ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Towards Operative Bug Triage with Software Records Decrease Procedures

Farhat Azeema
PG Scholar,
Dept of Computer Science,
Sri Indu Institute of Engineering & Technology.

K.V Nanda Kishore
Associate Professor,
Dept of CSE,
Sri Indu Institute of Engineering & Technology.

Dr. I.Satyanarayana
Principal,
Sri Indu Institute of Engineering & Technology.

Abstract:
In order to diminution the time cost in physical work, text classification practices are pragmatic to conduct automatic bug triage. Presently, software companies spend over 45 percent of cost in dealing with software bugs. A foreseeable step of fixing bugs is bug triage, which ambitions to decorously consign a developer to a new bug. In this scheme, it is addressed the problem of data reduction for bug triage, to reduce the scale and progress the reputation of bug data. It is here combined with instance selection with feature selection to simultaneously reduce data scale on the bug dimension and the word dimension. To define the order of applying instance selection and feature selection, it is extract attributes from historical bug data sets and build a predictive model for a new bug data set. It is practically scrutinized the enactment of data reduction on totally 600,000 bug reports of two large open source projects, namely Eclipse and Mozilla. The work provides an approach to leveraging techniques on data processing to form reduced and high-quality bug data in software development and maintenance.

Keywords:
Mining software repositories, ata preprocessing, data management in bug repositories, bug data reduction, feature selection, instance selection, bug triage.

I. INTRODUCTION:

In current software development, software depositories are large-scale databases for stowing the output of software development, like source code, bugs, emails, and specifications.

Traditional software analysis is not completely suitable for the large-scale and complex data in software repositories [5].Data mining has emerged as a promising means to handle software data. By leveraging data mining techniques, mining software repositories can uncover interesting information in software repositories and solve real world software problems. A bug repository (a typical software repository, for storing details of bugs), plays an important role in managing software bugs. Software bugs are inevitable and fixing bugs is expensive in software development. Software companies spend over 45 percent of cost in fixing bugs [9].

Large software projects deploy bug repositories (also called bug tracking systems) to support information collection and to assist developers to handle bugs[3]. In a bug repository, a bug is maintained as a bug report, which records the textual description of reproducing the bug and updates according to the status of bug fixing [4]. A bug repository provides a data platform to support many types of tasks on bugs, e.g., fault prediction [7] bug localization [2], and reopened bug analysis [3]. In this scheme, bug reports in a bug repository are called bug data.

There are two challenges related to bug data that may affect the effective use of bug repositories in software development tasks, namely the large scale and the low quality. On one hand, due to the daily-reported bugs, a large number of new bugs are stored in bug repositories. It is a task to physically examine such large-scale bug data in software development.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

On the other side, software techniques suffer from the low quality of bug data. Two typical characteristics of low-quality bugs are noise and redundancy. Noisy bugs may mislead related developers [4] while redundant bugs waste the limited time of bug handling. A time-consuming step of handling software bugs is bug triage, which aims to assign a correct developer to fixa new bug [1]. In outmoded software development, new bugs are manually triaged by an expert developer, i.e., a human triager. Due to the large number of daily bugs and the lack of expertise of all the bugs, manual bug triage is expensive in time cost and low in accuracy. To avoid the expensive cost of manual bugtriage, existing work [1] has proposed an automatic bug triage approach, which applies text classification techniquesto predict developers for bug reports.

In this method, a bug report is mapped to a document and a related developer is mapped to the label of the document. Then, bug triage is converted into a problem of text classification and is automatically solved with mature text classification techniques, e.g., Naive Bays [12]. Based on the results of text classification, a human triage assigns new bugs by incorporating his/her expertise. To expand the accuracy of text classification techniques for bug triage, some further techniques are investigated, e.g., a tossing graph approach [2] and a collaborative filtering approach[40]. However, large-scale and low-quality bug data in bug repositories block the techniques of automatic bug triage. Since software bug data are a kind of free-form text data (generated by developers), it is necessary to generate. It is 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[5].

Data reduction for bug triage aims to build a small-scale and high-qualityset of bug data by removing bug reports and words,which are redundant or non-informative. In the work, It iscombine existing techniques of instance selection and feature selection to simultaneously reduce the bug dimensionand the word dimension[9]. The reduced bug data contain bug reports and words than the original bug data and provide similar information over the original bug data. It is evaluate the reduced bug data according to two criteria: the scale of a data set and the accuracy of bug triage. To avoid the bias of a single algorithm, It is empirically examine the results of the instance selection algorithms and the feature selection algorithms. Given an instance selection algorithm and a feature selection algorithm, the order of applying these two algorithms may affect the results of bug triage. In this scheme, It is propose predictive model to determine the order of applying instance selection and feature selection. It is refer to such determination[7] as prediction for reduction orders. Drawn on the experiences in software metrics,1 It is extract the attributes from historical bug data sets. Then, it is train a binary classifier on bug data sets with extracted attributes and predicts the order of applying instance selection and feature selection for a new bug data set[10].The primary contributions of this scheme are as follows:

1) It is 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.2) It is 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 [12].3) It is build a binary classifier to predict the order of applying instance selection and feature selection. To the knowledge, the order of applying instance selection and feature selection has not been investigated in related domains [12].

II. LITERATURE SURVEY:

Bug repositories are widely used for maintaining software bugs, e.g., a popular and open bug repository, Bugzilla[5]. Once a software bug is found, a reporter (typically a developer, a tester, or an end user) records this bug to the bug repository.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A recorded bug is called a bug report, which has multiple items for detailing the information of reproducing the bug. In Fig. 1, it is show a part of bug report for bug 284541 in Eclipse.2 In a bug report, the summary and the description are two key items about the information of the bug, which are recorded in natural languages. As their names suggest, the summary denotes a general statement for identifying a bug while the description gives the details for reproducing the bug. Some other items are recorded in a bug report for facilitating the identification of the bug, such as the product, the platform, and the importance. 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. For example, in Fig. 1, the developer DimitarGiormov is the final assigned-to developer of bug 284541.A developer, who is assigned to a new bug report, starts to fix the bug based on the knowledge of historical bug fixing[6], [14]. Typically, the developer pays efforts to understand the new bug report and to examine historically fixed bugs as a reference .An item status of a bug report is changed according tithe current result of handling this bug until the bug incompletely fixed. Changes of a bug report are stored in an item history. Table 1 presents a part of history of bug 284541.This bug has been assigned to three developers and only the last developer can handle this bug correctly. Changing developers lasts for over seven months while fixing this bug only costs three days. Manual bug triage by a human triage is time consuming and error-prone since the number of daily bugs is large to correctly assign and a human triage is hard toaster the knowledge about all the bugs [12]. Existing work employs the approaches based on text classification to assist bug triage, e.g., [1], [5]. In such approaches, the summary and the description of a bug report are extracted as the textual content while the developer who can fix this bug is marked as the label for classification. Then techniques on text classification can be used to predict the developer for a new bug. In details, existing bug reports with their developers are formed as a training set-to train a classifier new bug reports are treated as attest set to examine the results of the classification. Into avoid the low accuracy of bug triage, a recommendation list with the size k is used to provide list of k developers, who have the top-k possibility to fix the new bug.

III. SYSTEM DESIGN

A. Data reduction for bug triage

The bug data reduction in the work, which is applied as a phase in data preparation of bug triage. It is combine existing techniques of instance selection and feature selection to remove certain bug reports and words[12]. A problem for reducing the bug data is to determine the order of applying instance selection and feature selection, which is denoted as the prediction of reduction orders. In this section, it is first present how to apply instance selection and feature selection to bug data, i.e., data reduction for bug triage. Then, it is list the benefit of the data reduction.

Algorithm 1. Data reduction based on FS→S

Input: training set T with n words and m bug reports,

reduction order FS→S

final number no of words,

Final number mI of bug reports,

Output: reduced data set T FI for bug triage

1) apply FS to n words of T and calculate objective values

for all the words;

2) select the top n_o words of T and generate a training

set T F ;

3) apply IS to mI bug reports of T_F ;

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4) terminate IS when the number of bug reports is equal toor less than m_ind generate the final training set T_{FI} .

B. Techniques used in data processing: Instance Selection and Feature Selection

In bug triage, a bug data set is converted into a text matrix with two dimensions, namely the bug dimension and the word dimension. In the current work, it is leverage the combination of instance selection and feature selection to generate reduced bug data set. It is replaced the original data set with the reduced data set for bug triage. Instance selection and feature selection are widely used techniques in data processing. 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) [18] while feature selection aims to obtain a subset of relevant features(i.e., words in bug data) [19]. In the work, It is employ the combination of instance selection and feature selection. In the work, FS ! IS and IS ! FS are viewed It is as two orders of bug data reduction. To avoid the bias from a single algorithm, it is examine results of the typical algorithms of instance selection and feature selection, respectively.

It is briefly introduce these algorithms as follows. Instance selection is a technique to reduce the number of instances by removing noisy and redundant instances [11],[8]. An instance selection algorithm can provide a reduced data set by removing non-representative instances [18]According to an existing comparison study and an existing review [37], It is choose the instance selection algorithms, namely Iterative Case Filter (ICF) [8], Learning Vectors Quantization (LVQ) [27], Detrimental Reduction Optimization Procedure (DROP), and Patterns by Ordered Projections (POP)[14].Feature selection is a preprocessing technique for selecting reduced set of features for large-scale data sets [15],[9]. The reduced set is considered as the representative features of the original feature set [20]. Since bug triage is converted into text classification, It is focus on the feature selection algorithms in text data. In this scheme, it is choosing the It sill-performed algorithms in text data [23] and software data [21], namely Information Gain (IG) ,x^2 statistic(CH), Symmetrical Uncertainty attribute evaluation(SU) and Relief-F Attribute selection (RF). Based on feature selection, words in bug reports are sorted according to their feature values and a given number of words with large values are selected as representative features. Accuracy: is an important evaluation criterion for bug triage. In the work, data reduction explores and removes noisy or duplicate information in data sets. Bug dimension: Instance selection can remove uninformative bug reports; meanwhile, It is can observe that the accuracy may be decreased by removing bug reports Word dimension: By removing uninformative words, feature selection improves the accuracy of bug triage This can recover the accuracy loss by instance selection.

IV. PROPOSED ARCHITECTURE

A. Prediction for reduction orders

To avoid the time cost of manually checking both reduction orders, It is consider predicting the reduction order for a new bug data set based unhistorical data sets[21].It is converting the problem of prediction for reduction orders into a binary classification problem. A bug data set is mapped to an instance and the associated reduction order (either FS ! IS or IS ! FS) is mapped to the label of a class of instances. Fig. 3 summarizes the steps of predicting reduction orders for bug triage. Note that a classifier can be trained only once when facing many new bug data sets[23]. That is, training such a classifier once can predict the reduction orders for all the new datasets without checking both reduction orders. To date, the problem of predicting reduction orders of applying feature selection and instance selection has not been investigated another application scenarios.[24]From the perception of software engineering, predicting the reduction order for bug data sets can be visited as a kind of software metrics, which involves activities for measuring some property for a piece of software [16]. HoItisver,the features in the work are extracted from the bug data set while the features in current work on software metrics are for individual software artifacts,3 e.g., an

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

individual bug report or an individual piece of code. In this scheme, to avoid ambiguous denotations, an attribute refers to an extracted feature of a bug data set while a feature refers to a word of a bug report.

B. Experimental evaluation: Data sets and evaluation

It is examine the results of bug data reduction on bug repositories of two projects, Eclipse and Mozilla [16]. For each project, it is evaluate results on five data sets and each data set is over 10,000 bug reports, which are fixed or duplicate bug reports. It is check bug reports in the two projects and find out that 45.44 percent of bug reports in Eclipse and28.23 percent of bug reports in Mozilla are fixed or duplicate. Thus, to obtain over 10,000 fixed or duplicate bug reports, each data set in Eclipse is collected from continuous 20,000 bug reports while each bug set in Mozilla is collected from continuous 40,000 bug reports. Table 3 lists the details of ten data sets after data preparation. [17]

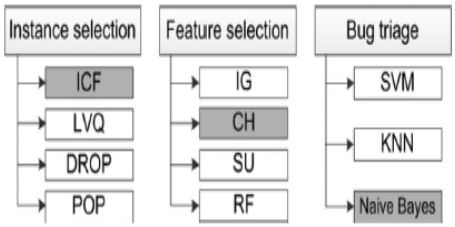
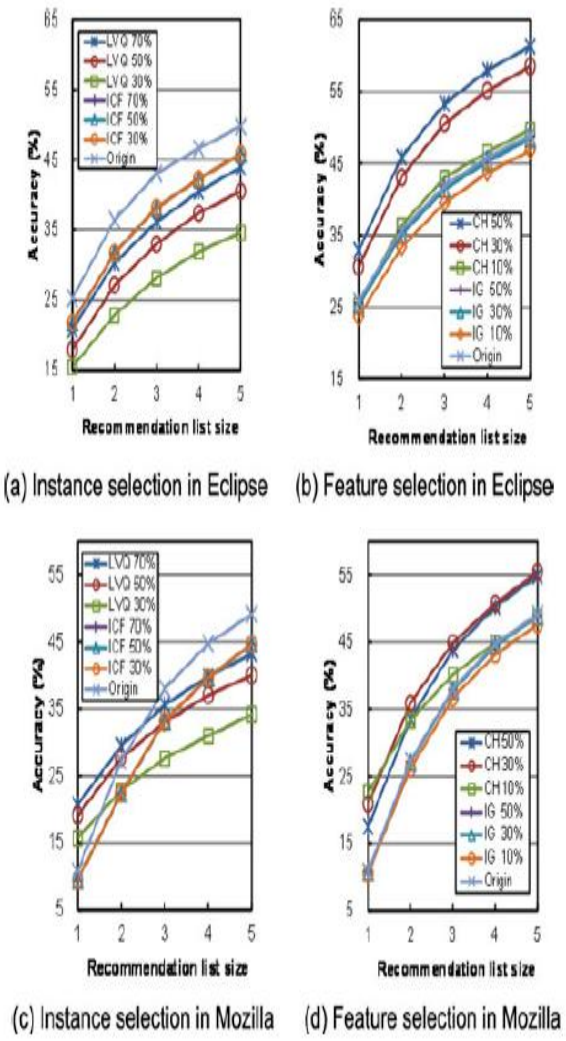


Fig1. Algorithms for instance selection, feature selection, and bug triage.

Among these algorithms, ICF, CH, and Naive Bays are It sill-performed based on the experiments of the bug data reduction. The results of data reduction for bug triage can be measured in two aspects, namely the scales of data sets and the quality of bug triage. Based on Algorithm 1, the scales of data sets (including the number of bug reports and the number of words) are configured as input parameters. The quality of bug triage can be measured with the accuracy of bug triage, which is defined as Accuracy = # correctly assigned bug reports in k candidate # all bug reports in the test set.



(a) Instance selection in Eclipse (b) Feature selection in Eclipse

(c) Instance selection in Mozilla (d) Feature selection in Mozilla

C. Experiments on Prediction for Reduction Orders: Data Sets and Evaluation

It is present the experiments on prediction for reduction orders in this part. It is map a bug data set to an instance, and map the reduction order (i.e., FS! IS or IS !FS.)to its label. Given a new bug data set, it is train a classifier to predict its appropriate reduction order based on historical bug data sets. To train the classifier, it is labeled each bug data set with its reduction order. In the work, one bug unit denotes 5,000 continuous bug reports. In Section5.1, It is having collected 298,785 bug reports in Eclipse and 281,180 bug reports in Mozilla. Then, 60 bug units (298,785/5,000=59:78) for Eclipse and 57 bug units (281,180/5,000=56:24) for Mozilla are obtained. Next,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

it is form bug data sets by combining bug units to training classifiers. It is considered continuous one to seven bug units as one dataset on Mozilla and finally collects 399 (57 X 7) bug datasets.

V. Conclusion and Future Enhancement:

In this pattern, it is collective feature selection with instance selection to reduce the scale of bug data sets as it progress the data quality. Bug triage is an affluent step of software maintenance in both labor cost and time cost. To regulate the order of applying instance selection and feature selection for a new bug data set, it is extracted the attributes of each bug data set and train a extrapolative model based on historical data sets. It is practically pondered the data reduction for bug triage in bug repositories of two large open source projects, namely Eclipse and Mozilla. The work provides an methodology to leveraging techniques on data processing to form condensed and high-quality bug data in software development and maintenance. In future work, it is planned on taming 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 expecting reduction orders, it is planned to pay efforts to find out the potential relationship between the attributes of bug data sets and the reduction orders.

References:

[1]J. Anvil, L. Hew, and G. C. Murphy, "Who should fix this bug?"in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.

[2]S. Artzi, A. Kie_zun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D.Ernst, "Finding bugs in It isb applications using dynamic test generationand explicit-state model checking," IEEE Softw., vol. 36,no. 4, pp. 474–494, Jul./Aug. 2010.

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

[4] C. C. Agawam and P. Zhao, "Towards graphical models for text processing," Know. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.

[5]Bugzilla, (2014). [Online].Avaialble: http://bugzilla.org/

[6]K. Balog, L. Azzopardi, and M. de Rijke, "Formal models forexpert finding in enterprise corpora," in Proc. 29th Annu. Int. ACMSIGIR Conf. Res. Develop. Inform. Retrieval, Aug. 2006, pp. 43–50.

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

[8] H. Brighton and C. Mellish, "Advances in instance selection forinstance-based learning algorithms," Data Mining Know. Discovery,

vol. 6, no. 2, pp. 153–172, Apr. 2002.

[9] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, "Informationneeds in bug reports: Improving cooperation betItisen developersand users," in Proc. ACM Conf. Comput. Supported CooperativeWork, Feb. 2010, pp. 301–310.

[10]V. Bol_on-Canedo, N. S_anchez-Maro~no, and A. Alonso-Betanzos,"A review of feature selection methods on synthetic data," Knowl.Inform. Syst., vol. 34, no. 3, pp. 483–519, 2013.

[11]V. Cerver_on and F. J. Ferri, "Another move toward the minimumconsistent subset: A tabu search approach to the condensed nearestneighbor rule," IEEE Trans. Syst., Man, Cybern., Part B, Cybern.,vol. 31, no. 3, pp. 408–413, Jun. 2001.

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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[13]Eclipse. (2014). [Online]. Available: http://eclipse.org/

[14] B. Fitzgerald, "The transformation of open sthece software," MISQuart., vol. 30, no. 3,pp. 587–598, Sep. 2006. [32] D. Lo, J. Li, L. Wong, and S. C. Khoo, "Mining iterative generatorsand representative rules for software specification discovery,"IEEE Trans. Knowl. Data Eng., vol. 23, no. 2, pp. 282–296, Feb. 2011.

[15]Mozilla. (2014). [Online]. Available: http://mozilla.org/

[16]D. Matter, A. Kuhn, and O. Nierstrasz, "Assigning bug reportsusing a vocabulary-based expertise model of developers," in Proc.6th Int. Working Conf. Mining Softw. Repositories, May 2009,pp. 131–140.

[17] G. Miao, L. E. Moser, X. Yan, S. Tao, Y. Chen, and N. Anerousis,"Generative models for ticket resolution in expert networks," inProc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining,2010, pp. 733–742.

[18] E. Murphy-Hill, T. Zimmermann, C. Bird, and N. Nagappan, "Thedesign of bug fixes," in Proc. Int. Conf. Softw. Eng., 2013, pp. 332–341.

[19] J. A. Olvera-L_opez, J. A.Carrasco-Ochoa, J. F. Mart_ınez-Trinidad,and J. Kittler, "A review of instance selection methods," Artif.Intell.Rev., vol. 34, no. 2, pp. 133–143, 2010.

[20] J. A. Olvera-L_opez, J. F. Mart_ınez-Trinidad, and J. A. Carrasco-nOchoa, "Restricted sequential floating search applied to objectselection," in Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.,2007, pp. 694–702.

[21]R. S. Pressman, Software Engineering: A Practitioner's Approach, 7th ed. New York, NY, USA: McGraw-Hill, 2010.

[22] J. W. Park, M. W. Lee, J. Kim, S. W. Hwang, and S. Kim,"Costriage: A cost-aware triage algorithm for bug reporting systems,"in Proc. 25th Conf. Artif. Intell., Aug. 2011, pp. 139–144.

[23] J. C. Riquelme, J. S. Aguilar-Ru_ız, and M. Toro, "Finding representativepatterns with ordered projections," Pattern Recognit., vol. 36,pp. 1009–1018, 2003.

Author's Details:

Farhat Azeema
PG Scholar, Dept of CS,
Sri Indu Institute of Engineering & Technology.

K.V Nanda Kishore
Associate Professor, Dept of CSE,
Sri Indu Institute of Engineering & Technology.

Dr. I.Satyanarayana
Completed B.E-Mechanical Engg. from Andhra University, M.Tech Cryogenic Engg. Specilization-IIT Kharagpur, Ph.D-Mechanical Engg.-JNTUH, Currently working as an Principal at Sri Indu Institute of Engg. & Tech, Sheriguda(Vi), IBP(M),RR Dist.