Efficient Process of Fixing Bug using Instance and Feature Selection

A. Anbu Vidhya
Department of Computer Science and Engineering
Sri Vidya College of Engineering & Technology
Virudhunagar, India

M. Sowmya Rani
Department of Computer Science and Engineering
Sri Vidya College of Engineering & Technology
Virudhunagar, India

***Abstract*— The bug triage is a required step for handling the software bugs and the time and cost taken to reduce the bug is little high. When the bug occurs the admin stores the detail of the bug here. We use the instance and feature selection algorithm to obtain the subset of the relevant instances and to give the enhanced solution. These algorithm are used to reduce the data here and we store the historical bugs and it can be used for later usage. The result shows that our data reduction step can effectively reduce the data size and increase the accuracy of the bug triage. We use the instance and feature selection algorithms to reduce the historical bug data. The reduced bug datas have less bug reports than the original bug data and provide similar information over the original bug data. When they login to find the solution of the bug, the details of the bugs are shown automatically. Here bugs are visible to everyone and can find the solutions to the projects and provide different solutions to the bugs. We have added a new step here, which will describe the status of the bug like whether it assigned to any developer or not and it is rectified or not.**

Keywords—Bug data reduction,Bug triage,Feature selection,Instance selection

I.INTRODUCTION

Many software companies spend most of the money in fixing the bugs. Large software projects have bug repository that collects all the information related to bugs. In bug repository, each software bug has a bug report. The bug report consists of textual information regarding the bug and updates related to status of bug fixing.

result during the bug fixing process such a bug tracking system, which is used by many large open source software projects. Based on the bug tracking system, the developers can easily search and maintain all the existing bugs. Bug triage, an important step for bug fixing, is to assign a new bug to a relevant developer for further handling. A general method for bug triage is to assign bugs manually[5]. In practice, due to the frequent changes of software development teams, it is difficult to identify the correct developer in manual triage. 37 bugs per day are submitted to the bug tracking system and 3 person-hours per day are required for the manual triage.

Normally, in companies the bugs have to be properly maintained . One has to take a great care in proper maintenance and resolution of the bugs.Redundant data increases the cost of the data processing and bug triage. The bug is assigned to a particular developer to fix the bugs, so the time increases[3].Low quality bugs decreases the effectiveness of fixing bugs in software development step.

Once a bug report is formed, a human triager assigns this bug to a developer, who will try to fix this bug. This developer is recorded in an item assigned-to. The assigned to will change to another developer if the previously assigned developer cannot fix this bug. The process of assigning a correct developer for fixing the bug is called bug triage[1]. Bug triage is one of the most time consuming step in handling of bugs in software projects. Manual bug triage by a human triager is time consuming and error-prone since the number of daily bugs is large and lack of knowledge in developers about all bugs. Because of all these things, bug triage results in expensive time loss, high cost and low accuracy[2].

The information stored in bug reports has two main challenges. Firstly the large scale data and secondly low quality of data. Due to large number of daily reported bugs, the number of bug reports is scaling up in the repository[6]. Noisy and redundant bugs are degrading the quality of bug reports. Bug fixing is a significant and time-consuming process in software maintenance. For a large-scale software project, the number of daily bugs is so large that it is impossible to handle them without delaying . The work of managing bugs increases the cost of software quality maintenance[4]. Many software projects use a bug tracking system to store and manage bugs submitted by users, including end users, testers, and developers.

The bug tracking system provides a platform, where users can communicate with each

In this paper,addresses the problem of data reduction for bug triage effectively, i.e., how to reduce the bug data to save the labor cost of developers and improve the quality to facilitate the process of bug triage.The solution presented in this paper is the replacement by all the developers in the company to effectively do the bug triage process. This system uses Classification and Prediction algorithm for reducing data from the bug sets and the performance increases from the existing system.

II.RELATED WORKS

Review and Evaluation of Feature Selection Algorithms in Synthetic Problems[1]

The main purpose of Feature Subset Selection is to find a reduced subset of attributes from a data set described by a feature set. A measure to evaluate FSA is devised that computes the degree of matching between the output given

by a FSA and the known optimal solution. An extensive experimental study on synthetic problem is carried out to assess the behavior of the algorithms in terms of solution accuracy and size as a function of the relevance, irrelevance, redundancy and size of the samples.

An Efficient Greedy Method for Unsupervised Feature Selection[2]

In data mining applications, data instances are typically described by a huge number of features. Most of these features are irrelevant or redundant, which negatively affects the efficiency and effectiveness of different learning algorithms. The selection of relevant features is a crucial task which can be used to allow a better understanding of data or improve the performance

Reducing Feature To Improve Code Change Based Bug Prediction[3]

This paper investigates multiple feature selection techniques that are generally applicable to classification-based bug prediction methods. The techniques discard less important features until optimal classification performance is reached. The total number of features used for training is substantially reduced

Reducing The Effort Of Bug Report Triage[4]

It improve the software development process in a number of ways, reports added to the repository need to be triage .To assist triager with their work, this article presents a machine learning approach to create recommenders that assist with a variety of decisions.

Memories Of Bug Fixes[5]

In this paper, all the occurred and the record of the old bug and new bugs are stored so they uses the bug finding algorithm using bug fixing memory to store all the bug for later uses it gives strong suggestions and detect the bugs and the repeated bugs can be used earlier.

Reduction Techniques For Instance-Based Learning Algorithm [6]

Instance-based learning algorithms are often faced with the problem of deciding which instances to store for use during generalization. Storing too many instances can result in large memory requirements and slow execution speed it uses drop and delete algorithms used to remove instances from concept description.

III.SYSTEM DESIGN

A. System Architecture

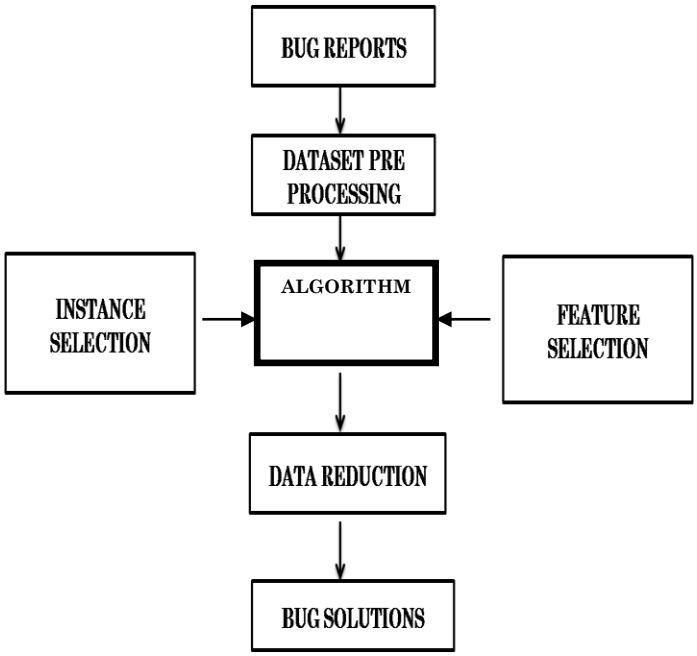


*Fig 1.*Illustration of finding solutions for bug triage. This figure represents the framework of the existing system on bug triage. Data reduction combines the instance and feature selection algorithms to reduce the bug datas. We have proposed a Naive Bayes classification for predicting the attributes.

B.Data Set Pre-Processing

Bug data records the textual description of reproducing the bug and updates according to the status of bug fixing. A bug repository provides a data platform to support many types of tasks on bugs, bug localization, and reopened bug analysis.

Bug data set provides to support information collection and assist developers to handle bugs[5]. Bug data set is prepared and stored by all the developers when they're faced complex bugs.

Instance selection and feature selection are widely used techniques in data processing[3]. For a given data set in a certain application, instance selection is to obtain a subset of relevant instances (i.e., bug reports in bug data) while feature selection aims to obtain a subset of relevant features (i.e., words in bug data). In our work, we employ the combination of instance selection and feature selection

C.Instance and Feature Selection

By applying the instance selection technique to the data set can reduce bug reports but the accuracy of bug triage may be decreased; applying the feature selection technique can reduce words in the bug data and the accuracy can be increased[1]. Meanwhile, combining both techniques can increase the accuracy, as well as reduce bug reports and words. They reduce the scale of bug data.

D. Data Reduction

The data reduction is mainly used for reducing the data scale and improving the accuracy of bug triage.

- ***Bug Dimension***

The aim of bug triage is to assign developers for bug fixing. Once a developer is assigned to a new bug report, the developer can examine historically fixed bugs to form a solution to the current bug report[4]. For example, historical bugs are checked to detect whether the new bug is the duplicate of an existing one; moreover, existing solutions to bugs can be searched and applied to the new bug. Thus, we consider reducing duplicate and noisy bug reports to decrease the number of historical bugs.

TABLE 1
Part of Details of Bug Reports

DATE	TRIAGER	STATUS
20-03-2016	A.Vidhya	New
10-02-2016	S.Siva	In Process
17-01-2016	M.Sowmya	50% Completed
23-08-2015	M.Raja	Completed

- ***Word Dimension***

By removing uninformative words, feature selection improves the accuracy of bug triage. We use feature selection to remove noisy or duplicate words in a data set[2]. Based on feature selection, the reduced data set can be handled more easily.

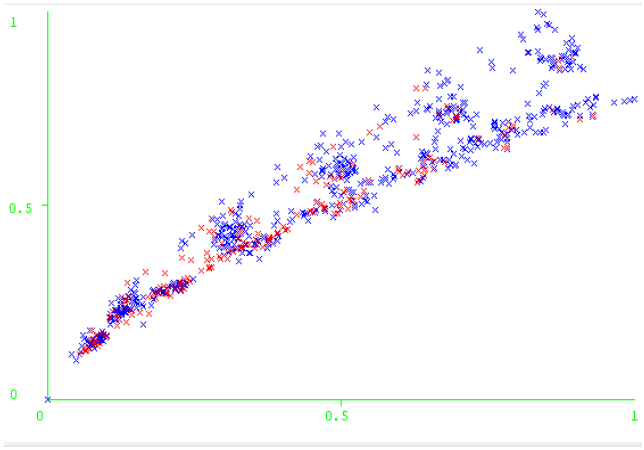
TABLE 2
Detailed Accuracy by Class of Data Sets

Projects	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
Mozilla	0.833	0.353	0.821	0.833	0.827	0.82	C0
	0.647	0.167	0.667	0.647	0.657	0.82	C1
Eclipse & Mozilla	0.594	0.262	0.848	0.594	0.698	0.709	0
	0.738	0.406	0.425	0.738	0.539	0.709	1

IV.RESULT AND DISCUSSION

In this section we provide the result of fixing bug using the classification method i.e., Naïve Bayes.

In this paper, we show that the data reduction can be used as a step in fixing bugs, which reduces the data size and increases accuracy of the bugs. In the experimental results in Table 2 ,we show the usage of Naïve Bayes to view the results of data reduction in fixing bugs for Mozilla and Eclipse & Mozilla.



In this figure 2, X axis - total instances and Y axis - total features in numerical values are taken.Its visualizing the naive bayes classification in graph format for both Eclipse and Mozilla.

It has some attributes and a plot matrix is obtained.Different plot matrix can be got by using different instance on Xaxis and Y axis.

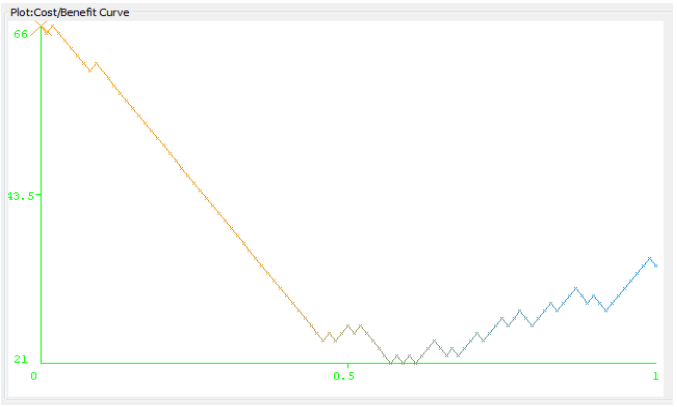


Fig 3. Cost/Benefit Curve in Mozilla

Here, In X-axis sample size in numerical values are taken and in Y axis cost/benefit values in numerical are used to draw the curve.

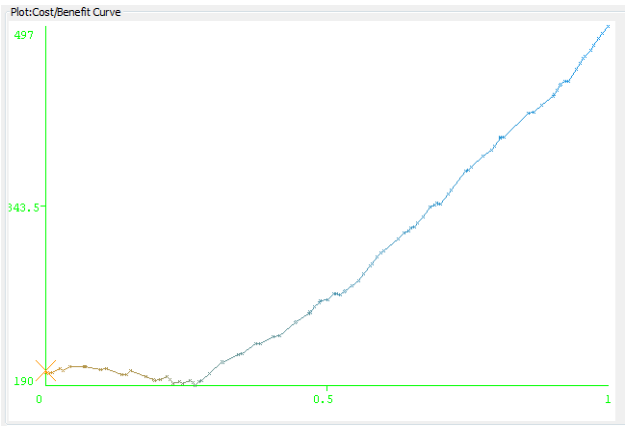


Fig 4.Cost/Benefit Curve in Eclipse & Mozilla

Here, In X-axis sample size in numerical values are taken and in Y axis cost/benefit values in numerical are used to draw the curve.

TABLE 3
Cross Validation Table for Projects

Projects	Total Instances	Correct Instances	Incorrect Instances
Mozilla	100	77	23
Eclipse & Mozilla	699	444	255

In this Table 3, the project instances are classified based on correct and incorrect instances. For different projects like Mozilla and Eclipse & Mozilla, the total instance values are classified based on this type. The root mean squared error for Mozilla is 0.3921 while for both it's 0.5651.

V.CONCLUSION AND FUTURE WORK

Bug triage is a costly step of software maintenance in both labor cost and time cost. In this paper, we combine feature selection with instance selection to reduce the scale of bug data sets as well as improve the data quality. To determine the order of applying instance selection and feature selection for a new bug data set, we extract attributes of each bug data set and train a predictive model based on historical data sets. We empirically investigate the data reduction for bug triage in bug repositories. Our work provides an approach to leveraging techniques on data processing to form reduced and high-quality bug data in software development and maintenance. In future work, we plan on improving the results of data reduction in bug triage to explore how to prepare a high quality bug data set and tackle a domain specific software task. For predicting reduction orders, we plan to pay efforts to find out the potential relationship between the attributes of bug data sets and the reduction orders.

VI. REFERENCES

[1] L.A. Belanche, F.F. González, "Review and Evaluation of Feature Selection Algorithms in Synthetic Problems ," Knowledge and Information System,March 2012
[2] Ahmed K. Farahat,Ali Ghodsi Mohamed ,S. Kamel,"An Efficient Greedy Method for Unsupervised Feature Selection", IEEE 11th International Conference on Data Mining,2011
[3] Shivkumar Shivaji , E. James Whitehead, Ram Akella , Sunghun Kim," Reducing Features to Improve Code Change Based Bug Prediction",IEEE Publications,March 2009
[4] John Anvik," Reducing the effort of bug report triage", ACM Transactions on Software Engineering and Methodology, Volume 20 Issue 3, August 2011
[5] Sunghun Kim,Kai Pan,E.James Whitehead,Jr,"Memories Of Bug Fixes",Nov 2011
[6]D.RandallWilson,Tony R.Martinez,"Reduction Techinques For Instance-Based Learning Algorithm", Machine Learning,Volume 38,Issue 3.