IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

Review on Domain Specific Bug Triage with Software Data Reduction Techniques

Supriya B Boraste [1], P.B. Koli [2]

M.E Student, Kalyani Charitable Trust's Late G.N. Sapkal College of Engineering, Nasik, India [1]

Assistant Professor, Kalyani Charitable Trust's Late G.N. Sapkal College of Engineering, Nasik, India [2]

Abstract: Bugs are very essential aspects in a software company. The process of fixing bugs is called as a bug triage. Bug triage is an unavoidable step in a software company. In bug triage a correct developer is given to a new bug for fixing it. To manually perform the bug triage is very costly and even time consuming. So text classification techniques are used which uses automatic bug triage. There is a problem of large data i.e the data should be reduced and the quality of the data should be increased. To perform this instance selection and feature selection are used simultaneously. For this we should know the order for applying instance selection and feature selection, and to know the order we extract the attributes from the bug data sets. For the experiments we are using two open source projects such as eclipse and Mozilla. And our result shows that the data is reduced with high quality bug data sets.

Keywords: Mining Bug repositories, bug data reduction, attribute extraction, instance and feature selection.

I. INTRODUCTION

In current software expansion, software repositories are large databases for storing the output of software development. Repositories consist of source code, emails, bugs and specification. To manually perform the bug triage is very costly and even time consuming. bug triage. Software projects in a company consist of bug repositories which consist of bug data and it helps developers to handle bug. Updates according to the status of bug fixing. There are two challenges associated to bug data that may influence the effectual use of bug repositories they are huge scale and the low quality of data. Two typical characteristics of low-quality bugs are noise and redundancy. Both of these characteristics affect the bug triage process. So in this paper the two major issues are the large data and low quality. This two issue need to be solved to facilitate the bug handling process. In our work, we combine existing techniques of instance selection and feature selection to simultaneously reduce the bug dimension and the word dimension which improves the quality of the bug data.

II. LITERATURE SURVEY

1)"Bug report networks: Varieties, strategies, and impacts in an F/OSS development community," R. J. Sandusky, L. Gasser, and G. Ripoche, May 2004, To investigate the relationships in bug data, form a bug report network to examine the dependency among bug reports.

2) "Understanding a developer social network and its evolution," Q. Hong, S. Kim, S. C. Cheung, and C. Bird, Sep. 2011, Besides studying relationships among bug reports, build a developer social network to examine the collaboration among developers based on the bug data in Mozilla project. This developer social network is helpful to understand the developer community and the project evolution.

3) J. Xuan, H. Jiang, Z. Ren, and W. Zou, "Developer prioritization in bug repositories," 2012, in Proc. 34th Int. Conf. Softw. Eng., By mapping bug priorities to developers, identify the developer prioritization in open source bug repositories. The developer prioritization can distinguish developers and assist tasks in software maintenance.

4) 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., 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.

5)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, 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.

6) C. Sun, D. Lo, S. C. Khoo, and J. Jiang, "Towards more accurate retrieval of duplicate bug reports," in Proc. 26th IEEE/ACM Int. Conf. Automated Softw. Eng., 2011, propose a duplicate bug detection approach by optimizing a retrieval function on multiple features

7) 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, To improve the quality of bug reports, they have manually analyzed 600 bug reports in open source projects to seek for ignored information in bug data.

8) J. Xuan, H. Jiang, Z. Ren, and Z. Luo, "Solving the large scale next release problem with a backbone based multilevel algorithm," IEEE Trans. Softw. Eng , Sept./Oct. 2012. Based on the comparative analysis on the quality between bugs and requirements, they transfer bug data to requirements databases to supplement the lack of open data in requirements engineering.

9) 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., In contrast to existing work on studying the characteristics of data quality, our work can be utilized as a preprocessing technique for bug triage, which both improves data quality and reduces data scale.

10) D. _Cubrani_c and G. C. Murphy, "Automatic bug triage using text categorization," in Proc. 16th Int. Conf. Softw. Eng. Knowl. Eng., Jun. 2004, first propose the problem of automatic bug triage to reduce the cost of manual bug triage. They apply text classification techniques to predict related developers.

11) J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proc. 28th Int. Conf. Softw. Eng., May 2006, examine multiple techniques on bug triage, including data preparation and typical classifiers. Anvik and Murphy extend above work to reduce the effort of bug triage by creating development-oriented recommenders.

12) 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, find out that over 37 percent of bug reports have been reassigned in manual bug triage. They propose a tossing graph method to reduce reassignment in bug triage.

13) 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, To avoid low-quality bug reports in bug triage, they train a semi-supervised classifier by combining unlabeled bug reports with labeled ones.

14) 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, convert bug triage into an optimization problem and propose a collaborative filtering approach to reducing the bugfixing time.

15) H. Zhang, L. Gong, and S. Versteeg, "Predicting bug-fixing time: An empirical study of commercial software projects," in Proc. 35th Int. Conf. Softw. Eng.,May 2013, models the time cost of bug fixing and predicts the time cost of given bug reports;

16) E. Shihab, A. Ihara, Y. Kamei, W. M. Ibrahim, M. Ohira, B. Adams, A. E. Hassan, and K. Matsumoto, "Predicting re-opened bugs: A case study on the eclipse project," in Proc. 17th Working Conf. Reverse Eng., Oct. 2010, reopened-bug analysis , they identifies the incorrectly fixed bug reports to avoid delaying the software release.

17) T. M. Khoshgoftaar, K. Gao, and N. Seliya, "Attribute selection and imbalanced data: Problems in software defect prediction," in Proc. 22nd IEEE Int. Conf. Tools Artif. Intell., Oct. 2010, To improve the data quality, they examine the techniques on feature selection to handle imbalanced defect data.

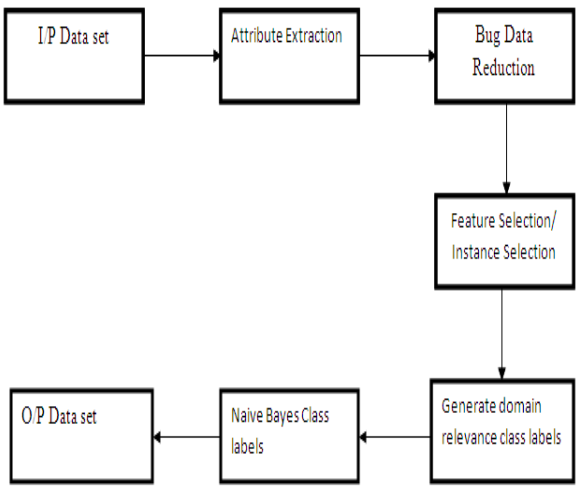
18) S. Shivaji, E. J. Whitehead, Jr., R. Akella, and S. Kim, "Reducing features to improve code change based bug prediction," IEEE Trans. Soft. Eng., vol. 39, no. 4, Apr. 2013, proposes a framework to examine multiple feature selection algorithms and remove noise features in classification-based defect prediction.

19) S. Kim, H. Zhang, R. Wu, and L. Gong, "Dealing with noise in defect prediction," in Proc. 32nd ACM/IEEE Int. Conf. Softw. Eng., May 2010, Besides feature selection in defect prediction, they present how to measure the noise eqresistance in defect prediction and how to detect noise data.

20) 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, process the defect data with quad tree based k-means clustering to assist defect prediction.

III.PROPOSED SYSTEM

To fix the bugs in an software company bug triage process is used. In this process correct developer is assigned to a new bug. But manual bug triage process is very time consuming and costly. So to avoid time cost, automatic bug triage in which text classification techniques are used. The problem which is addressed for this is the large bug dataset. And the large bug dataset affects the quality of the bug datasets. So to reduce the bug dataset we use feature selection and instance selection techniques as shown in block diagram. This techniques reduces the bug data in both bug and word dimensions. And even we want to know the order of applying the instance selection and feature selection for this the attributes of the historical bug datasets are extracted. This gives us the reduced and quality bug dataset. Even would like to built the domain specific system for which the domain relevance class labels are generated as shown in block diagram.



IV. CONCLUSION

In this paper we propose a approach that reduces the scale of the data and increases the quality of bug data by using instance selection and feature selection simultaneously. And even the prediction order is determined by extracting the attributes of the bug data sets. We perform the experiment of the data reduction for bug triage in bug repositories of two large open source projects such as Eclipse and Mozilla. Our work provides an technique on data processing which forms the reduced and high-quality bug data in software development and maintenance.

REFERENCES

[1] 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.
[2] 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.
[3] 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.
[4] C. C. Aggarwal and P. Zhao, "Towards graphical models for text processing," Knowl. Inform. Syst., vol. 36, no. 1, pp. 1–21, 2013.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] V. Cerver_on and F. J. Ferri, "Another move toward the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule," IEEE Trans. Syst., Man, Cybern., Part B, Cybern., vol. 31, no. 3, pp. 408–413, Jun. 2001.
[11] 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.
[12] B. Fitzgerald, "The transformation of open source software," MIS Quart., vol. 30, no. 3, pp. 587–598, Sep. 2006.
[13] A. K. Farahat, A. Ghodsi, M. S. Kamel, "Efficient greedy feature selection for unsupervised learning," Knowl. Inform. Syst., vol. 35, no. 2, pp. 285–310, May 2013.
[14] N. E. Fenton and S. L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, 2nd ed. Boston, MA, USA: PWS Publishing, 1998.
[15] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in Proc. 13th Int. Conf. Mach. Learn., Jul. 1996, pp. 148– 156.
[16] 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.
[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," J. Mach. Learn. Res., vol. 3, pp. 1157–1182, 2003.
[18] 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.

DOI 10.17148/IJARCCE.2015.412121