Amalgamation of Automated Testing and Data Mining ：A Novel Approach in Software Testing

Sarita Sharma[1], Anamika Sharma[2]

**[1] Computer Science Deptt., DAV Institute of Management
Faridabad, Haryana 121001, India**

**[1] Computer Science Deptt., DAV Institute of Management
Faridabad, Haryana 121001, India**

Abstract

Software engineering comprehends several disciplines devoted to prevent and remedy malfunctions and to warrant adequate behavior. Testing is a widespread validation approach in industry, but it is still largely ad hoc, expensive, and unpredictably effective. In today's industry, the design of software tests is mostly based on the testers' expertise, while test automation tools are limited to execution of pre-planned tests only. Evaluation of test outputs is also associated with a considerable effort by human testers who often have improper knowledge of the requirements specification. This manual approach to software testing results in heavy losses to the world's economy. This paper proposes the potential use of data mining algorithms for automated induction of functional requirements from execution data. The induced data mining models of tested software can be utilized for recovering missing and incomplete specifications, designing a minimal set of regression tests, and evaluating the correctness of software outputs when testing new, potentially inconsistent releases of the system.

Keywords: Automation, bagging, Data Mining

1. Introduction

As we know that creating a reliable and hence a fault free software is one of the major goals of a software developer. This makes software testing one of the most important and critical phases in the software development life cycle.

The process of software testing includes four phases [1] namely modeling the software environment, selection of test cases, running and evaluating test cases and measuring testing progress. As the test case automation is increasing the volume of the test is growing. It makes the selection of test cases very difficult, making the test case reduction highly desirable. After selecting and running the test cases the tester has to evaluate whether the selected test cases expose a fault or not. Traditionally this step was done manually by the human tester, which required a lot of time. As the software systems are growing larger the burden on the human tester is increasing. Using an automated oracle to support the activities of human tester can reduce the cost of the testing process and hence the related maintenance costs [2]. Data Mining is an analytic process designed to explore data (usually large amounts of data - typically business or market related) in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has the most direct business applications.

2. Crucial Concepts in Data Mining

The concept of bagging (voting for classification, averaging for regression-type problems with continuous dependent variables of interest) applies to the area of predictive data mining, to combine the predicted classifications (prediction) from multiple models, or from the same type of model for different learning data. It is also used to address the inherent instability of results when applying complex models to relatively small data sets. Suppose the data mining task is to build a model for predictive classification, and the dataset from which to train the model (learning data set, which contains observed classifications) is relatively small. You could repeatedly sub-sample (with replacement) from the dataset, and apply, for example, a tree classifier (e.g., C&RT and CHAID) to the successive samples. In practice, very different trees will often be grown for the different samples, illustrating the instability of models often evident with small data sets. One method of deriving a single prediction (for new observations) is to use all trees found in the different samples, and to apply some simple voting: The final classification is the one most often predicted by the different trees.

3. Automated Software Testing

Today, rigorous application testing is a critical part of virtually all software development projects. As more organizations develop mission – critical systems to support their business activities, the need is greatly increased for testing methods that support business objectives. It is necessary to ensure that these systems are reliable, built according to specification and have the ability to support business processes.

[3]Software testing using an automatic test program will generally avoid the errors that humans make when they get tired after multiple repetitions. The test program won't skip any tests by mistake. The test program can also record the results of the test accurately. The results can be automatically fed into a database that may provide useful statistics on how well the software development process is going. On the other hand, software that is tested manually will be tested with a randomness that helps find bugs in more varied situations. Since a software program usually won't vary each time it is run, it may not find some bugs that manual testing will. Automated software testing is never a complete substitute for manual testing.

Automation allows the testing organization to perform consistent and repeatable test. When applications need to be deployed across different hardware or software platforms, standard or benchmark tests can be created and repeated on target platforms to ensure that new platforms operate consistently.

3.1Automated testing Lifecycle

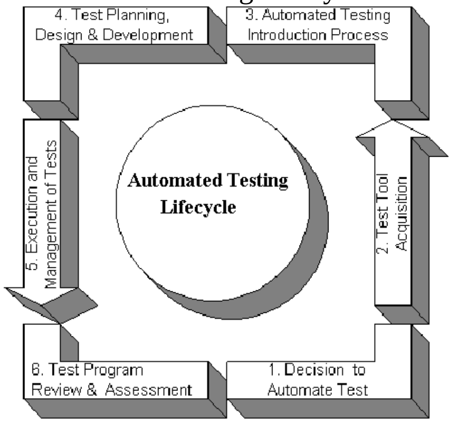

Fig 1. Automated Testing Lifecycle

The Automated Test Lifecycle comprises of six primary components:

- Decision to automate Testing
- Test Tool Acquisition
- Automated Testing introduction process
- Test Planning Design and Development
- Execution and Management of Tests
- Test Program Review and Assessment

3.2 Typical Testing Steps

Most software testing projects can be divided into general steps
a) Test Planning: This step determines like ‘which’ and ‘when’.
b) Test Design: This step determines how the tests should be built the level of quality.
c) Test Environment Preparation: Technical environment is established during this step.
d) Test Construction: At this step, test scripts are generated and test cases are developed.
e) Test Execution: This step is where the test scripts are executed according to the test plans.
f) Test evaluation: After the test is executed, the test results are compared to the expected
results and evaluations can be made about the quality of an application.

3.3 Challenges of Automated Testing

Though automated testing has a lot of benefit, but it also has some associated challenges. Following list includes some of the major challenges of automation [4]:

- Selection of Test Tool
- Customization of Tool
- Selection of Automation Level
- Development and Verification of Script
- Implementation of Test Management System

3.4 New Approach To Automatically Generate Test Cases

Data Mining algorithms can be efficiently used for automated modeling of tested systems. Induced Data Mining models can be utilized for recovering system requirements, identifying equivalence classes in system inputs, designing a minimal set of regression tests, and evaluating the correctness of software outputs.

For this a new approach to automatically generate test cases from SRS and mining of test cases has been discussed.

- A formal transformation of a detailed SRS to a UML state model
- The generation of test cases from the state model
- Mining of Test cases. The introduction of agents can bring enhancement

3.5 Need for effective test automation

The systematic production of high-quality software, which meets its specification, is still a major problem. Although formal specification methods have been around for a long time, only a few safety-critical domains justify the enormous effort of their application. The state of the practice, which relies on testing to force the quality into the product at the end of the development process, is also unsatisfactory. The need for effective test automation adds to this problem, because the creation and maintenance of the test ware is a source of inconsistency itself and is becoming a task of comparable complexity as the construction of the code.

3.6 The Approach

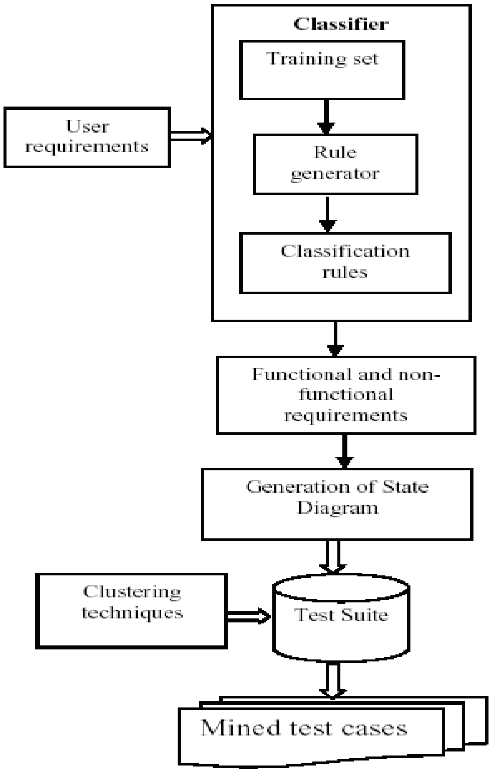

Fig 2. Automatically Generated Test Cases From Software Requirements Specification mining System

(1) Generation of classification rules.
(2) Generate test cases from the UML state machine.
(3) Finally data mining techniques are applied on the generated test cases in order to further reduce the test suite size.

3.7 Mining Techniques for Test Suite Reduction

Data mining is the process of extracting patterns from data. As more data are gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform these data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery.

While data mining can be used to uncover patterns in data samples, it is important to be aware that the use of non-representative samples of data may produce results that are not indicative of the domain. Similarly, data mining will not find patterns that may be present in the domain, if those patterns are not present in the sample being "mined". There is a tendency for insufficiently knowledgeable "consumers" of the results to attribute "magical abilities" to data mining, treating the technique as a sort of all-seeing crystal ball. Like any other tool, it only functions in conjunction with the appropriate raw material: in this case, indicative and representative data that the user must first collect. Further, the discovery of a particular pattern in a particular set of data does not necessarily mean that pattern is representative of the whole population from which that data was drawn. Hence, an important part of the process is the verification and validation of patterns on other samples of data. The term data mining has also been used in a related but negative sense, to mean the deliberate searching for apparent but not necessarily representative patterns in large numbers of data. To avoid confusion with the other sense, the terms data dredging and data snooping are often used. Note, however, that dredging and snooping can be (and sometimes are) used as exploratory tools when developing and clarifying hypotheses.

According to [5] , A program fails when it does not do what it is required to do .The purpose of testing a program is to discover faults that cause the system to fail rather than proving the program correctness . A successful test should reveal a problem in software; tests that do not expose any faults are useless, since they hardly provide any indication that the program works properly [6]. In developing a large system, the test of the entire application (system testing) is usually preceded by the stages of unit testing and integration testing . The activities of system testing include function testing, performance testing, acceptance testing, and installation testing Ideally, a minimal test suite can be generated from a complete and up-to-date specification of functional requirements. Unfortunately, frequent changes make the original requirements documentation, even if once complete and accurate, hardly relevant to the new versions of software [7]. To ensure effective design of new regression test cases, one has to recover (reverse engineer) the actual requirements of an existing system. In [8], several ways are proposed to determine input-output relationships in tested software. Thus, a tester can analyze system specifications, perform structural analysis of the system's source code, and observe the results of system execution. While available system specifications may be incomplete or outdated, especially in the case of a "legacy" application, and the code may be poorly structured, execution data seems to be the most reliable source of information on the actual functionality of an evolving system. In paper [9],the idea is extended initially introduced as input-output analysis of execution data can be automated by the IFN (Info-Fuzzy Network) methodology of data mining [11] [10]. In [9] the proposed concept of IFN-based testing has been demonstrated on individual discrete outputs of a small business program. The current study evaluates the effectiveness of the IFN methodology on a complex expert-system application having multiple continuous outputs.

4. RELATED STUDIES

The Paper [12] describes that Data Mining algorithms can be efficiently used for automated modeling of tested systems. Paper [13] proposes a way to process large amounts of data using a machine learning technique from association rule mining. [14] In this paper Case-Based Reasoning and Data mining are used as efficient methods for effort estimation and automated testing .

5. CONCLUSION

With this new proposed approach the system will automatically generate test cases from SRS and mining of test cases. Firstly a formal transformation of a detailed SRS to a UML state model, secondly the generation of test cases from the state model and lastly mining of Test cases. The introduction of agents can bring enhancement to the proposed Approach. Thus it will facilitate the mining and knowledge extraction from test cases and this technique can be utilized with automated software testing. This new approach gives a new direction to automated software testing using data mining and hence a novel approach in software testing.

6. FUTURE SCOPE

More new approaches can be researched to enhance the mining and knowledge extraction from test cases and thus making software testing automation more and more powerful.

REFERENCES

[1] J. Whittaker, "What Is Software Testing? And Why Is It So Hard?", IEEE Software,Vol. 17, No. 1, January/February 2000.

[2] M. Vanmali, M. Last and A. Kandel, "Using a Neural Network in Software Testing Process", International Journal of Intelligent Systems., Vol. 17, No. 1, pp. 45-62, January 2002.

[3] Dave Kelly ,"Software Test Automation and the Product Life Cycle", Symantec Corporation, Volume Number: 13 (1997),Issue Number: 10

[4] Hughes Software Systems Ltd. Test Automation, http://www.hssworld.com/whitepapers/whitepaper_pdf/test_automation.pdf, December 2002.

[5] Pfleeger, S.L. Software Engineering: Theory and Practice.2nd Edition, Prentice-Hall, 2001.

[6] Kaner, C., Falk, J., Nguyen, H.Q. Testing Computer Software. Wiley, 1999.

[7] El-Ramly, M. Stroulia, E. Sorenson, " From Run-time Behavior to Usage Scenarios: An Interaction-pattern Mining Approach.",Proceedings of KDD-2002 (Edmonton, Canada, July 2002), ACM Press, 315 – 327.

[8] Schroeder P. J., and Korel B, "Black-Box Test Reduction Using Input-Output Analysis", Proc. of ISSTA '00, 173-177, 2000.

[9] Last M., and Kandel, A.,"Automated Test Reduction Using an Info-Fuzzy Network. to appear in Annals of Software Engineering", Special Volume on Computational Intelligence in Software Engineering, 2003 .

[10] Last M., and Maimon, O." A Compact and Accurate Model for Classification",IEEE Transactions on Knowledge and Data Engineering, 2003.

[11] Maimon O., and Last, M.,"Knowledge Discovery and Data Mining – The Info-Fuzzy Network (IFN) Methodology",Kluwer Academic Publishers, Massive Computing, Boston, December 2000.

[12] Mark Last,"Data Mining For Software Testing",Ben-Gurion University of the Negev,2005

[13] E.M.H. Lee and K.C.C. Chan (PRC) , "A Data Mining Approach to Discover Temporal Relationship among Performance Metrics from Automated", http://www.actapress.com/Abstract.aspx?paperId=23837

[14] Ilkhani, A.; Abaee, G.; "Extraction test cases by using data mining; reducing the cost of testing ",Dept. of

Comput. Sci., Shahabdanesh Inst. of Higher Educ., Tehran, Iran , 8-10 Oct. 2010

Ms. Sarita Sharma did her Master in Computer Application from IGNOU, New Delhi, India, She did her M.phil (Computer Science) from Ch. Devi Lal University,Sirsa , India and is pusuing Ph.D from Singhania University,Rajasthan, India. She is presently working as Associate Professor in Deptt. Of Computer Science, DAV Institute of Management , India. She has guided more than 90 students in their Projects and has published a number of papers in National/International journals. She is a member of Computer Society of India. Her areas of interest include Software Engineering, Data Mining, Relational Databases, Computer Languages etc. She has about 15 years of teaching experience.

Ms. Anamika Sharma did her Master in computer Science from Gurukul University Haridwar in 1998 and pursuing Ph.D in computer Science from Singhania University Rajasthan. She is having about 13 years of teaching experience of postgraduate courses. She has guided more than 70 students in their project and published number of papers in national/International journals. She is a member of Computer Society of India .Her main Area of research include Computer Graphics, Data mining, Software testing and Quality assurance and object oriented Analysis and Design. At Present she is working in DAV Institute of Management, Faridabad as Associate Professor in Computer Science Dept.