International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

USING DATA REDUCTION TECHNIQUES FOR EFFECTIVE BUG TRIAGE

P.ARUMUGAM #1, M.PRIYA*2
#1 M.Phil Scholar , Sri Vasavi College , Erode, Tamilnadu, India
*2 Head of the Department Computer ScienceUG, Sri Vasavi College (SFW), Erode, Tamilnadu, India

ABSTRACT- Data mining is the practice of evaluating large pre-existing set of data in order to generate the new proceeds data. Bug triaging is just like a part of mining in which it assign a bug from large data set to the developer. Currently, large amount of economic share of IT sector is spending on manually handling the bugs. Triaging is incredible aspect of development that benefits users and developer alike. It was seen that in numbers of approaches they try to reduce the bugs assigning problem but also suffer from number of problems like economic rise, redundant data, poor productivity etc. To overcome all issues Bud triage with reduction technique is adapted by us. It uses Feature Selection and Instance selection reduction algorithm which will help to reduce the data by scale and dimension and provide us with rich repository of data. This will be beneficial to predict out the correct developer to whom the bug can be assigned. Anatomization (Analysis) of Bug triage will show that how the combination of Feature selection and Instance selection will help to achieve the good set of result.

1. INTRODUCTION

Programming organizations spend more than 45 percent of expense in managing programming bugs. An unavoidable stride of fixing bugs is bug triage, which plans to accurately allocate a designer to another bug. To diminish the time cost in manual work, content classification systems are connected to lead programmed bug triage. In this paper, we address the problem of information lessening for bug triage, i.e., how to decrease the scale and enhance the nature of bug information.

We consolidate case choice with highlight determination to all the while lessen information scale on the bug measurement and the word measurement. To focus the request of applying occasion determination and highlight choice, we remove properties from authentic bug information sets and construct a prescient model for another bug information set. We observationally research the execution of

Information diminishment on absolutely 600,000 bug reports of two vast open source ventures, to be specific Eclipse and Mozilla. The outcomes demonstrate that our information diminishment can viably lessen the information scale and enhance the precision of bug triage. Our work gives a way to deal with utilizing strategies on information preparing to shape decreased and superb bug information in programming advancement and support.

Bug triage is an extravagant stride of programming support in both work cost and time cost. In this paper, we consolidate highlight choice with occurrence choice to decrease the size of bug

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

information sets and also enhance the information quality. To focus the request of applying case determination and highlight determination for another bug information set, we concentrate traits of every bug information set and train a prescient model taking into account authentic information sets. We observationally research the information lessening for bug triage in bug vaults of two expansive open source ventures, to be specific Eclipse and Mozilla.

Our work gives a way to deal with utilizing systems on information preparing to shape lessened and brilliant bug information in programming advancement and support. In future work, we anticipate enhancing the consequences of information decrease in bug triage to investigate how to set up a high quality bug information set and tackle a space specific programming assignment. For foreseeing decrease orders, we plan to pay endeavors to find out the potential relationship between the characteristics of bug information sets and the decrease orders. The instance selection and feature selection techniques are used to eliminate the stop words and create a reduced bug dataset and then K-NN approach is used to classify data. We are using combinational approach of both in our proposed paper and then we are applying Support Vector Machine classifier to triage the bug to appropriate developer for bug resolution.

2. RELATED WORK

A. Data Reduction Techniques for Proficient Bug Triage

We formalize a new cryptographic IT companies give over 45% of revenue in marketing with software bugs. Bug triage is important step in fixing bugs; its aim is to correctly assign a developer to new bug. To reduce time, text classification methods are adapted to perform automatic bug triage. Here, we concentrate on how to minimize the scale and improvement of quality of bug data. We combine instance selection with feature selection to reduce data size on the bug dimension and the word dimension simultaneously. Attributes from historical bug datasets is collected for determining the order to be applied for instance and feature selection, predictive model is implemented for a new bug data set. Our contribution results in reduced and high-quality bug data.

A bug repository plays a vital role in managing software bugs. Software bugs are unavoidable and fixing bugs in IT development is more expensive. IT companies give over 45% of revenue in marketing with software bugs. Extensive software projects setup bug repositories (also called bug tracking systems) to base information gathering and help the developers manage bugs.

A bug repository maintains a bug in a bug report, this report maintains textual description of reproducing the bug and updates according to the state of bug fixing. Bug repository issues a data platform to support several kinds of operations on bugs such as fault prediction, bug localization, and reopened bug analysis .In this, bug reports in a bug repository are called bug data. There are 2 challenges related to bug data which may influence the efficient use of bug repositories in software development tasks that is the large scale and the low quality. A large number of new bugs are stored in bug repositories because of the daily report bugs. It is a challenge to manually test such large-scale

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

bug data. As far as this, software techniques experience from the low quality of bug data. Noise and redundancy are 2 typical features of Low-quality bugs. A noisy bug deceives the related developers whereas redundant bugs waste time in handling bugs.

B. A Tabu Search Approach to the Condensed Nearest.

This paper presents a new approach to the selection of prototypes for the nearest neighbour rule which aims at obtaining an optimal or close-to-optimal solution. The problem is stated as a constrained optimization problem using the concept of consistency. In this context, the proposed method uses tabu search in the space of all possible subsets. Comparative experiments have been carried out using both synthetic and real data in which the algorithm has demonstrated its superiority over alternative approaches. The results obtained suggest that the tabu search condensing algorithm offers a very good trade-off between computational burden and the optimality of the prototypes selected.

Among other nonparametric approaches, distance-based classification rules are especially appealing both to researchers and practitioners because of their interesting properties for performance and implementation issues. In particular, the (k-)nearest neighbour (NN) rules frequently appear in the specialized literature either by themselves, or as a common reference to compare the performance of other approaches. This is due not only to their very good asymptotic behaviour even for the plain NN (1-NN) rule but also for the convenient trade-off between ease of implementation and performance in practical situations.

I. 3. PROPOSED SCHEME OF WORK

In this project we address the problem of data reduction for bug triage, 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. Data reduction for bug triage aims to build a small-scale and high-quality set of bug data by removing bug reports and words, which are redundant or non-informative. In our work, we combine existing techniques of instance selection and feature selection to simultaneously reduce the bug dimension and the word dimension. The reduced bug data contain fewer bug reports and fewer words than the original bug data and provide similar information over the original bug data. We evaluate the reduced bug data according to two criteria: the scale of a data set and the accuracy of bug triage.

In this project, we propose a predictive model to determine the order of applying instance selection and feature selection. We refer to such determination as prediction for reduction orders.

Drawn on the experiences in software metrics,1 we extract the attributes from historical bug data sets. Then, we train a binary classifier on bug data sets with extracted attributes and predict the order of applying instance selection and feature selection for a new bug data set.

ADVANTAGES

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

1. Applying the feature selection technique can reduce words in the bug data and the accuracy can be increased.
2. Meanwhile, combining both techniques can increase the accuracy, as well as reduce bug reports and words.
3. Based on the attributes from historical bug data sets, our predictive model can provide the accuracy of 71.8 percent for predicting the reduction order.
4. To present the problem of data reduction for bug triage. This problem aims to augment the data set of bug triage in two aspects, namely a) to simultaneously reduce the scales of the bug dimension and the word dimension and b) to improve the accuracy of bug triage.
5. Propose a combination approach to addressing the problem of data reduction. This can be viewed as an application of instance selection and feature selection in bug repositories.
6. To build a binary classifier to predict the order of applying instance selection and feature selection. To our knowledge, the order of applying instance selection and feature selection has not been investigated in related domains.

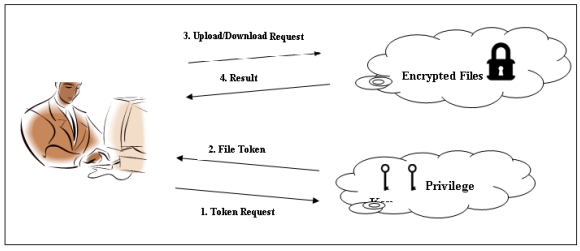

Fig 1 : Architecture for Authorized Deduplication

SYSTEM MODEL

In this first module, we develop some entities:

DATASET COLLECTION

To collect and/or retrieve data about activities, results, context and other factors. It is important to consider the type of information it want to gather from your participants and the ways you will analyze that information. The data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable. after collecting the data to store the Database.

PREPROCESSING METHOD

Data pre-processing or Data cleaning, Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data. And also used to removing the unwanted data. Commonly used as a preliminary data mining practice, data pre-processing transforms the data into a format that will be more easily and effectively processed for the purpose of the user.

FEATURE SELECTION/ INSTANCE SELECTION

The combination of instance selection and feature selection to generate a reduced bug data set. We replace the original data set with the reduced data set for bug triage. Instance selection is a technique to reduce the number of instances by removing noisy and redundant instances. By removing uninformative words, feature selection

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

improves the accuracy of bug triage. It recover the accuracy loss by instance selection.

BUG DATA REDUCTION

The data set can reduce bug reports but the accuracy of bug triage may be decreased. It improves the accuracy of bug triage. It tends to remove these words to reduce the computation for bug triage. The bug data reduction to reduce the scale and to improve the quality of data in bug repositories. It is reducing duplicate and noisy bug reports to decrease the number of historical bugs.

PERFORMANCE EVALUATION

In this Performance evaluation, algorithm can provide a reduced data set by removing non-representative instances. The quality of bug triage can be measured with the accuracy of bug triage. To reduce noise and redundancy in bug data sets.

4. PERFORMANCE STUDY

The Bug Repository data set is a collection of models and metrics of software systems and their histories. The goal of such a data set is to allow people to compare different bug prediction approaches and to evaluate whether a new technique is an improvement over existing ones. In particular, the data set contains the data needed to.

Prediction technique is run based on source code metrics and/or historical measures and/or process information (logs data).

Compute the performance of the prediction by comparing its results with a set, i.e., the number post release defects reported in bug tracking system. Bug prediction is performed at the class level using the designed data set. However class data is aggregated to derive the package or subsystem information, since for each class it specifies the package that contains it.

BUG REPORT ANALYSIS

An automatic bug triaging system is presented by recommending one experienced developer for each new bug report. These following steps are performed.

1. Representation Framework

We have a collection of bug reports, B = {b1, b|B|}. Each bug report has a collection of term, T = {t1… t|T|}, and a class label (developer), c C = {c1, …, c|C|}.

2. Term selection methods

High dimensionality of term space is reduced using term selection by selecting the most discriminating terms for classification task. The methods give a weight for each term in which terms with higher weights are assumed to contribute more for classification task than terms with lower weights.

3. Chi-Square (X2)

In statistics, the X2 test is used to examine independence of two events. The events, X and Y, are assumed to independent if P (XY) = P(X)P(Y) term selection, the two events are the occurrence of the term and the occurrence of the class.

4. Term Frequency Relevance Frequency (TFRF)

The basic behind the TFRF method is that the more high frequency for a term in the positive category than in negative category, the more contributions it makes in selecting the positive instances from negative instances.

5. Distinguishing Feature Selector (DFS)

DFS is new novel term selection method. It provides global discriminatory powers of the features over entire text collection rather than being class specific. DFS considers the following requirements:

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

Distinct term is a term that occurs frequently in a single class and not in other classes,

• Irrelevant term is a term that rarely occurs in a single class and not in other classes,

• Distinctive term is a term that occurs frequently in all classes is irrelevant,

• A term that occurs in some of the classes. DFS assigns scores to features between 0.5 (least discriminative) and 1.0 (most discriminative).

TASK SCHEDULING AND NOTIFICATION

The development group with has run into this issue so many times that we decided to write our own solution to this problem and make solve, this module reaches up to developer as well as customer from Bug Report Analysis phase . It's called Revalee as in reveille: a signal to arise and it is a Service that freezes your job requests in overall process until it's time to thaw them out.

5. CONCLUSION AND FUTURE WORK

Bug triage is an expensive step of software maintenance in both labor cost and time cost. In this project, 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, to extract attributes of each bug data set and train a predictive model based on historical data sets. Empirically investigate the data reduction for bug triage in bug repositories of two large open source projects, namely Eclipse and Mozilla. 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.

6. REFERENCES:

[1] J. Anvik and G. C. Murphy, "Reducing the effort of bug report triage: Recommenders for development-oriented decisions," ACM Trans. Soft. Eng. Methodol., vol. 20, no. 3, article 10, Aug. 2011.

[2] D. _Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004, pp. 92–97.

[3] P. S. Bishnu and V. Bhattacherjee, "Software fault prediction using quad tree-based k-means clustering algorithm," IEEE Trans. Knowl. Data Eng., vol. 24, no. 6, pp. 1146–1150, Jun. 2012.

[4] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," Data Mining Knowl. Discovery, vol. 6, no. 2, pp. 153–172, Apr. 2002.

[5] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, "Information needs in bug reports: Improving cooperation between developers and users," in Proc. ACM Conf. Comput. Supported Cooperative Work, Feb. 2010, pp. 301–310.

[6] V. Cerver_on and F. J. Ferri, "Another move toward the minimum consistent subset: A tabu search approach to the condensed nearest neighbor

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 8, 2017 ISSN: 2348-6600
Reference ID: IJCS-193 PAGE NO: 1183-1189
Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

rule," IEEE Trans. Syst., Man, Cybern., Part B, Cybern., vol. 31, no. 3, pp. 408–413, Jun. 2001.

[7] M. Grochowski and N. Jankowski, "Comparison of instance selection algorithms ii, results and comments," in Proc. 7th Int. Conf. Artif. Intell. Softw. Comput., Jun. 2004, pp. 580–585.