ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Implementation of Efficient Approach for Bug Triage System with Data Classification Technique

Shreya Arudkar[#1], Amit Pimpalkar[#2]

Department of Computer Science & Engineering, Rashtrasant Tukadoji Maharaj Nagpur University,

Nagpur, Maharashtra, India[#1,2]

ABSTRACT*:* Software companies spend more than 45 percent of cost in fixing bugs. Software bugs can never be avoidable. Traditionally these bugs fix by manual assignment to a particular developer; this approach causes too much dependency. The alternative approach is the Bug Triage System, which fixes the bug and assigns the reported bug to a developer automatically so that it decreases the time and cost in manual work. Combination of instance selection and feature selection is to simultaneously reduce data scale on the bug dimension and the word dimension. We applied machine learning technique in bug triage to predict which developer should be assigned on the bug, based on its description by applying text categorization.

KEYWORDS: data reduction, instance selection, feature selection, bug triage, machine learning technique.

I. INTRODUCTION

Machine Learning (ML) algorithms allowing us to predict and determine the correct action in many domains based solely on prior knowledge with almost no human interaction. The possible applications of these techniques are very broad ranging from medicine to transportation, engineering, research and many more.

To reduce the effort necessary to maintain the list of the issues, the software development community came up with web-based applications designed specifically to track them. This type of software is called bug tracker where one entry of an issue is called a bug report. A bug report usually contains the summary, description and assignee of the discovered bug, as well as other fields (priority, status, comments etc.). Assignee is the developer who is assigned to investigate and possibly resolve the bug. This naturally raises an important question—who should fix the bug? The goal of this thesis is to investigate, propose and eventually implement a solution that answers this question in real-time and requires nearly no human interaction.

Software bugs can never be avoidable and at the same time fixing bugs is expensive in software development. Large software projects deploy bug repositories to support information collection and to assist developers to handle bugs. A bug repository will contain all report which plays an important role in managing software bugs. In a bug repository, a bug is maintained as a bug report, which records the complete description of reproducing the bug and updates according to the status of bug fixing. A bug repository gives a data platform to support many types of tasks on bugs, e.g., fault prediction, bug localization, and reopened bug analysis.

The bug reports in a bug repository are called bug data. This bug report is then assigns to a developer, who starts to fix the bug. If the assigned developer is unable fix the bug, the bug is migrated to another developer. The process of assigning a bug report to an appropriate developer is called bug triage. As the work of BTS is to choose the appropriate developer for fixing bugs, follow the existing work to remove unfixed bug reports.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

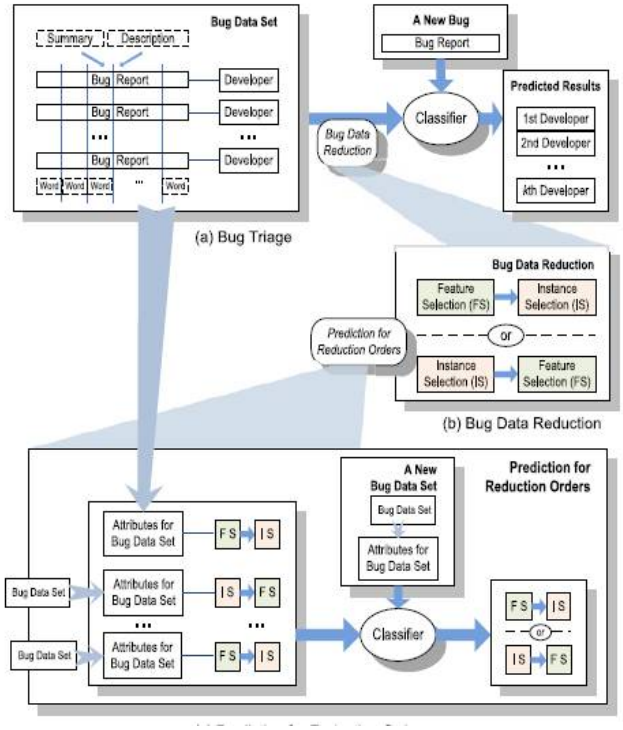

Figure 1. Architecture for Bug Tracking System

II. RELATED WORK

Cubrani proposed the problem of automatic bug triage. The Machine Learning technique, Text Categorization was applied to assist in bug triage by using text categorization. Text categorization is a technique of automatically sorting a set of documents into categories from a predefined set where a developer gets predicted using the bug's description, for this they had used supervised machine learning technique using Naive Bayes classifier to predict the correct developer [5,6].

Xuan presented a semi-supervised approach for automatic bug triage using text classification. This approach combined the Naive Bayes classification approach and expectation maximization to take the advantage of both labeled and unlabeled bug reports. He trained a classifier with a fraction of labeled bug reports. This labeled numerous unlabeled bug reports. From the result of, this semi-supervised approach improves the classification accuracy of bug triage by up to 6% and it avoids low-quality bugs [7].

When a bug report has been assigned to a specific developer, then if the assigned developer is unable to fix the bug, the assigned developers can forward or reassign the bug to other developer. This process of reassignment of bug from developer to developer is called "Bug Tossing" [8]. In addition, a model of bug tossing is built to reduce the number of reassignment of bug reports. The Markov chains based tossing graph approach was proposed to capture the bug tossing history which improved the bug assignment and reduced unnecessary tossing steps.

Instance selection and feature selection are commonly used techniques in data processing. For a given data set, instance selection is to obtain a subset of relevant instances (i.e., bug reports in bug data) [9] while feature selection expects to obtain a subset of relevant features (i.e., words in bug data) [10].

Set of machine learning tools and a probabilistic graph-based model (bug tossing graphs) that are highly-accurate predictions, and laid the foundation for the next generation of machine learning-based bug assignment. They used methodology like choosing effective classifiers and features, incremental learning, multi-featured tossing graphs to achieve their goal [11].

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

This technique models the reassignment of bugs as Enriched Adaptive Bug Triaging System (EABTS) which was based on actual path model. The graph structure captured the relationship among developers as the number of tosses and also captures the propinquity exists among developers. Therefore, this graph structure was enriched. The technique was based on Ant routing. Ant routing is inherently adaptive in nature. Their work gave a sub graph that consists of developers who were frequently involved in bug resolution [12].

To investigate the quality of bug data, design questionnaires to developers and users in three open source projects. Based on the analysis of questionnaires, they characterize what makes a good bug report and train a classifier to identify whether the quality of a bug report should be improved [17]. Duplicate bug reports weaken the quality of bug data by delaying the cost of handling bugs. To detect duplicate bug reports, they design a natural language processing approach by matching the execution information [18].

The existing systems machine learning techniques are ineffective for large project. Therefore P. Bhattacharya et al. [19] proposed a method for bug triage. Goal of this paper is to find the optimal set of machine learning techniques to improve bug assignment accuracy in large projects. This paper used a comprehensive set of machine learning tools as well as a probabilistic graph-based model (bug tossing graphs) that lead to highly-accurate predictions, and laid the foundation for the next generation of machine learning-based bug assignment. They used methodology like Choosing effective classifiers and features, Incremental learning, Multi-featured tossing graphs to achieve their goal.

III. PROBLEM DEFINATION

1. Software techniques suffer from the low quality of bug data. Due to the large number of daily bugs and the lack of expertise of all the bugs. Two typical characteristics of low-quality bugs are noise and redundancy.
2. The bug tracking system is used only to keep track of problem reports but not feature requests.
3. A technique to perform fine-grained analysis of source-code history but its combination with program analysis cannot support developers in answering the proposed questions for bug investigation.
4. Not 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.
5. The input minimization technique used to improving and faults of web application but does not track input parameters through the database.

IV. OBJECTIVE

1. 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.
2. And is automatically solved with mature text classification techniques, e.g., Random Forest, Based on the results of text classification, a human triager assigns new bugs by incorporating his/her expertise.

V. PROPOSED SYSTEM

Proposed work relies on study of data reduction on text classification and automatic assignment of bugs to an appropriate developer.

A. Data Reduction

Data reduction for bug triage system is combination of instance selection and feature selection to generate a reduced bug dataset.

Instance Selection can be defined as follows: Let X be an instance where X = (x1, x2,...,xM), with X belonging to a class c given by Xc, and an M-dimensional space in which xi is the value of the ith feature of the sample X. Then, let us assume that there is a training set T R which consists of N instances, and a test set T S composed of T instances. Let RS $\subseteq$ T R be the subset of selected samples that result from the execution of an IS algorithm; then, we classify a new pattern T from T S from a data mining algorithm acting over the instances of RS. We focus our efforts on enhancing the 1-NN rule. The reason for not employing a value k > 1 for the k-NN is to give the classifier the greatest possible sensitivity to noise during the reduction process. In this manner, an evolutionary IS algorithm can better detect the noisy instances and the redundant ones presented in the training set.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

Feature selection for each dataset, is computed by using the formula N/dM, where *N*, *d* and *M* are respectively the number of instances, the median of the features and the number of classes.

B. Bug Tracking System:

We propose to apply machine learning techniques to assist in bug triage to predict which developer should be assigned on the bug based on its description by applying text categorization, i.e. how the quality of bug data would be improved.

Algorithm:

Random Forest Algorithm

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation.

Random forests can be used to rank the importance of variables in a regression or classification problem in a natural way.

Input: node M having m1, m2, m3..... Variables

1. Grow a forest of many trees. (R default is 500)
2. Grow each tree on an independent bootstrap sample from the training data.
3. At each node:
   a. Select m variables at random out of all M possible variables (independently for each node).
   b. Find the best split on the selected m variables.
4. Grow the trees to maximum depth (classification).
5. Vote/average the trees to get predictions for new data.

VI. RESULT AND ANALYSIS

In the previous sections, Random Forest algorithm for detecting best developer for any bug is used. In this section, the performance (accuracy) of proposed approach is evaluated.

Data Collection: “Eclipse” dataset in csv (Comma-separated Values) format is uploaded and saved in bugtriagedataset excel sheet.

Pre-processing: In Eclipse dataset pre-processing has done. There are 334 bug reports.

The bugtriagedataset has 8 attributes named ID, Product, Comp, Assignee, Status, Resolution, Summary and Changed.

Data Reduction: Both feature selection and instance selection can reduce the scale of training set (for words or for bug reports) and feature selection can improve the accuracy rates.

Assignments of developer: This step takes efficiency, status (busy or available) and number of assigned bugs for calculating developer rank.

Efficiency = (resolved bug/total assigned bug)*100;

By getting the rank of developer the assignment of bug can be happened.

Accuracy: Comparative analysis is shown with graphical representation below.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

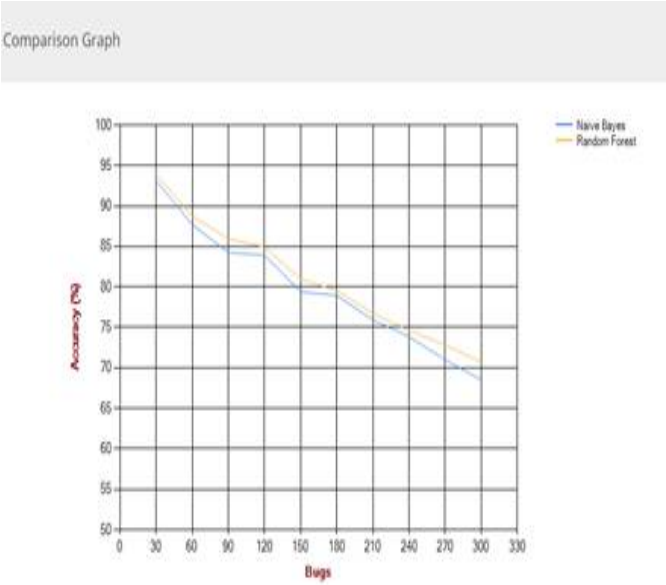

Figure 2. Accuracy of Random Forest and Naive Byes Algorithm

This represents the accuracy comparison for existing system (Naive Bayes) and proposed mechanism (Random Forest). he graph shows the accuracy for random forest is 70.96% for approx 300 bugs and Naive Bayes is 68.5% for the same.

Time Complexity: Time complexity for building a complete unpruned decision tree is O(v*nlog(n)), where n is the number of records and v is the number of variables/attributes.

Experimental Setup: The experimental setup has been implemented using frontend as ASP.net & MySQL as backend. Following are the execution environment such as Windows 7(64 bit), Intel core i5 processor (2.30GHz), Visual Studio 2012 and 4 GB RAM, so the result may vary depending upon Execution Environment.

VII. CONCLUSION

Data processing techniques like instance selection and feature selection are used for data reduction. This system can be used for any open source projects that generate huge bug data. The techniques of instance selection and feature selection are used to reduce noise and redundancy in bug data sets. The data reduction effectively reduces the bug data set into high quality bug data which can be used for effective bug triaging. The prediction order is determined by extracting the attributes of the bug data sets. Our system is based on Random Forest which provides an efficient approach on data processing to reduce the scale and provide high-quality bug data in software development and maintenance. We have focused on improving the accuracy of tracking system.

In future work, accuracy of the tracking system will be improved and another improvement that will be made is to consider the words which are written in abbreviated form or shorthand notation while writing a bug report. Mapping such words systematically with their actual forms and reasonably improves the triaging process.

REFERENCES

[1] Shreya Arudkar and Amit Pimpalkar, "Design of an Effective Mechanism for Automated Bug Triage System", International Journal of Research in Science & Engineering, Volume 3, Issue 1, January 2017.
[2] Shreya Arudkar and Amit Pimpalkar, "Implementation of an Efficient Bug Tracking System", MANTECH Publications Recent Trends in Computer Science and Software Technology, Volume 2, Issue 1, February 2017.
[3] Jifeng Xuan, He Jiang, Yan Hu, Zhilei Ren, Weiqin Zou, Zhongxuan Luo, and Xindong Wu, "Towards Effective Bug Triage with Software Data Reduction Techniques" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 1, JANUARY 2015.
[4] W. Zou, Y. Hu, J. Xuan, and H. Jiang, "Towards training set reduction

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 6, Special Issue 11, May 2017

for bug triage," in Proc. 35th Annu. IEEE Int. Comput. Soft. Appl. Conf., Jul. 2011.
[5] D. Cubrani and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Know l. Eng., Jun. 2004.
[6] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Softw. Eng., May 2006, pp. 361–370.
[7] J. Xuan, H. Jiang, Z. Ren, J. Yan, and Z. Luo, "Automatic bug triage using semi-supervised text classification," in Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng., Jul. 2010, pp. 209–214.
[8] G. Jeong, S. Kim, and T. Zimmermann, "Improving bug triage with tossing graphs," in Proc. Joint Meeting 12th Eur. Softw. Eng. Conf. 17th ACM SIGSOFT Symp. Found. Softw. Eng., Aug. 2009, pp. 111–120.
[9] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," Knowl. Inform. Syst., vol. 35, no. 2, pp. 249–283, 2013.
[10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.
[11] T. Zhang, G. Yang, B. Lee, I. Shin "Role Analysis-based Automatic Bug Triage Algorithm", 2012.
[12] V.Akila, Dr.G.Zayaraz, Dr.V.Govindasamy "Effective Bug Triage – A Framework", International Conference on Intelligent Computing, Communication & Convergence, 114 – 120, 2015 .
[13] S. Artzi, A. Kie _ zun, J. Dolby, F. Tip, D. Dig, A. Paradkar, and M. D. Ernst, "Finding bugs in web applications using dynamic test generation and explicit-state model checking," IEEE Softw., vol. 36, no. 4, pp. 474–494, Jul./Aug. 2010.
[14] 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.
[15] C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.
[16] K. Balog, L. Azzopardi, and M. de Rijke, "Formal models for expert finding in enterprise corpora," in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval, Aug. 2006, pp. 43–50.
[17] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schr€oter, and C. Weiss, "What makes a good bug report?", Oct. 2010, IEEE Trans. Softw. Eng.,
[18] X. Wang, L. Zhang, T. Xie, J. Anvik, and J. Sun, "An approach to detecting duplicate bug reports using natural language and execution information," in Proc. 30th Int. Conf. Softw. Eng., May 2008.
[19] P. Bhattacharya, L. Neamtiu, C. R. Shelton "Automated, highly-accurate, bug assignment using machine learning and tossing graphs" , 2012.
[20] Partha S. Bishnu and Vandana 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.
[21] 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.
[22] Sjifeng xuan, He jiang,yan hu, zhilei ren, weiqin zou, zhongxuan luo, and xindong wu "Towards Effective Bug Triage with Software Data Reduction Techniques" IEEE transactions on knowledge and data engineering, Vol. 27, No. 1, January 2015.
[23] Shay artzi, adam kie zun, julian dolby, frank tip, danny dig,amit paradka "Finding Bugs in Web Applications Using Dynamic Test Generation and Explicit-state Model Checking" IEEE transactions on knowledge and data engineering ,Vol.18, May.2010
[24] Vicente cerveron and francesc j. ferri "Another move toward the minimum consistent subset: A Tabu Search Approach to the Condensed Nearest Neighbor Rule". IEEE transactions on knowledge and data engineering , Vol. 22, No. 1, Feb. 2011
[25] Dominique matter, adrian kuhn, oscar nierstrasz" Assigning Bug Reports using Vocabulary -Based Expertise Model of Developers" IEEE transactions on knowledge and data engineering , Vol. 22, No. 1, Nov. 2012
[26] Ahmed lamkanfi, serge demeyer,emanuel giger , bart goethals" Predicting the Severity of a Reported Bug" IEEE transactions on knowledge and data engineering , IEEE transactions on knowledge and data engineering ,Vol.12,No.4, Nov.2011.
[27] Shivkumar shivaji ,E. James Whitehead, jr. ram akella ,sunghun kim" Reducing Features to Improve Code Change Based Bug Prediction" IEEE trans, cybern.Vol.31,No 5, May.2010.
[28] Thomas zimmermann ,nachiappan nagappan , brendan Murphy"Characterizing and Predicting which Bugs Get Reopened" IEEE transactions on knowledge and data engineering , IEEE transactions on knowledge and data engineering ,Vol.15,No.6, Aug.2011.
[29] R. e. walpole, r. h. myers, s. l. myers, and k. ye, x. wang, l. zhang, t. xie, j. anvik, and j. sun, "An Approach to Detecting Duplicate Bug Reports using Natural Language and Execution Information," in proc. 30th int. conf. softw. eng., May 2008.