Research Paper Engineering

Automatic Bug Classification Using Data Reduction Techniques

Anita Kunjir Student & MMIT, Lohegaon, Pune, Maharashtra

Laxman Mulik Student & MMIT, Lohegaon, Pune, Maharashtra

Bhupesh Kumar Student & MMIT, Lohegaon, Pune, Maharashtra

ABSTRACT

Now a day software companies spend 45 present cost for software bugs. Set of bug fixing is bug triage which is main goal is assigning new bug to correct potential developer. The existing bug triage system work with text classification techniques, which build classifiers from training data sets of bug report. These approaches facing problem from the large scale and low quality bug sets. In this paper we propose feature selection and instance selection techniques for bug triage to reduction bug data sets. In this paper we are studied combination of the feature selection algorithm CH, instance selection algorithm ICF. We evaluate the training set reduction on the bug data of Mozilla. The data reduction can be reduce data scale and improving the accuracy of bug triage

KEYWORDS : Bug data reduction technique, feature selection technique, instance selection technique, bug triage, prediction for reduction orders technique.

INTRODUCTION

Data mining the mining repositories of software that uncover related information in software repositories and solve real world software related problems. A bug repository is an important for managing bugs. Manually fixing bugs is time consuming process in software maintenance. In Big software projects large number of bugs are occur so it is impossible to handle without delaying [1]. The bugs increases automatically cost of software quality maintenance will be increases. Bug tracking is to store and manage the bug by all type of users related software project and provide service as users all are communicate with each other during the process of fixing bugs.

In this paper, bug reports in bug repository known as bug data. Two challenges related to bug data that is affect the effective use of bug repositories in software development are (1) large scale and (2) low quality [4]. Number of new bugs are stored in to the bug repositories. Bug triage is time consuming process goal is bug assign to correct developer to fix new bug. Existing systems when new bugs occurred manually triaged by developer. So number of daily bugs and lack of experts all the bugs in bug triage is manual costly in time and low accuracy [5].

To avoid high cost of manual bug triage we proposed an automatic bug triage system approach which is with applying text classification and prediction techniques to predict developers for bug reports. This bug reports are mapped as document and related developer mapped as label of that document. Then bug triage is convert into problem of text classification and is automatically solve with text classification techniques like Naive Bayes. The Large scale and low quality bug data in bug repositories is techniques of automatic bug triage.

We finds problem of data reduction for bug triage as how to reduce the bug data to save the labor cost and improve the quality of bug triage process. Data reduction technique for bug triage is aim to build small scale and high quality data sets of bug data by reducing bug reports and words which are not required. We combine existing techniques of instance and feature selection to reduce the bug and word dimension. We evaluate the bug reduce data according to the scale of data set and accuracy of bug triage.

The simultaneously order of applying these two algorithms are an instance selection and feature selection may affect results of bug triage. We evaluate the data reduction technique for bug triage on bug reports of large open source software projects are namely Mozilla. An experimental results shows that when applying the instance selection technique can reduce bug reports but the accuracy of bug triage may be decreased and when applying the feature selection technique can reduce words in the bug data and the accuracy can be increased. Combines both techniques can increase the accuracy as well as reduce bug reports and words. For example when 50 percent of bug data using (instance selection) and 70 percent of words using (feature selection) removed are the accuracy of Naive Bayes on Mozilla.

LITERATURE SURVEY

Towards More Accurate Retrieval of Duplicate Bug Reports. Bug reporting however an awkward circulated process is. End clients and analyzers in the bug reporting framework.[3] This reasons an issue as distinctive designers ought not to be relegated the same imperfection. Bug reports are copy of others is normally done physically by a man called the triage. This procedure however is not versatile for frameworks with substantial client base on the procedure could take much time. Automatic bug triage using text categorization in to the feature selection (FS) and instance selection (IS) for the bug triage problem. The existing bug triage approaches are based on the text categorization. The rest work of bug triage proposed in is supervised text categorization approach using Naive Bayes. Develop this work with some other administered learning calculations proposal rundown and complex marking heuristic.

PRAPOSE WORK

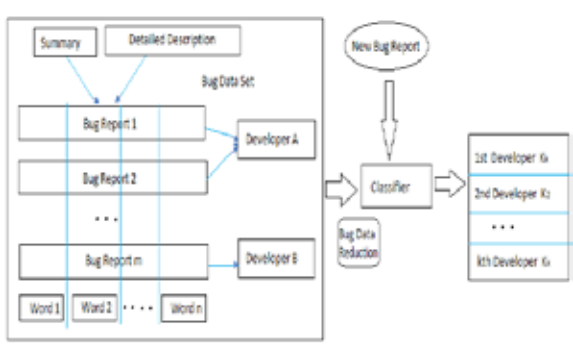

Fig1: Automatic Bug Triage System

A. Bug Triage:

The working of assigning relevant developer for fixing bug is called bug triage. In bug triage stores bug data sets also called as bug report. In fig1 show the bug report the summary and description are two items about the information of the bugs which are in natural languages. The summary denoted as general statements for identifying the bug while description gives the details of bug. Bug data sets view in text matrix there each row indicate one bug report and each column indicate one word. For low accuracy of bug triage recommendation list with the size m is used to provide list of mth developers who have top-m possibilities to fix a new bug.

Summary and the description of bug report are extract the textual contents while the developer who can fix this bug in detail existing bug reports with their developers are formed as a training sets to train a classifier is Naive Bayes typical classifier in bug triage. When enter the new bug report in system it will first apply the bug data re-

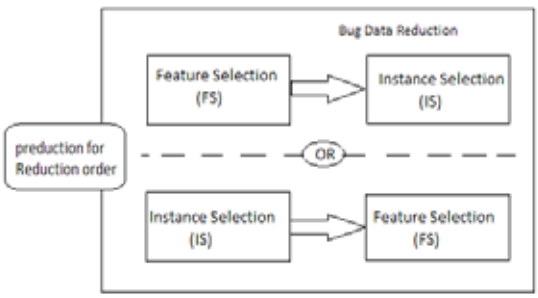

Fig 2: Bug Data Reduction

In above fig2 we combines the techniques are instance selection and feature selection to removes certain bug reports and words. A problem of reducing bug data is to determine the order of applying instance selection and feature selection which is denoted as the prediction for reduction orders.

In data reduction technique we first present how to apply instance selection and feature selection to bug data that means data reduction for bug triage and then we list the benefit of the data reduction. The details of prediction for reduction orders.

Algorithm For data reduction techniques

Algorithm 1:

Data reduction based on **FS** $\longrightarrow$ **IS**
Input: training set **T** with **n** words and **m** bug reports
Reduction order **FS** $\longrightarrow$ **IS**
Final number **nF** of words
Final number **mI** of bug reports
Output: Reduced data set **TFI** for bug triage
Steps:
1. Apply **FS** to **n** words of **T** and calculate objective values for all the words.
2. Select the top **nF** words of **T** and generate a training set **TF.**
3. Apply **IS** to **mI** bug reports of **TF.**
4. Terminate **IS** when the number of bug reports is equal to or less than **mI** and generate the final training set **TFI .**
[5]

OR

Algorithm 2:
Data reduction based on **IS** $\longrightarrow$ **FS**
Input: training set **T** with **n** words and **m** bug reports
Reduction order **IS** $\longrightarrow$ **FS**
Final number **nF** of words
Final number **mI** of bug reports
Output: Reduced data set **TFI** for bug triage
Steps:
1. Apply **IS** to **mI** bug reports of **T.**
2. Terminate **IS** when the number of bug reports is equal to or less than **mI** and generate a training set **TI .**
3. Apply **FS** to **n** words of **TI** and calculate objective values for all the words.
4. Select the top **nF** words of **TI** and generate the final training set **TIF.**

Combination of instance selection (IS) and feature selection (FS) to generate reduced bug data set. We replacing the original bug data set with the reduced bug data set for bug triage. Instance selection and feature selection are widely used techniques in data processing for given data set in a certain applications. Instance selection (IS) is to obtain subset of relevant instances or bug reports in bug data while Feature selection (FS) goal to obtain subset of relevant features or words in bug data.

To distinguish the orders of applying instance selection and feature selection we give the denotation as given an instance selection algorithm IS and feature selection algorithm FS.We use FS $\longrightarrow$ IS to denote the bug data reduction which first applies FS and then IS on the other side IS $\longrightarrow$ FS denotes first applying IS and then FS. In Algorithm 1 we briefly present how to reduce the bug data based on FS $\longrightarrow$ IS and in Algorithm 2 we present how to reduce the bug dataset based on IS $\longrightarrow$ FS. The output of bug data reduction is a new and reduced data set. Two algorithms FS and IS are applied sequentially. In this FS $\longrightarrow$ IS and IS $\longrightarrow$ FS are viewed as two orders of bug data reduction.

Instance selection is technique to reduce noisy and redundant instance an instance selection algorithm provide reduced data set by removing non representative instances. According to an existing review and study we choose instance selection algorithms namely Iterative Case Filter (ICF).

Feature selection is a preprocessing technique for selecting a reduced set of features for large scale datasets since, bug triage is converted into text classification focus on the feature selection algorithms in text data. In this project, we choose well-performed algorithm in text data reduction namely x2 statistic (CH).

Two main benefits of data reduction technique are reducing the Data Scale and Improving the Accuracy.

Reducing the Data Scale reduce scales of data sets to save the labor cost of developers.

Bug dimension is reducing duplicate and noisy bug report to decrease number of historical bugs 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.

Word dimension using feature selection to remove noisy or duplicate words in dataset. FS reduced dataset can be handled more easily by automatic technique than original dataset.

Improving the Accuracy is an important evaluation criterion for bug triage.

Bug dimension IS remove uninformative bug reports we can observe that the accuracy may be decreased by removing bug reports.

Word dimension by removing uninformative words feature selection of bug triage improves the accuracy of bug triage. It can recover accuracy loss by IS improves the

C. Prediction For Reduction Order:

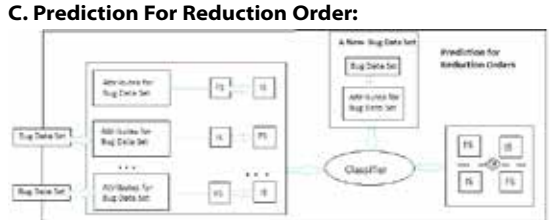

Fig 3: Production for Reduction Order Technique

Instance selection algorithm IS and feature selection algorithm FS, FS $\longrightarrow$ IS and IS $\longrightarrow$ FS are viewed as two orders for applying reducing techniques. Challenge is how to determine the order of reduction techniques that means how to choose one between FS $\longrightarrow$ IS and IS $\longrightarrow$ FS. We refers to this problem as the prediction for reduction orders.

Reduction Orders to apply data reduction to each new bug data set need to check the accuracy of both two orders FS $\longrightarrow$ IS and IS $\longrightarrow$ FS and choose a better one. we converts the problem of prediction for reduction orders into a binary classification problem. Bug data set is mapped to an instance and associated reduction order ei-

ther FS ⟶ IS or IS ⟶ FS is a mapped to the class of instances. Classifier can be trained only once when facing many new bug data sets.

Attributes for a Bug Data Set to build a binary classifier to predict the reduction orders. We assume the extract 18 attributes to describe bug data set. Such that we divide these 18 attributes into two categories are bug report category (B1 to B10) and developer category (D1 to D8).[8]

(C) **Data Preparation** is evaluate to the experimental results of the bug data set reduction we employ the bug data of Mozilla. We choose Mozilla since the bug data set is easy to obtain of bug triage can work well. For a new coming bug report have summary and description are representing items which are used for the manual bug triage. Thus for each bug report, we use the summary and description to obtain the words.

Index	Attribute name	Description
B1	# Bug reports	Total number of bug reports.
B2	# Words	Total number of words in all the bu reports.
g B3	Length bug reports	Average number of words of all the bug reports.
B4	# Unique words	Average number of unique words in each bug report.
B5	Ratio of sparseness	Ratio of sparse terms in the text matrix. A sparse term refers to a word with zero frequency in the text matrix.
B6	Entropy of severities	Entropy of severities in bug reports. Severity denotes the importance of bug reports.
B7	Entropy of priorities	Entropy of priorities in bug reports. Priority denotes the level of bug reports.
B8	Entropy of products	Entropy of products in bug reports. Product denotes the sub-project.
B9	Entropy of components	Entropy of components in bug reports. Component denotes the sub-sub-project.
B10	Entropy of words	Entropy of words in bug reports.
D1	# Fixers	Total number of developers who will fix bugs.
D2	# Bug reports per fixer	Average number of bug reports for each fixer.
D3	# Words per fixer	Average number of words for each fixer.
D4	# Reporters	Total number of developers who have reported bugs.
D5	# Bug reports per reporter	Average number of bug reports for each reporter.
D6	# Words per reporter	Average number of words for each reporter.
D7	# Bug reports by top 10 percent reporters	Ratio of bugs, which are reported by the most active reporters.
D8	Similarity between fixers and reporters	Similarity between the set of fixers and the Set of reporters, defined as the Tanimoto. similarity

(d) **Experimental Setup to compare the result** of experiments we employs the Naive Bayes classifier as the bug triage approach. To improve the quality of bug triage, we follow the existing work to use recommendation list. A list with the size k can provide k developers as the prediction result for each new coming bug report. [5]

The accuracy is significant evaluation for bug triage it measures the quality of prediction and two others precision and recall are used to measure correctness of bug triage

EXPECTED RESULT

We examine the results of bug data reduction on bug repositories of project Mozilla. For a project we evaluate results on five data sets and each data set is over 10,000 bug reports, which are fixed or duplicate bug reports. We check bug reports in the two projects and find out that 28.23 percent of bug reports in Mozilla are fixed or duplicate. We implement the instance selection (ICF) algorithm and the Naive Bayes classifier in our project.

Results of Feature Selection and Instance Selection we show the results of each algorithm in this part can presents the accuracy rates of the CHI and ICF on Mozilla. For CHI we select 10% 30% and 50% as the ratio of final number of words, such setup is based on experience of text feature selection. For ICF we set the ratio of final number of bug reports as 30% 50% and 70%. The ratio value is based on the instance selection.

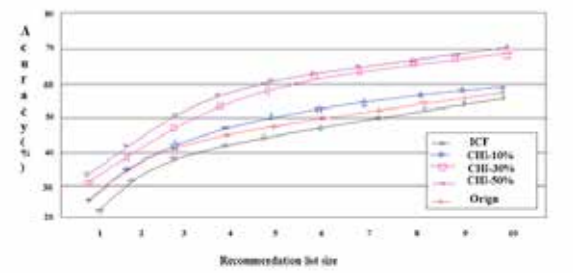

Fig 4: The accuracy rates of CH and ICF on Mozilla

It can be found CH techniques work much better than original experiments. The improvement of accuracy for Mozilla is up to 13% we can see that CH achieves good performance if 30% or 50% is selected as ratio of final number of words production staff in the same order provided by the author.

CONCLUSION AND FUTURE SCOPE

Bug triage is software maintenance in both labor cost and time cost. In this project we combines the feature selection with instance selection to reduce scale of bug data sets as well as improve the accuracy of data. To determine the order of applying instance and feature selection for new bug data set we extract attributes of bug data set and trains prediction model based on historical data sets. We investigate the data reduction for bug triage in bug repositories of large open source project is Mozilla. In our work we provide approach to leveraging techniques on data processing into the form of reduced and high quality bug data in development in software and maintenance.

In future work apply the training set reduction of the bug triage to improve the software quality, Since machine learning becomes one of the powerful tools in software engineering training data set reduction can be useful for work based on machine learning.

ACKNOWLEDGMENT:

Many thanks to Mrs. G. V. Mane with Department of Computer Engineering, Savitribai phule Pune University for sharing the labeling heuristic for bug triage. Thanks to MMIT computer labs for providing related tool and suggestion.

REFERENCES:

1. Dr. Cubranic and G.C.Murphy, "Automatic bug triage using text categorization" Jun 2004, pp.92_97.
2. J. Anvik, "Automating bug report assignment," Proc. Intl. Conf. Software Engineering (ICSE 06), ACM, May 2006, pp. 937-940.
3. D. Matter, A. Kuhn, and O. Nierstrasz, "Assigning bug reports using a vocabulary-based expertise model of developers," IEEE, May 2009, pp. 131-140.
4. C. C. Aggarwal and P. Zhao, "Towards Graphical models for text processing", Aug 2013.
5. Mozilla. (2014). [Online]. Available: http://mozilla.org/
6. Bugzilla, (2014). [Online]. Available: http://bugzilla.org/
7. Yan Hu, Jifeng Xuan, "Towards Training Set Reduction for Bug Triage," Feb 2015, pp. 131-140.
8. Laxman M., Anita K., "Automated Bug triage System using the data reduction techniques", Oct 2015.