World Applied Sciences Journal 28 (5): 654-658, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.28.05.1383

Predicting Reliability from Frequent Errors Using A-priori Algorithm

[1]*Rajni Sehgal and* [2]*Deepti Mehrotra*

[1]Assistant Professor, AUUP, India
[2]Director ASCS, Noida, India

Abstract: With the increase in the complexity of software systems, the fault rate increases, which results in an exponential increase of errors and leads to decrease of reliability of the system. For predicting the reliability of any system all errors are recorded in the error logs. This is because predictable and frequently occurring errors which are of crucial interest from the huge error log, is of utmost important. In this paper it is attempt to show how error logs are being analyzed to predict reliability, using the A-priori algorithm

Key words: Error log · Frequent errors · A-Priori · Reliability

INTRODUCTION

There is undoubtedly a massive global effort to discover and develop the perfectly flawless system which will be free of errors and absolutely reliable. Systems are becoming more and more sophisticated hence requiring highly efficient diagnostic tools to keep them working at high efficiency by both acquiring as also analyzing complete information about the errors. Many existing error log recording Tools available in the market have been used and evaluated for their utility to produce different kinds of statistics. BUGZILLA is used for our preliminary work of study. Every error log has errors with certain attributes with which filtration is done in this study. Analysis of error log data dispenses vital information on all errors namely nature of fault, time of Occurrence of fault, status of fault, severity of fault, priority of fault and so on. Filtration of analyzed error logs is the next step for which A-priori algorithm is being used. This process is the key to the study of errors and their corrections. The process of filtration with this algorithm namely A-priori may also be called "Data Mining" which is explained as the process by which specific patterns hidden from the users in data bases can be uncovered. As the process of analysis and filtration continues relentlessly in the system, the efficient use of the data mining techniques with the A priori algorithm is the key to achieving reliability. With repeated trials in this study more efficient software in the future may be proposed. There are several other mining algorithms being used by developers.

However, this study is confined to the A Priori algorithm. This paper is divided in to 5 sections.

Section I is the Introduction where how error logs are recorded and mined to find out the frequently occurring errors to predict the reliability of the system is discussed.

Section II is about the related work where the work done by various authors in the field of mining to predict the software reliability is discussed.

In Section III a new model to predict the software reliability using the A-priori algorithm is discussed

In Section IV different tools and sample data set which has been used during this study is discussed

Section V is about the Results and Discussion where the frequent occurring errors from the error log with support 0.33 and confidence level 0.75 is given

Related Work: Every software which undergoes in maintenance phase, huge error logs are created. These error logs contain lots of information like title of the bug, ID of the bug, severity of the bug, priority of the bug and so on. These error logs can be suitably mined for the identification of the frequently occurring bug. Considerable work has been done in the area of prediction of software reliability. V.B. Singh, Krishna Kumar Chaturvedi [1] proposes a new tool named Bug Tracking and Reliability Assessment System (BTRAS) for the bug tracking/reporting and reliability assessment. In this study they have used BUGZILLA to record the bug and some classification. Techniques of data mining to report and fix the bug Ziming Zheng, llinois inst *et al.* present a log preprocessing method for failure detection which is based

Corresponding Author: Rajni Sehgal, Assistant Professor, AUUP, India.

on three steps (i) event categorization (ii) event filtering (iii) causality-related filtering to combine correlated events for filtering through A-Priori association rule mining [2]. Olivier Vandecruys *et al.* mining the software repositories with comprehensive data mining techniques to develop the fault prediction models to predict the faults [3]. R. K. Sahoo, A. J. Oliner, *et al.* describing proactive prediction and control system for large clusters. They collect the event log; filtering techniques are applied to model the data in to set of primary and derived variables [5]. J. L. Hellerstein, S. Ma and C. Perng, *et al* proposes an algorithm for reducing the burden of real time operation data like which sensor generates by mining historical data that are readily available [7]. H. Mannila, Hannu Toivonen and A. Inkeri Verkamo given a new algorithm for the discovery of all frequent episodes from a given class of episodes, which is very telecommunication alarm management [9]. Agrawal and R. Srikant propose a new algorithm that discovers the generalized sequential patterns by using the data mining techniques [10]. Ting-Ting Y. Lin, Daniel P. Siewiorek developed a new technique Dispersion Frame Technique (DFT) DFT can extract intermittent errors from the error log and uses only one fih of the error log entry points required by statistical methods for failure prediction [14]. Risto Vaarandi presents a novel clustering algorithm for log file data sets which helps one to detect frequent patterns from log files, to build log file profiles and to identify anomalous log file line [16].

There are so many techniques available of data mining to find out the frequently occurring errors from the error log but In this paper, we have used only A-Priori algorithm for identification of frequently occurring errors from error log. Many versions of A-Priori are available, but we have used A-Priori algorithm method

Proposed Work: In view of the background of related work on A-priori algorithm described above, the authors have been able to propose an extensive use of the A Priori algorithm for the prediction of software reliability for the reasons described above. It is an effective algorithm for data mining for specific types of error logs which are hampering reliability of systems on a large scale.

To briefly describe the study work being done by us, the following steps are being used in order to test the process.

Step1: Error comes from user during the Testing
Step2: Errors are recorded by the analyzing Tool i.e. BUGZILLA

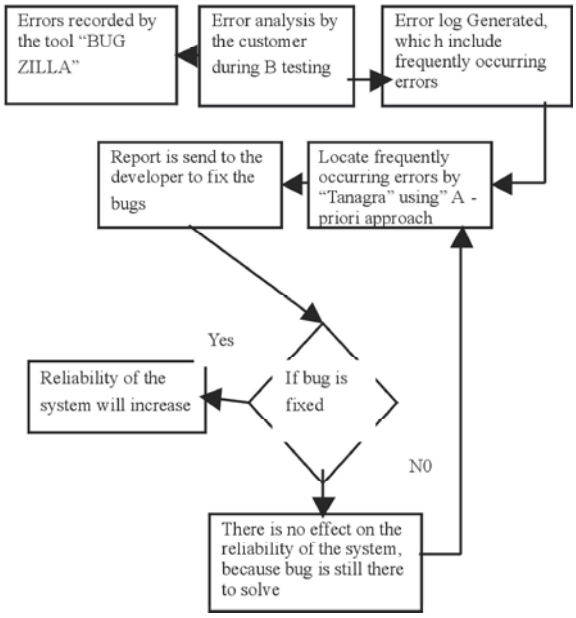

Fig. 1: Flow chart for predicting the errors from error log

Step 3: Recorded errors from the BUGZILLA tool analysis are called error logs which have the following attributes:-[1]

Title: Title of the Bug.

Description: Detailed description of the bug including time of occurrence, cause of occurrence, location of occurrence and type of occurrence.

Version: Specifies the version of the project.

Priority: Priority may be assigned based on level of severity.

Severity: Specify the threatened impact on the system

Status: Current Status of the bug (new, opened, confirmed, closed.)

Title: Title of the Bug.

Step4: Find the frequently occurring errors with the help of "Tanagra" tool using the A Priori algorithm approach.

Step5: The Report is sent to the developer who will attempt to fix the bugs. If successful the reliability of the system will be enhanced.

Table 1: Tools used during the study

Tools	
Bugzilla	Used to record the error which comes from users
Tanagra	Data mining tool used to find out frequently occurring errors.

Table 2: Sample data set created by BUGZILLA

Id	Sev	Pri	OS	Status	Res.
1798	nor	P2	All	CLOS	INVA
1799	nor	P2	All	CLOS	INVA
1800	nor	P2	All	CLOS	FIXE
1807	nor	P2	All	CLOS	FIXE
1809	nor	P2	All	CLOS	FIXE
1811	nor	P2	All	RESO	LATE
1812	nor	P2	All	CLOS	FIXE
1813	nor	P2	All	CLOS	FIXE
1814	nor	P2	All	CLOS	FIXE
1815	nor	P2	All	CLOS	FIXE
1816	nor	P2	All	CLOS	FIXE
1817	nor	P2	All	CLOS	FIXE
1818	nor	P2	All	CLOS	FIXE
1820	nor	P2	All	CLOS	FIXE
1821	nor	P2	All	CLOS	INVA
1822	nor	P3	All	CLOS	INVA
1823	nor	P3	All	CLOS	INVA

A-Priori Approach: All non-empty subsets of a frequent item set must also be frequent [4].

This is how frequent data sets are created

C_k = As a candidate item set of size k

f_k = As a frequent item set of size k

Main steps of iteration are Find frequent set f_{k-1}

- Join step: By Joining f_{k-1} is generated with itself (Cartesian product f_{k-1}Xf_{k-1}) [4].
- Prune step (A-priori property): Any (k - 1) size item set that is not frequent cannot be a subset of a Frequent k size item set, hence should be removed [4].

Frequent set f_k has been attained

RESULT AND DISCUSSIONS

Executing the error log in Tanagra tool

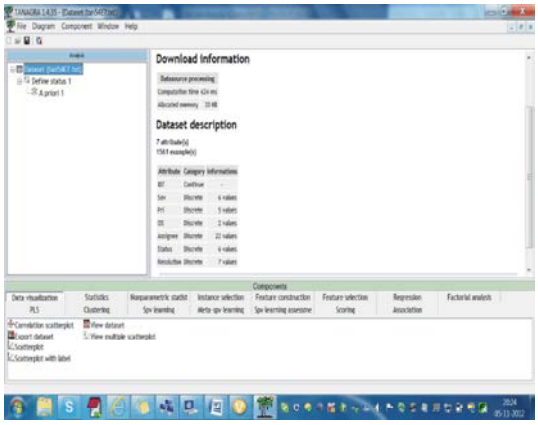

Fig. 2: Data set describes about the data of error log.

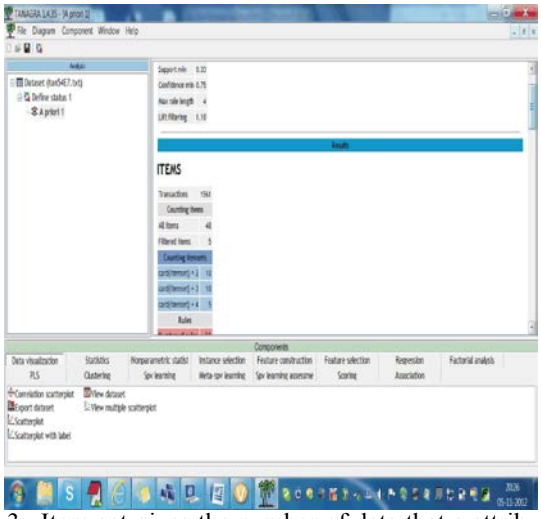

Fig. 3: Item set gives the number of data that a attribute contains i.e. severity has 6 items, priority has 5 items, status has 6 items, assignee has 22 items

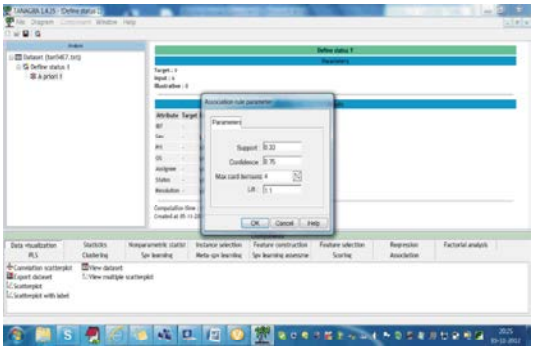

Fig. 4: Support count and confidence level "Support" indicates the frequencies of the occurring patterns in the rule; "confidence" denotes the strength of implications. For this error log we have taken support count 0.33 and confidence level 0.75.

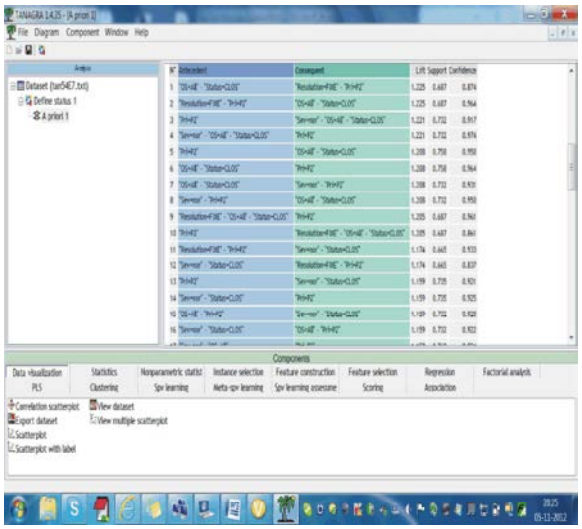

Fig. 5: Based on attributes of the dataset Tanagra tool has been locating frequent error log, above is the frequent errors which are occurring frequently

CONCLUSION

There are a lot of tools available for fixing and tracking the bugs during maintenance phase of the software, Identification of the frequent item set will help the developer to identify the most frequent bugs and fixed them on priority basis.Thus removal of these errors will enhance the reliability of the software.Threshold value of the number of bugs in frequent item set can be selected by giving the suitable value of support and confidence. If we get smaller frequent item set we can say reliability of the system has increased.

Future Scope: This work can be further enhanced by doing user profiling. On basis of their bug removal efficiency, we can identify the duplicate bugs and provided to the same developers who has already fixed the bug. Moreover Confusion Matrix can be drawn with the help similar kind error log

REFERENCES

1. Singh, V.B., 2011. Krishna Kumar Chaturvedi "Bug Tracking and Reliability Assessment System" (BTRAS), International Journal of Software Engineering and Its Applications.
2. Ziming Zheng, llinois inst *et al.*, 2009. System log pre-processing to improve failure prediction, IEEE Explore.
3. Olivier Vandecruys, David Martens *et al.*," 2008. Mining software repositories for comprehensive software fault prediction models", Elsevier.
4. Analysis of System Error Log Using Association Mining, UACEE International Journal of Computer Science and its Applications.
5. Sahoo, R.K., A.J. Oliner, *et al.*, 2003. "Critical Event Prediction for Proactive Management in Large scale Computer Clusters". In Proc. of SIGKDD.
6. Agrawal, R., T. Imielinski and A. Swami, 1993. "Mining Association Rules Between Sets of Items in Large Databases," In Proc. of SIGMODE.
7. Hellerstein, J.L., S. Ma and C. Perng, *et al.*, 2002. "Discovering actionablepatterns in event data". In IBM Systems Journal, 41(3).
8. Gujrati, P., Y. Li, Z. Lan, *et al.,* 2007. "A Meta-Learning Failure Predictor for Blue Gene/L Systems". In Proc. of ICPP.
9. Mannila, H. ,Hannu Toivonen and A. Inkeri Verkamo, 1997. "Discovery of Frequent Episodes in Event Sequences". Data Min. Knowl. Discov. 1(3): 259-289.
10. Agrawal and R. Srikant, 1995. "Mining Sequential Patterns", In Proc. ICDE,
11. Tzvetkov, P., X. Yan and J. Han, 2003. "Tsp: Mining top-k closed sequential patterns". In Proc. ICDM.
12. Yan, X., J. Han and R. Afshar "Clospan: Mining closed sequential patterns in large datasets". In Proc. SDM.
13. Vilalta, R. and S. Ma, 2002. "Predicting Rare Events in Temporal Domains", In Proc. of ICDM.
14. Ting-Ting, Y., Lin and Daniel P. Siewiorek, 1990. Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis, IEEE Transactions on reliability, VOL. 39, NO. 4, 1990 October
15. Risto Vaarandi, A Breadth-first algorithm for mining frequent patterns from event logs, IEEE International Conference, 2003-04.
16. Risto Vaarandi, 2003. "A Data Clustering Algorithm for Mining Patterns from Event Logs", Ieee International Conference,
17. Jiawei Han, Jian Pei and Yiwen Yin, 2000. "Mining Frequent Patterns without Candidate Generation", Proceedings of the ACM SIGMOD International Conference on Management of Data.
18. Seminar of Popular Algorithms in Data Mining and Machine Learning, TKK, 12.3.2008, Lauri Lahti.
19. Mining event logs with SLCT and Loghound,Risto Vaarandi, IEEE, 2008.

20. Cios, K.J., W. Pedrycz and R. Swiniarski, 1998. Data Mining Methods for Knowledge Discovery. Dordrecht, The Netherlands: Kluwer.
21. Agrawal, R. and R. Srikant, 1994. "Fast Algorithms for Mining Association Rules", in proceedings of the 20th international conference on Very Large Databases, pp: 487-499.
22. Inmon, W.H., 1996. "The data warehouse and data mining," Commun.ACM, 39: 49-50.