The Journal of Systems and Software 97 (2014) 192–206

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Estimating confidence interval of software reliability with adaptive testing strategy

Junpeng Lv*, Bei-Bei Yin, Kai-Yuan Cai

School of Automation Science and Electrical Engineering, Beihang University, China

ARTICLE INFO

Article history:
Received 15 May 2013
Received in revised form 29 June 2014
Accepted 2 August 2014
Available online 11 August 2014

Keywords:
Adaptive testing
Confidence interval
Reliability assessment

ABSTRACT

Software reliability assessment is a critical problem in safety-critical and mission-critical systems. In the reliability assessment of such a system, both an accurate reliability estimate and a tight confidence interval are required. Adaptive testing (AT) is an on-line testing framework, which dynamically selects test cases from different subdomains to achieve some optimization object. Although AT has been proved effective in minimizing reliability estimator variance, its performance on providing the corresponding confidence interval has not been investigated. In order to address this issue, an AT strategy combined with Bayesian inference (AT-BI) is proposed in this study. The novel AT-BI strategy is expected to be effective in providing both a low-variance estimator and a tight confidence interval. Experiments are set up to validate the effectiveness of the AT-BI strategy.

1. Introduction

Computer hardware and software play an important role in daily life, especially in safety-critical and mission-critical systems, i.e., flight control systems, nuclear reactor plants, medical systems, and so on (Amin et al., 2013; Palviainen et al., 2011). Due to the complexity of modern software systems, it is usually difficult to guarantee the reliability of a software system. However, high reliability is usually required in safety-critical and mission-critical systems to minimize the probability of a catastrophic loss of life or property. Therefore, how to ensure the accuracy of the assessment result is a critical problem for the reliability assessment process of such a safety-critical or mission-critical system. Under such circumstances, not only a reliability estimate but also the corresponding *s*-confidence interval (CI) of the software system should be provided after the reliability assessment. Note that, in the reliability assessment process of a software system, the code of the software under test is frozen without removing detected defects, and this could be different from scenarios in which testing strategies are investigated to improve the reliability (Beizer, 1990; Whittaker, 1993). In fact, the assessment process is aimed at collecting data from the assessment process to calculate the reliability estimate and the corresponding CI (Pressman, 2004; Binder, 1999; Frankl et al., 1998), and the detected defects can be fixed after the assessment.

A "good" estimator should usually have four properties, i.e., unbiasedness, consistency, efficiency (e.g., minimum variance) and sufficiency (Pham, 2006). For a safety-critical or mission-critical system, an estimate that deviates from the true reliability value may lead to overconfidence in the reliability of the software system and cause an unexpected loss. Therefore, constructing an estimator with these four properties, especially with minimum variance, should be an optimization goal for the reliability assessment. In statistics, low variance can guarantee that the distribution of the estimation results, which are provided by the assessment processes with the same configuration, should be concentrated, that is, low variance ensures the predictability of the estimation results. Therefore, an unbiased reliability estimator with lower variance improves the efficiency of assessment, because fewer testing resources are required to get a more predictable result. Constructing such an estimator is very important for the reliability assessment process of a safety-critical or mission-critical system, in which the testing budgets are limited. However, a sole estimator is not sufficient to demonstrate the reliability of such an important system, because the estimation accuracy is not explicitly investigated. Since an estimate can seldom be the same as the true value, decision-makings based on such an estimate can be risky if how close the estimate is to the true value is unknown.

Under such circumstances, the corresponding CI determined by the chosen estimator should be taken into account. A CI indicates a clear interval of estimates as well as the confidence level at which the true reliability value will lie in this interval. Usually, a small-width CI is preferred since it means the true value will lie in a tight interval at some confidence level. In aircraft airworthiness

* Corresponding author. Tel.: +86 01082315087.
E-mail address: realljp@buaa.edu.cn (J. Lv).

http://dx.doi.org/10.1016/j.jss.2014.08.004
0164-1212/

certification, the failure rates/reliabilities of all aircraft with the same configurations are the major concern. However, there is usually not enough time and resources to test the system on every aircraft, since the sophisticated simulation environment built for testing flight control systems is usually shared by multiple projects. Thus, the assessment process of such a system is required to be as more dependable and efficient as possible. In this case, not only a predictable and efficient estimator but also a tight CI is required to provide accurate and dependable estimates without demanding additional testing resources.

In our previous work (Cai, 2002; Cai et al., 2004, 2005, 2008; Hu et al., 2013), adaptive testing (AT) is adopted to construct an efficient reliability estimator by minimizing the estimator variance. AT is a software testing technique that results from the application of feedback and adaptive control principles. It is an on-line and dynamic approach, which evolves along with the testing process. The theoretical foundation of AT is solid and the performance of AT is validated by simulation and experimental results. Since only reliability estimate is not sufficient to demonstrate how dependable the reliability assessment result is, the corresponding CI should be calculated based on the collected testing data. However, a tight CI is not considered as the optimization goal in previous AT strategy, and thus whether the AT strategy with a minimum-variance estimator can provide a tight CI cannot be theoretically investigated.

In order to address this issue, a new AT strategy combined with Bayesian inference (AT-BI) is proposed in this study. Bayesian theory (Bayes, 1763), which aims to provide a subjective confidence according to obtained evidence, is often adopted to calculate the corresponding CI based on collected testing data. The advantage of this approach focuses on the utilization of prior knowledge, e.g., prior software fault density/reliability knowledge. With such prior knowledge, the assessment process can be more efficient, which also meets the essence of AT. In this study, a subjective confidence of the reliability assessment is adopted to construct the optimization goal for AT-BI, and Bayesian inference theory is then applied to determine the optimal reliability assessment strategy. By doing so, the proposed AT-BI strategy can provide both an efficient estimator and a tight CI.

The remainder of this paper is organized as follows. Section 2 formulates the problem of reliability assessment with an operational profile and shows the calculation method for the CI. The AT-BI strategy is proposed in Section 3. Experiments are set up to evaluate the effectiveness of AT-BI in Section 4. General discussions are presented in Section 5. Conclusions and future work are listed in Section 6.

2. Problem formulation

The research studies on software reliability assessment can usually be divided into two branches: the continuous-time base (Chatterjee et al., 1997, 2011a,b) and the discrete-time base. The former focuses on the reliability behavior measured in terms of time, e.g., CPU execution time, which is appropriate for a wide scope of systems. However, as pointed out by Cai (2000), this assumption is not satisfied in many systems. For example, the reliability behavior of a bank transaction processing system should be measured in terms of how many transactions are successful, rather than of how long the software system operates without failures. In the reliability assessment of a flight control system, more attentions are paid to the take-off and landing processes rather than the cruising process, since more accidents happened in the former two processes. In this case, it is not very rational to model the system reliability in terms of time, because the types and proportions of applied maneuvers are quite different in these three processes. Therefore, a discrete-time and operational-profile based testing strategy is a good choice for the reliability assessment of such a system. This study investigates the discrete-time based reliability assessment, which is that same as what the previous AT studies have focused on.

In the final phase of software testing, the concerned software is subjected to testing for validation or acceptance, and reliability assessment is conducted based on the collected testing data. For some mission-critical or safety-critical software systems, observing few failures are acceptable in the reliability assessment process since the defects can be fixed after the assessment. On the other hand, the number of tests that can be applied in the assessment is usually limited because of the stringent schedule constraints. Under such circumstances, the code of software is frozen and no debugging is conducted during this phase even if software failures are observed. Although no modifications are applied to the software under test during the assessment, the detected defects should be fixed after the assessment. In this case, the central problem is how to apply a given number of tests such that the resulting software reliability estimate and CI are most dependable. Usually, the reliability estimate is provided by a reliability estimator, and the optimal case is that the constructed estimator has minimum variance. However, since the testing resources are limited and the failure rate is rather low, the number of observed failures can be few and the reliability estimate might not be quite accurate. In this case, a CI should be calculated to improve the confidence in the assessment results. Ideally, the estimator variance and the width of the corresponding CI should be minimized at the same time in order to provide the most accurate and dependable assessment result.

2.1. Assumptions

In order to formulate the problem of reliability assessment, a few assumptions are listed as follows.

(1) The software under test or reliability assessment is frozen. This assumption just means the code will not be modified during the assessment process, and the detected defects should be removed after the assessment.
(2) The input domain of the software under test comprises m subdomains of test cases, denoted as $\{D_1, D_2, \ldots, D_m\}$. This means the reliability assessment process is based on partition testing. In practice, the number of subdomains should be determined by the testers, that is, the test cases are generated and partitioned according to some criteria, e.g., functionality. One important thing is that the test cases in some subdomain should have some properties in common, and the partition should represent the operational usage of software under test in practice.
(3) Each test case will lead the software under test to failure or success, and

$$Pr\{observing failures by test cases from \quad D_i \quad \} \equiv \theta_i,$$

where $\theta_i \in (0, 1)$ and $i = 1, 2, \ldots, m$. Note that, this assumption requires a perfect test oracle to verify the system behavior. Since this paper focuses on measuring the performances of different reliability assessment strategies, the problem of choosing a perfect test oracle will not be discussed.
(4) The output of the software under test is independent of the testing history. There is possibility that one test is deemed failure-free, but actually it leads to some failure, which cannot be revealed due to our limited knowledge of test oracle. Under such circumstances, this faulty status may cause the following tests to fail even if these tests do not detect any defects, which will lead to some error in reliability estimation. However, this should be a problem of test oracle and this problem can be solved if additional correlated information is retrieved to enhance the test oracle.