A Survey on Software Data Reduction Techniques for Bug Triage Process

Snehal Chopade [1], Prof. Pournima More[2]

[1]ME , Computer Engineering .
G. H. Raisoni College of Engg & Management Wagholi, Pune

[2]Professor , Computer Engineering .
G. H. Raisoni College of Engg & Management
Wagholi, Pune

ABSTRACT

Software companies are trade with large number of software bugs. It is well exclusive and mandatory too. Bug triaging process is nothing but to assign effective and proper developer for bug fixing. In software development, the new bugs are manually triaged by expert developer i.e. human triager. Due to the large number of daily bugs and the lack of know-how, the manual bug triage is expensive in time cost and low in correctness. To avoid the expensive cost in manual bug triage, an automatic bug triage approach is used to predict developers for bug report. Data reduction technique is used to build a small scale and high quality set of bug data for bug triage. The arrangement of feature selection and instance selection techniques are used to reduce the word dimension and bug dimension of data scale. Attributes are extracted from historical bug data sets and a predictive model for a new bug data set is build to find the order of applying feature selection and instance selection.

Keywords: Mining software repositories, application of data preprocessing, data management in bug repositories, bug data reduction, feature selection, instance selection, bug triage, prediction for reduction orders.

1. INTRODUCTION

Software companies spend almost half of their project money in fixing the bugs. Large software projects have bug repository that holds all the data related to bugs and is well maintained for further processing. In bug repository, each software bug has a bug report. The bug report consists of textual data of the bug.

After the formation of bug report, a human triage assigns this bug to a developer, who will try to fix this bug. If the assigned developer cannot fix this bug, then new developer is assigned for fixing that bug. This process of assigning a correct potential developer to fix a new bug is called bug triage.

2. RELATED WORK

2.1 Literature Survey

In [1] this paper, Jeong, G. introduced a graph model supported Markov chains that captures bug moving history. This model has many desirable qualities. First, it reveals developer networks which might be accustomed discover team structures and to search out appropriate consultants for a brand new task. Second, it helps to higher assign developers to bug reports. Once a bug report has been appointed, developers will designate the bug to different developers; this method is named bug moving. one in every of the common reasons for bug moving is that bugs atypically appointed to developers by mistake.

In [2] they mentioned that Bug triaging is a fallible, tedious and time intense task. They're going with Revisiting Bug sorting and determination Practices. during this paper they studied relating to bug triaging and fixing practices, together with bug reassignments and re-openings, at intervals the context of the Mozilla Core and Firefox comes, that they envisage to be representative samples of a large-scale open supply coding system project. in addition they have decide to conduct qualitative and analysis of the bug assignment practices. They need an inclination to own AN interest in providing insights into many areas: sorting practices, review and approval processes; root cause analysis of bug reassignments and reopens in open offer coding system projects; and proposals for enhancements design of bug trailing systems

In [3] the optimizing recommendation accuracy downside is analyzed associated proposes a solution that's primarily an instance of content-based recommendation (CBR). However, cosmic microwave background is standard to cause over-specialization, recommending exclusively the classes of bugs that each developer has solved before. This downside is important in apply, as some veteran developers could also be over laden, and this is often able to slow the bug fixing

methodology. During this paper, they take a pair of directions to handle this problem: initial, a bent to develop the matter as associate optimization downside of every accuracy and worth. Second, they adopt a content-boosted cooperative filtering (CBCF), combining associate existing cosmic microwave background with a cooperative filtering recommender (CF), which boosts the recommendation quality of either approach alone.
In [4] Current techniques either use knowledge retrieval and machine learning to go looking out the foremost similar bugs already fastened and counsel knowledgeable developers, or they analyze modification knowledge stemming from ASCII document to propose knowledgeable bug solvers. Neither technique combines matter similarity with modification set analysis nor thereby do exploits the potential of the advanced between bug reports and alter Levant choices (i.e., words in bug data) among the planned system, the mixture of instance alternative and have choice is utilized. The planned systems are enforced in java language therefore it will be platform freelance. As there isn't any restriction on the size of bug's data, a tester can add sizable amount of bugs among the system. this could be one all told the largest advantages of the planned system. Since all the bug's data is receptive all the developers, it takes less time for the developer to need the selection. Developer can quickly decide on the bug to repair. Since bug sorting aims to predict the developers WHO can fix the bugs, we've a bent to follow this work to urge obviate unfixed bug reports, e.g., the new bug reports or will-not-fix bug reports. Thus, we've a bent to alone decide on bug reports, that square measure mounted and duplicate (based on the items standing of bug reports). Moreover, in bug repositories, several developers have alone fastened solely a number of bugs. Such inactive developers may not offer enough data for predicting correct developers. In our work, we've a bent to require away the developers, WHO have mounted however 10 bugs.

In [5], they counsel a semi-supervised text classification approach for bug sorting to avoid the insufficiency of labeled bug reports in existing supervised approaches which mixes naïve Thomas Bayes classifier and expectation maximization to require advantage of each labeled and unlabeled bug reports. This approach trains a classifier with a section of labeled bug reports then the approach iteratively labels several unlabeled bug reports and trains a brand new classifier with labels of all the bug reports. They additionally used a weighted recommendation list to boost the performance by discouraging the weights of multiple developers in coaching the classifier. Experimental outcome on bug reports of Eclipse demonstrate that new approach is sweet than the prevailing supervised approaches in terms of classification accuracy of bug sorting by up to six however doesn't give automatic bug sorting with a bug repository.

In [6], they examine the create use of 5 term choice strategies on the correctness of bug task to reduce time and value of bug triaging and additionally re-balance the load between developers supported their data therefore, they do experiments on four real datasets. the primary term choice technique, Log Odds magnitude relation (LOR) measures the percentages of the word occurring within the positive category normalized by the negative category. The second term choice technique, Chi- sq. (X2) take a look at is employed to look at independence of 2 events. The third term choice technique, Term Frequency connection Frequency (TFRF) is employed to pick out a lot of high frequency for instances within the positive class than within the negative class. The fourth term choice technique, Mutual data (MI) is employed to measures the shared dependence of 2 random variables. The fifth term choice technique, characteristic Feature Selector (DFS) provides total prejudiced powers of the options over the whole text set instead of being category specific. The investigational outcome demonstrates that by choosing a tiny low range of discriminating terms, the F-score will be considerably increased.

In [7], they propose the combination of both feature selection and instance selection techniques to improve the accuracy of bug triage to evaluate the training set reduction on the bug data of Eclipse. As a result, 70% words and 50% bug reports are removed after the training set reduction. The experimental results show that the new and small training sets can provide better accuracy than the original one. The drawbacks of their approach are low precision rate and cannot be directly transferred to other projects as the results are based on parts of the bug data from the Eclipse only.

2.2 Existing System

The problem of data reduction for bug sorting, i.e., the way to scale back the bug data to avoid wasting the labor price of developers and improve the standard to facilitate the method of bug sorting is addressed by Jifeng Xuan[9]. By exploitation data reduction technique for bug sorting, a small-scale and high-quality set of bug data is build by removing bug reports and words, that square measure redundant or non-informative. to cut back the bug dimension and therefore the word dimension, the mixture of feature choice and instance choice is employed. The reduced bug data contain fewer bug reports and fewer words than the first bug data and supply similar info over the first bug data. The reduced bug data is evaluated consistent with 2 criteria: the dimensions of a knowledge set and therefore the accuracy of bug sorting. to see the order of applying instance choice and have choice a prognostic model is build. The model is referred as prediction for reduction orders. First ,the attributes from historical bug data sets square measure extracted.

Then, a binary classifier is trained on bug data sets with extracted attributes and predict the order of applying instance choice and have choice for a brand new bug data set.

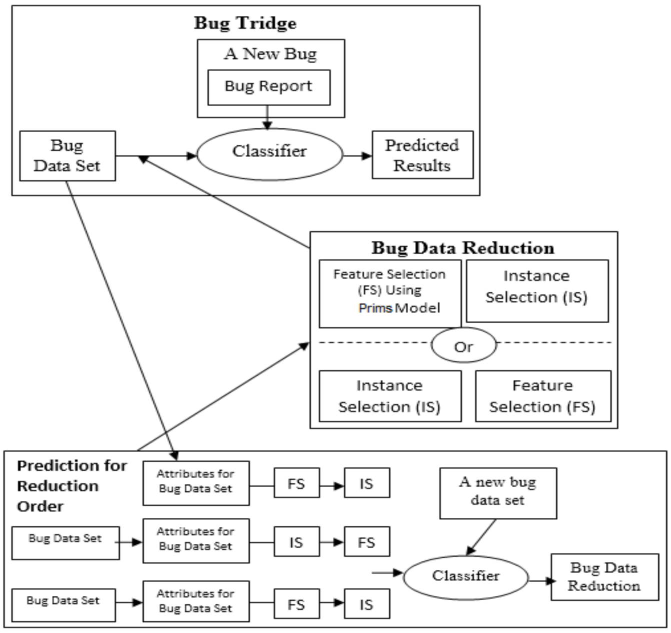

Figure 1.Existing System Architecture

3. PROPOSED SYSTEM

Basically, there are 2 forms of users within the projected system. Initial is that the developer and second is that the tester. Developer can get code bugs assigned to him. Developer will work on only 1 code bug at a time. Tester will add new bugs to the system. Within the projected system, to save lots of the labor price of developers, the data reduction for bug sorting is formed. Data reduction principally has 2 goals. Firstly, reducing the data scale and second, up the accuracy of bug sorting. Techniques of instance choice and increased feature choice rule are used for knowledge reduction. Within the projected system, the mix of instance choice and have choice is employed. Also, bug sorting in keeping with domain-specification is projected.

4.CONCLUSION

One of the expensive step in software maintenance is Bug Triaging, principally once it involves the matter of labor and time value. The recent technique aims to create reduced and high-quality bug data in software system development and thereby maintenance. the assorted data reductions techniques ar used for data reduction. The advantage of planned system is, it combines feature choice with instance choice to decrease the extent of bug data sets still as improve the data quality.

References

[1] Jeong, G., Kim, S., & Zimmermann, T. (2009, August), “Improving bug triage with bug tossing graphs” in Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (pp. 111-120). ACM.

[2] Baysal, O., Holmes, R., & Godfrey, M. W. (2012, June), “Revisiting bug triage and resolution practice” In User Evaluation for Software Engineering Researchers (USER), 2012 (pp. 29-30) IEEE.

[3] Park, Jin-woo, Mu-Woong Lee, Jinhan Kim, Seung-won Hwang, and Sunghun Kim, "CosTriage: A Cost-Aware Triage Algorithm for Bug Reporting Systems." In AAAI. 2011.

[4] Kevic, Katja, Sven Christian Muller, Thomas Fritz, and Harald C. Gall. "Collaborative bug triaging using textual similarities and change set analysis", In Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on, pp. 17-24. IEEE, 2013..

[5] Xuan, Jifeng, He Jiang, Zhilei Ren, Jun Yan, and Zhongxuan Luo "Automatic Bug Triage using Semi- Supervised Text Classification" in SEKE, pp. 209-214, 2010.

[6] Alenezi, Mamdouh, Kenneth Magel, and Shadi Banitaan. "Efficient bug triaging using text mining." Journal of Software 8.9 (2013): 2185-2190.

[7] Zou, Weiqin, Yan Hu, Jifeng Xuan, and He Jiang. "Towards training set reduction for bug triage." In Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual, pp. 576-581. IEEE, 2011.

[8] S Hu, Hao, Hongyu Zhang, Jifeng Xuan, and Weigang Sun. "Effective bug triage based on historical bug-fix information." In Software Reliability Engineering (ISSRE), 2014 IEEE 25th International Symposium on, pp. 122-132 IEEE, 2014.

[9] JifengXuan, He Jiang, Member, Yan Hu, ZhileiRen, Weiqin- Zou, ZhongxuanLuo, and Xindong Wu,“Towards Effective Bug Triage with Software Data Reduction Techniques”, in IEEE transactions on knowledge and data engineering, vol. 27, no. 1, January 2015 (General Internet site)