trans, Software Engineering,vol.37,no.4,Sept./Oct.2011.
- [9] Mozilla.(2013).[Online].Available: <http://Mozilla.org/>
- [10] W.Zou, Y.Hu, J.Xuan,"Towards training set reduction",in Proc,15th Int Conf,Comput.Soft.Appl.Conf.,July.2011,pp.57-64.