Bayesian approach to reliability modelling for a probability of failure on demand parameter

BÖRCSÖK J., SCHAEFER S.
Department of Computer Architecture and System Programming
University Kassel,
Wilhelmshöher Allee 71, D-34121 Kassel
GERMANY
j.boercsoek@uni-kassel.de  schaefer@uni-kassel.de  www.rs.eecs.uni-kassel.de

Abstract: The testing of safety-critical software systems yields often a small amount of errors. The reliability analysis is therefore subject to restrictions, e.g. because of high variances. The use of Bayesian statistics incorporates additional data into the reliability analysis to overcome these restrictions. A model is developed that can be used for an analysis of software with low demand mode. The model is discussed and evaluated.

Key-Words: software reliability, bayesian reliability, reliability model, static analysis, PFD

1 Introduction

1.1 Motivation
In a normal software-development process it is necessary to calculate reliability metrics like failure rate, failure intensity, reliability or availability. For safety-related or safety-critical software it is mandatory to calculate these parameters. Otherwise the software cannot be released due to regulations. These calculations require data from the software-development process. The required data is obtained in the test process as failure data, i.e. the failure occurrences and times are recorded. The failure data is then used to compute the failure rate and the dependent reliability parameters using different models.

For safety-related software these parameters are based on scarce data, because the testing of safety-related software yields very few failures if the development process was carried out according to the safety standards. Therefore the calculated reliability parameters can have a high variance and are not trustworthy enough to be taken into account.

1.2 Requiring additional information
Consequently it is necessary to acquire additional information, to reduce the statistical variance and to make the reliability parameters more trustworthy. Extra data can come from different areas and phases of the software development process. This can include experience and parameters from past or similar projects or additional information directly gained from the source code of the software, e.g. through formal methods a correctness proof can be made. This is usually not practicable for whole software systems, because the proof methods work only under certain circumstances or with special restrictions [3]. The use of extra data from past or similar projects is also problematic, because small changes in the project can lead to a very different behaviour regarding reliability.

The additional information that is used in this paper is derived from static analysis. It is a procedure that can automatically analyse the source-code of the software, without actual execution, in regard to certain properties, e.g. security issues, deadlocks, memory leaks or value analysis.

Failures are the result of one or more errors in a program state. An error is usual generated from a variable that has an out of range value, e.g. the Ariane V launch failure was caused by an overflow of a program variable [5]. Value analysis tries to detect the possibility of these kinds of errors, so that variables can be made safe, i.e. an overflow is made impossible through programming techniques [1]. Not all program variables can be made safe in this regard, either because of performance issues, additional checks cost processing time, or because of the high costs of time and money.

The extra data that is used is therefore number of variables in the program that are safe, i.e. these variables can infer no errors and therefore no failures, and the number of variables that are unsafe,
i.e. these variables can potentially be the cause of an failure.

2 Reliability parameter
The reliability parameters that are regarded in this paper are dependent on discrete probability parameters. One of the most important reliability parameter according to the IEC 61508 standard [4] is the so called probability of failure on demand (PFD). It is defined as the mean probability to perform a designed function on demand. The PFD is used for components with a low demand mode. A function has a low demand mode, when it is used only scarce inside a system, i.e. it is executed for only a fraction of the time compared to the surrounding system. Therefore it is appropriate to model this parameter as the result of a discrete random process, in which the calculated parameter describes the probability the success or the failure of the function with low demand mode.

<table>
<thead>
<tr>
<th>Safety Integrity Level</th>
<th>PFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIL1</td>
<td>$10^{-1} - 10^{-2}$</td>
</tr>
<tr>
<td>SIL2</td>
<td>$10^{-2} - 10^{-3}$</td>
</tr>
<tr>
<td>SIL3</td>
<td>$10^{-3} - 10^{-4}$</td>
</tr>
<tr>
<td>SIL4</td>
<td>$10^{-4} - 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 1: SIL level with their according PFD values

3 Bayesian reliability
3.1 Basics
Adding extra information or a-priori information into a model is usually done with the use of Bayesian statistics. In reliability analysis this is called Bayesian reliability. The classic approach to reliability analysis produces parameters of a probability function with its expected values and its variance. A confidence interval has to be estimated for these parameters, i.e. the probability that the interval boundaries are trustworthy in regard to the parameters. The advantage of the Bayesian approach is that it gives the credible interval, i.e. the probability that the calculated reliability parameter lies in the interval. This gives additional certainty in the calculated values for the reliability parameters.

Bayesian reliability uses two sets of information: $\lambda$ describes the reliability parameters that have to be calculated. This is for example the failure rate, failure intensity or the probability of failure on demand. The set $D$ is the data that is available after testing. It consists usually of failure occurrences and times. These sets are used in the following relation:

$$g(\lambda | D) = \frac{f(D | \lambda)g(\lambda)}{\int_0^{\infty} f(D | \lambda)g(\lambda)d\lambda} \tag{1}$$

The reliability parameters for the model are calculated in the posteriori distribution $g(\lambda|D)$. It calculates the parameters with the highest probability given the recorded failure times. The additional information that is gained from the value analysis is incorporated into the model through the prior distribution $g(\lambda)$. The prior distribution is the initial estimate of the parameters before any failure times are collected. The probability that the given initial estimate of the reliability parameters can produce the collected data is calculated in the likelihood function $f(D | \lambda)$. The denominator of equation (1) is called the marginal distribution and it represents a normalizing factor for the posteriori distribution. A complete theoretical background of Bayesian reliability is given in [2].

4 Bayesian modelling
In Bayesian statistics it is necessary to specify a probability model. Only then the posterior distribution can be calculated. The model has then to be evaluated to confirm that the Bayesian model yields usable results. The model is based on the chosen likelihood function and the chosen prior function. The form of the prior distribution can be derived from the chosen likelihood, so that the resulting posterior distribution is in the same family of probability distributions. The model is then mathematical more convenient, if the model is producing sound results.

4.1 Modelling of PFD
For a PFD calculation software is analysed that has only discrete runs. A function that performs safety measures, e.g. emergency shutdown, is in low demand mode and therefore a PFD calculation can be applied. This is interpreted as the probability that this software function will fail if a safety measure has to be performed. Each demand to the software function can be seen as Bernoulli trial. A sum of $n$ runs is performed by the software function and the interesting parameter is the number of unsuccessful runs, i.e. the number of failures $k$. The Bayesian parameter $\lambda$ in equation (1) can then be interpreted as parameter of these Bernoulli trials. It is interpreted as the probability that an event occurs. In this case the event is defined as the inability to perform the intended function, i.e. a failure. The
number of events \( k \) that occur and the number of trials \( n \) represent the collected data \( D \).

A number of independent and identical Bernoulli trials are called a binomial distribution. The number of trials is the amount of runs \( n \). The collection of data \( D \) is done by testing or simulation of the software function \( n \) times. It will result in \( k \) failures of the software and \( n-k \) successful runs of the software. This gives a likelihood function in the form of a binomial distribution, with an unknown parameter \( \lambda \):

\[
f(k \mid \lambda) = \binom{n}{k} \lambda^k (1 - \lambda)^{n-k}
\]  

(2)

Figure 1 shows two binomial distributions with different estimations for \( \lambda \).

The unknown parameter \( \lambda \) can be estimated with the use of a prior distribution. The prior is chosen because of mathematical considerations and because it has to fit the underlying model. The model should also be useful when no prior information is available. The prior distribution should reflect that. For easy interpretation of the posterior parameters the prior distribution is chosen so that it belongs to the same family as the posterior distribution, it is then called a conjugate prior in respect to the likelihood. The beta distribution can be used as a conjugate prior for the binomial distributed likelihood function in equation (3):

\[
g(\lambda) = Be(a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \lambda^{a-1}(1-\lambda)^{b-1}
\]  

(3)

The parameters \( a \) and \( b \) are both shape parameters. The beta distribution is defined for \( 0 \leq \lambda \leq 1 \), which makes sense because the PFD is a probability. The beta distribution can take very different forms and is therefore very flexible and powerful in modelling. Figure 2 shows four beta distributions with different shape parameters.

![Fig. 2: Four examples of a beta distribution](image)

The most interesting cases are the first and the last forms, in relation to the legend, of the example figure. The first form has the parameters \( a=1 \) and \( b=1 \). It is a special case in which the beta distribution reduces to the uniform distribution. It can be interpreted as the complete absence of any prior information. The probability for every \( \lambda \) between zero and one is therefore equal, before any data is collected.

The example with the parameters \( a=2 \) and \( b=10 \) is useful because it has the desired form for the PFD, it is concentrated around small values for the PFD. If there is extra information before the data collection which can give the prior distribution this form, then the posterior distribution has a similar form. It is absolute necessary to check if the form of the prior distribution is sound in regard to the collected data.

Building the posterior distribution from the binomial likelihood and the beta prior yields a probability distribution of the following form:

\[
g(\lambda \mid a,b,k,n) = \frac{\binom{n}{k} \lambda^k (1 - \lambda)^{n-k} Be(a,b)}{\int_0^n \lambda^k (1 - \lambda)^{n-k} Be(a,b) d\lambda}
\]  

(4)

This is a beta distributed probability of the form:

\[
g(\lambda \mid a,b,k,n) = \frac{\Gamma(n+a+b)}{\Gamma(k+a)\Gamma(n+b-k)} \lambda^{k+a-1}(1-\lambda)^{n-b+k-1}
\]

\[
g(\lambda \mid a,b,k,n) = Be((k+a),(n+b-k))
\]  

(5)

The PFD depends on the two parameters of the prior distribution \( a \) and \( b \), and on the number of events \( k \).
and trials \( n \), i.e. failures that are collected through the testing or simulation phase.

4.1.1. Parameter interpretation
The interpretation of the parameter \( a \) and \( b \) of the prior distribution is as follows. If the beta distribution is used as conjugate prior, with the binomial likelihood, the parameter \( a \) and \( b \) describe the number of events of the Bernoulli trial. In this case \( a \) is the number of failures on demand and \( b \) is the number of successes on demand. The prior data is collected through static value analysis and is comprised of the number of overall variables in the software code and the number of safe or unsafe variables. The basic reasoning is, if there is no prior information, i.e. the number of safe variables is not known and all variables have to be regarded as unsafe variables, then the prior distribution has to reflect that absence of information in the form of a uniform distribution. The beta distribution is flexible enough to support this (see form one in figure 2) with the parameters \( a=1 \) and \( b=1 \). The argument for choosing the parameters is that with no prior information and with no data from actual runs no information about the successful performance of the software is possible. The successful run of a software demand has then the same probability as the failure of a software demand. This is reflected in the values of the parameters \( a \) and \( b \), which represent the number of failures and the number of successes. The values of \( a \) and \( b \) is 1, so that for every successful run there is an unsuccessful run.

The prior data has no information about the number of runs, because it is derived from static analysis, which is performed without actual running the software. Instead the relation of unsafe and safe variables is interpreted as relation of unsuccessful and successful runs.

The parameter \( a \) can then be interpreted as the number of unsuccessful runs normalised to 1 and \( b \) is the number of successful runs in relation to \( a \). This relation is a result of the relation of unsafe variables to overall variables, i.e. if half of the variables are considered safe this gives a relation of 1:2 and the values of the parameters of the prior distribution are \( a=1 \) and \( b=2 \).

This model makes a constraint in regard to the actual number of uses of the variables within the software. Different variables are used a different number of times. Some variables are used in every run and some are used not once in an actual run. Because of the mentioned constraint every variable has the same probability of being used. In an actual run of the software the number of variables that are executed are in direct relation to the number of unsafe and safe variables, if this relation is 1:2 then only half as much unsafe variables are used than safe variables. Therefore the prior probability for a failure on demand has the same relation and can be used in the prior distribution.

This model has several advantages. It is easy to use. The prior distribution is easy to calculate. The form of the prior distribution fits the actual data. In the special case of absence of information the results the results of the posterior distribution differ very little from the result if only the likelihood distribution is used to calculate the reliability parameter.

5 Evaluation of the model
The model has to be checked to test that the delivered results of the model have reasonable meaning. In the following section different case are assumed and the results are calculated and their fitness interpreted. The cases do not use real world data. Instead only synthetic data is used to check if the model theoretically produces meaningful results. Before the model is used in real world applications the model testing has to be extended with real world data.

5.1. Application
In this case it is assumed the analysed software is used in an environment that has a requirement for the safety integrity level (SIL) 2. It can be seen in table 1 that SIL2 requires a PFD reliability parameter between \( 10^{-2} \) and \( 10^{-3} \). It is assumed that the testing of the software is done 1000 times \((n=1000)\) and that 5 failure are counted \((k=5)\). If a binomial distribution for the PFD parameter is assumed that parameter can be calculated with a maximum likelihood estimation \( L(\lambda) \). The maximum likelihood estimation (MLE) maximises the function \( L(\lambda) \) in regard to the parameter \( \lambda \) and gives thereby the most likely value for \( \lambda \) given the recorded data:

\[
L(\lambda) = \prod_{i=1}^{n} f(x_i)
\]

\[
L(\lambda) = \binom{n}{x_i} \lambda^{x_i} (1 - \lambda)^{n-x_i}
\]

\[
\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{k}{n}
\]
Figure 3 shows the resulting ML estimation, with a maximum at $\lambda = 0.005$ which corresponds to the expected mode for this data $k/n = 0.005$. By adding different prior information it is evaluated, if the model yields meaningful results.

The first examined case is the absence of additional information, which gives a prior beta distribution of the form $Be_{1\_prior}(1,1)$. Then a value analysis is performed and the result gives relation of 1:20 of unsafe variables to overall variables. This relation is further improved with two additional cases with the relations of 1:50 and 1:100. The resulting prior distributions for these priors can easily be calculated with equation (5) and the given data $k = 5$ and $n = 1000$. The resulting posterior distributions are $Be_{1\_post}(6,996)$, $Be_{2\_post}(6,1015)$, $Be_{3\_post}(6,1045)$ and $Be_{4\_post}(6,1095)$ and are shown in figure 4.

5.1.1. Discussion

These posteriori distributions have the same form as the underlying likelihood function of figure 3. This should be the case, because the main source of the parameter estimation is still the collected data, which is represented by the likelihood function. The prior information should therefore not alter the form of the likelihood. It also should not skew or shift the likelihood too much. In that sense all of the above posterior distributions are suitable for the collected data. The advantage that the Bayesian model gives is that from the posterior distribution credible intervals can easily be obtained. The credible interval describes the probability that a calculated parameter lies in a certain interval. The example distributions have 95% credible intervals of $CI_{1\_post}(0.0022, 0.0116)$, $CI_{2\_post}(0.0022, 0.0114)$, $CI_{3\_post}(0.0021, 0.0111)$ and $CI_{4\_post}(0.002, 0.0106)$.

The parameter that was estimated directly from the likelihood function has an analogous interval the confidence interval, which has the problem that there are different methods to calculate the interval and the meaning of the interval is slightly different, compared to the credible interval. The confidence interval describes the probability of the interval boundaries in contrast to the probability of the reliability parameter. The 95% confidence interval of the example in figure 3 is $CI_{LF}(0.0006, 0.0094)$. Here the extra information makes a difference. The intervals for the posterior distribution are smaller than the interval in the likelihood function, which means the actual parameter is more trustworthy.

The reliability parameter itself is not much changed. This makes sense as the main source of information is the test data, and the prior information should not shift that parameter much. In comparison the calculated modes, the value with the highest probability, are $Mo_{1\_post}(0.005)$, $Mo_{2\_post}(0.0049)$, $Mo_{3\_post}(0.0048)$ and $Mo_{4\_post}(0.0045)$. For the posterior distribution without prior information the mode is the same as the mode of the likelihood function.

5.1.2. Special case

The development of safety critical software is done very thorough and according to a set of rules and standards. Therefore the software is often in a very mature state, when the data collection, the testing process, begins and it is possible that no failures occur. But the software cannot be regarded as free of errors, because that is an assumption that is too optimistic. The resulting distribution for the likelihood function has the following form:
The important statistical characteristics for the binomial distribution with the calculated parameter \( \lambda = 0 \) are the mode and the expected mean, which are both 0. The software has then to be interpreted as free of errors, which cannot be used in real projects. The posterior distribution for that case, with the same prior distributions as above, yield similar distribution forms:

The modes of all four distributions are also 0 and therefore not helpful. But for the expected means of all these distributions applies \( \lambda \neq 0 \). These distributions regard the software as not free of errors and are therefore more usable.

**6 Conclusion**

The described model demonstrates good prospects. The calculations give meaningful results that can be easily interpreted and used. The use of additional information increases the trust in the reliability calculations, which is also displayed in the mathematical results, e.g. the confidence or credible intervals.

An improvement is achieved for the special case that no data is collected. The traditional models fail to produce meaningful results in this case. The model used in this paper can be used to overcome this deficit.

**6 Future Work**

Additional models can be developed and evaluated, both for discrete data and for continuous data. It has to be examined if different prior distributions or likelihood functions yield better results. Data from real software projects has to be collected and applied to the model, to see if the promising results hold under real world conditions.

The number of uses of safe and unsafe variables has to be collected and has to be incorporated into the model to get a more accurate relation for the prior distribution. It has to be examined if additional prior information is easy to obtain and can be incorporated into this model to increase the usefulness of this model, especially for the special case, when no failure data could be collected.

**References**


