Acceptance Criteria for Critical Software Based on Testability Estimates and Test Results

Antonia Bertolino
Istituto di Elaborazione della Informazione, CNR
Pisa, Italy

Lorenzo Strigini
Centre for Software Reliability, City University
London, U.K.

Abstract

Testability is defined as the probability that a program will fail a test, conditional on the program containing some fault. In this paper, we show that statements about the testability of a program can be more simply described in terms of assumptions on the probability distribution of the failure intensity of the program. We can thus state general acceptance conditions in clear mathematical terms using Bayesian inference. We develop two scenarios, one for software for which the reliability requirements are that the software must be completely fault-free, and another for requirements stated as an upper bound on the acceptable failure probability.

1. Introduction

The only direct method for predicting the operational reliability of a program is inference from "statistical" testing under an operational input profile [1, 2]. For safety-critical software, acceptance requires a long testing campaign with no failures. However, an amount of operational testing sufficient to warrant a high confidence that the software is as reliable as required in some current applications is infeasible [3, 4, 5].

A way forward is to combine the evidence from testing with any other evidence available. This combination can be made rigorous through Bayesian methods, in which the assessor can update the prior probability of an event, on the basis of new observed data, to produce a posterior probability, representing how the strength of belief allowable in the event taking place varies with new evidence. In particular, the use of prior probabilities in Bayesian reasoning explicitly describes the fact that predictions on the basis of statistical inference must also depend on pre-existing information about the events in question.

When judging on the basis of the results of testing, a kind of clearly helpful information is how effective the testing is at discovering faults. One would think that a series of successes in highly effective tests would give the same confidence as a longer series with less effective tests. A measure of test effectiveness that has gained some popularity is testability, the probability of a test detecting a failure conditional on the program being faulty, introduced by Voas and co-authors [6, 7, 8, 9, 10] and proposed as a basis for assessing software. The underlying intuition is that a statement about the internal structure of a program (to the effect that any bugs are likely to produce a high failure rate) allows one to draw stronger conclusions from testing than allowed by black-box considerations alone. In [11], we gave a rigorous, Bayesian inference procedure for obtaining the probability that a program is correct, knowing its testability, the test results, and the prior probability of it being correct. However, in that paper we used a point estimate of program
testability. This amounts to assuming that, if a program does contain faults, it is bound to have a certain, known probability of failure per execution (failure intensity), which is clearly a simplifying, but unrealistic assumption. In reality, we will instead have at most an understanding of which values of the failure intensity are more or less likely. In this paper, we offer two improvements:

1) we describe testability in terms of the prior distribution of the failure intensity of a program. This yields a prediction method which is more applicable in realistic situations, and eliminates the need to reason with a rather abstruse concept like the probability that a program would fail, if it were possible for it to fail;

2) we show how to use this prediction method when the criterion for accepting a program is either the probability that the program is correct (completely fault-free), or the probability that the program has an acceptable failure intensity in operation.

We consider a scenario in which software undergoes a long series of independent test cases, without failure. This is the typical case of interest for safety-critical software, but the mathematics can easily be extended to the case of any number of observed failures. For reasons of space, we only consider a few examples of simple prior distributions, to illustrate some essential facts about reasoning with testability. The other assumptions are that testing takes place with the operational input distribution, and all and only the actual failures of the program under test are detected (perfect oracle assumption).

In Section 2, we introduce the notion of failure intensity as a random variable and its probability distribution. Section 3 deals with the representation of assumptions about testability in terms of this distribution. Sections 4 and 5 describe the use of Bayesian inference in judgement about accepting software, according to the acceptance criteria 1) and 2), respectively, and illustrate the method with numerical examples. Section 6 summarises our results and their possible developments and discusses their practical uses.

2. Distributions of the Failure Intensity

In the assessment of software reliability, uncertainty derives from two sources: we do not know which inputs, if any, will cause the software to fail; and we do not know when and whether such inputs will be presented to the software in operation. It is reasonable to describe this uncertainty by stating that, under a given input profile, a program has a certain probability of failure when executed once, called a failure intensity (often called a "failure rate"). The failure intensity of a program is uncertain because of our limited knowledge about the program: we thus consider it as a random variable, $\Theta$, with a certain probability distribution. A way of picturing this is to think about the program to be assessed as having been "extracted at random" from the population of all the programs that could have been produced for the same purpose and under the same known conditions: they have different values of $\Theta$, and the distribution describes the frequencies with which different values of $\Theta$ appear in this population. Or, the distribution of $\Theta$ can be thought of as representing the degrees of one's beliefs ("subjective probabilities", based on whatever evidence is available) that different possible values are the actual value of $\Theta$ for this program. The latter is the Bayesian interpretation. With Bayesian inference, we represent what we expect about the program before testing it via a prior distribution: we can for instance take into account the reliability levels achieved in past products of the same development process. By applying Bayes' rule, we then obtain a (posterior)