A SURVEY ON DATA REDUCTION TECHNIQUES OF BUG TRIAGE

Priyanka D. Patil

1Department of Computer Engineering, S.S.V.P.S.B.S. Deore College of Engineering, Dhule, Maharashtra

Abstract—The Software companies spend heavily on software bug and its corresponding issues. Bug in software degrades quality of software unless solved with great precision and accuracy. To accomplish this process called as bug triage is conducted. Companies spend heavily on bug triage process. In approach data reduction is done on bug data set which will reduce the scale of the data as well as increase the quality of the data. Instance selection and feature selection is also used simultaneously with historical bug data. The system will apply prediction on the history bug dataset through which intelligence can be added which will predict the most appropriate developer for the associated bug. Thus the System work will build a predictive model for bug dataset and reduce the involvement of skilled resources and which will in turn add to reduction of cost.

Keywords—bug triage, bug data reduction, instance selection, feature selection, prediction

I. INTRODUCTION

Many software companies spend almost half money of their project in fixing the bugs. In large software projects bug repositories holds all the information associated with bugs and also maintained for additional processing. There are two software development process modern software development and traditional software development. Modern software repositories are large scale database for storing the output of software development. Examples are source code, bugs, emails and specifications. Traditional software analysis is not completely suitable for large scale and also complex data in software repositories [1].

Each software projects has own bug repository. A bug repository is a software repository which contains all the information related to software bugs. A software bug is a problem. It causes a program and also system to crash or produce invalid output or to behave unplanned way. A repository is a vital role for managing software bugs. Software bugs are unavoidable. Fixing bug is expensive in software development. Bug repositories also called bug tracking systems. It maintenance information and support developer to handle bugs. A bug kept in bug repository as a bug report. In bug report, it records the textual report of repeating the bug and also updates the status of bug fixing [2].

A bug repository offers a data platform to providing many types of tasks on bugs such as fault prediction bug localization and reopened bug analysis. In bug reports contains bug data. After the creation of bug report, a human trigger assigns this bug to a developer, who will try to fix this bug. If the assigned developer cannot fix this bug, then other new developer is allocated for fixing that bug. This process of allocating a correct possible developer to fix a new bug is called bug triage. Therefore a bug repository shows a significant role in handling software bugs [3].

Many open source software projects have an open bug repository that permits both developers and users to submit defects or issues in the software, suggest possible enhancements, and mention on existing bug reports. In open source large-scale software projects, the number of every day bugs is so huge then it creates the triaging process very difficult and challenging [4].

There are two tasks related to bug data that may mark the effective use of bug repositories in software development namely the large scale and the low quality. Manual bug triage is time consuming and error prone process. Therefore avoiding the expensive cost of manual bug triage it proposed an automatic bug method. In automatic bug approach, a bug report is mapped to a
document and related developer is mapped to the label of document. The automatic bug approach focuses on to reduce the data to save the labor cost of developers and also improve the quality to help the process of bug triage [1].

II. LITERATURE SURVEY

Some of the techniques used in the field of bug triage:

a. **Assigning Bug Report through Recommendation**: The aim of this is to reduce the human involvement in triage. In this triage process, firstly, reports are automatically selected from a project’s issue tracking system. Then, from these selected reports, features i.e. specific piece of data are collected and reports with similar features are grouped under a label. The label shows the class or category to which the features belong. Then, these extracted data and labels are fed to a supervised machine learning algorithm. Then, recommender is created for specific development-oriented decision. Next, when recommender ask to make a prediction for new bug report, that time features are extracted from the new bug report and are fed to the recommender. Then, recommender gives a list of potential labels.

b. **Text Categorization Techniques**: Bug triage consumes more time for handling software bugs. In traditional software development, a human trigger is used i.e. expert developers were manually triaged the new bugs. But manual Bug triage is expensive in time and accuracy because of large number of daily bugs and the lack of expertise of all bugs. To reduce this expensive cost of manual bug triage, first propose the problem of automatic bug triage. Using machine learning techniques to assist in bug triage by using text categorization. Text categorization is also known as text classification which is a technique of automatically sorting a set of documents into categories from a predefined set.


H.Jiang et al [3]. They were introduced semi-automated scheme for the assignment of bug reports to a developer. In this scheme, supervised machine learning algorithm is used. A supervised machine learning algorithm takes a set of document with known label and generates a classifier. It can be used to assign a label to an unknown document. In this way works a supervised machining learning algorithm for bug assignment.

John Karsten Anvik et al [4]. It focuses on same a machine learning approach to create triage assisting recommenders. The main goal of this approach to reduce the human involvement in triage. The extracted data and labels are fed to a supervised machine learning algorithm. Recommender is created for specific development-oriented decision. Then recommender ask to make a prediction for new bug report, that time reports are extracted from the new bug report and are fed to the recommender. Then, recommender gives a list of potential labels.

Lu Zhang et al [5]. They proposed on solving this duplicated bug problem using natural language processing. This system solves the triages problem which he needs to face in examining all the bug reports. System marks the report duplicate if it finds it similar with any other previous report thus the triager does not need to examine that particular report. The system uses text mining to obtain similarity between the current report and previous reports. If found similar it marks the current new report as duplicate.

T. Zimmermann et al [6]. They were introduced bug tossing and it builds a model of bug tossing. Bug reports are tossed i.e. reassigned to other developers. It studies on assignment and reassignment of developers empirically. A tossing graph approach captures past bug tossing history to improve bug assignment and reduce redundant tossing steps.

O.Nierstrasz et al [7]. They focus on automatically suggests developers who have the suitable expertise to handle a bug report. The developers are expertise using the vocabulary which was found
in their source code contribution. Then mention developers whose support vocabulary is lexically similar to the vocabulary of the bug report.

J.Xaun et al [8]. In this paper introduced a semi-supervised approach for automatic bug triage using text classification. It is used to avoid the absence of labeled bug reports in present supervised approaches. This approach combines naive Bayes classifier and expectation maximization for the advantage of both labeled and unlabeled bug reports.

Nicolas Bettenbur et al [9]. They proposed a model which suggests the bug report creator what aspects should be used in bug report to maintain the quality of the bug report and to reduce the gap between developer requirement’s and the testing team.

S.W.Hwang et al [10]. They proposed content-based recommendation and focus on a content-boosted collaborative filtering which combines content-based recommendation with a collaborative filtering recommender. To address this challenge, they develop a topic-model to reduce the thinness and enhance the quality of content-boosted collaborative.

J.Al-Kofahi et al [11]. In this paper a bugzie a novel approach. Bugzie is based on fuzzy set-based modeling of bug-fixing expertise of developers. Bugzie uses a fuzzy set to denote the developers who are capable of fixing the bugs which are related to each term. Bugzie completes higher accuracy and efficiency than other approaches.

Sunghun Kim et al [12]. They proposed a system which uses classifiers to predict that the change under concern will create bug or not. The classifiers such as Naïve Bayes and support vector machine are used. The classifiers are trained with the history bug report and they predict the possibility of bug. Accordingly the new developer before comes to know that his change might create a problem for a system and he changes his development approach.

III. ADVANTAGES

Different Bug Traiging Techniques and its advantages are-
1. Dynamic Test Generation Technique-
   a. Minimizes condition
2. Bugzilla Approach-
   a. Dynamic bug reopening and reassignment are possible
3. Markov-chain modeling-
   a. Reduced-tossing events
   b. Model increased the prediction accuracy by up to 23%
4. Cost aware triage with content-based collaborative filtering-
   a. Reduced the sparseness and enhance the quality of content-based collaborative filtering
5. Collaborative prototypical technique-
   a. Providing context for a more informed decision in collaborative
6. Naïve-baye’s expectation maximization technique-
   a. Avoid the deficiency of labeled-report in existing supervised approach
7. Supervised-machine learning using a naïve-baye’s classifier-
   a. Classification accuracy significantly lighten the load that the tiagers face
8. Instances and Feature selection Approach-
   a. Automatic assigning of developer
   b. Low time and cost

IV. PROPOSED WORK

An unavoidable step in fixing bus is bug triage. The bug triage has a hug dataset of past bugs and their solving developers. The system focuses on the problem of data reduction for bug triage. The other problem is of feature and instance selection. The system focuses the problem of data reduction by pre-processing the dataset by removing the noisy and redundant data from the set which
reduces the scale of the bug dimensions. To address the accuracy the system builds a classifier model using naive Bayes using instance and feature selection.

The system aims to predict developer to whom the corresponding bug should be assigned based on the analysis on history dataset. The system performs preprocessing on the bug description to remove noisy, redundant and unsolved bugs to achieve better accuracy. Then a vector space model is generated of bug and its short summary.

The prediction phase is applied to the preprocessed information in order to predict the developer for the bug under consideration. The output of the system is a score of for each developer against the bug under consideration. System selects the best of the lot and gives the final predicted output. Following figure shows system architecture of reducing bug data for bug triage.

**A. System Objective:**

Proposed system focuses on the issues lacking or still need advancement in the field of bug triage.

1. Build a predictive model: The system aims at building a predictive model for predicting appropriate developer for the bug under consideration.

2. Minimize human resource in bug triage: Bug triage is an inevitable step of fixing bugs which includes large human resources in discussion and developer assignment. Proposed system will build a predictive model which will analyze history bugs dataset and predict a developer for the bug.

3. Time and cost effective solution: The systems aims to minimize human interference in the process of bug triage which results in saving human resource involvement and time. Thus system in turn ends up in being cost and time effective.

4. Data Reduction: System aims to reduce the bug data available for prediction. Bug data contains large amount of noisy and redundant data. Reducing this dataset can greatly increase accuracy and reduce the system processing time.

**Figure 1. System Architecture**
V. CONCLUSION

In this paper, I have surveyed various approaches of bug triage. Thus using proposed model the system to predict developer to whom the corresponding bug should be assigned based on the analysis on history dataset. The system can also maintain both manual labor charge and time rate.

REFERENCES


