Chapter

4

Techniques for Prediction Analysis and Recalibration

Sarah Brocklehurst and Bev Littlewood
City University, London

4.1 Introduction

The previous chapter gives a comprehensive summary of many software reliability models that have appeared in the literature. Unfortunately, no single model has emerged that can be universally recommended to a potential user. In fact, the accuracy of the reliability measures arising from the models tends to vary quite dramatically: some models sometimes give good results, some models often perform inadequately, but no model can be trusted to be accurate in all circumstances. Worse than this, it does not seem possible to identify a priori those data sets for which a particular model will be appropriate [Abde86b].

This unsatisfactory position has undoubtedly been the major factor in the poor take-up of these techniques. Users who have experienced poor results adopt a once-bitten-twice-shy approach, and are unwilling to try new techniques. It is with some trepidation that we claim that the approach presented in this chapter has largely eliminated these difficulties. It might be as well, therefore, before giving some details of the techniques and examples of their use, to declare our credo. We believe that it *is* now possible *in most cases* to obtain *reasonably accurate* reliability measures for software and *to have reasonable confidence that this is the case* in a particular situation, as long as the reliability levels required are *relatively modest.* The italicized caveats here are important, because there are some limitations to what can currently be achieved, but they should not be so restrictive as to deter you from attempting to measure and predict software reliability in industrial contexts.

We begin by recalling briefly the nature of the software reliability problem. In the form in which it has been most studied, this is a problem of dynamic assessment and prediction of reliability in the presence of the reliability growth which stems from fault removal. A program is executing in a test (or real) operating environment, and attempts are made to fix faults when these are found as a result of the observation of software failures. There is therefore reliability growth, at least in the long term, although there may be local reversals as a result of poor fixes causing the introduction of new faults. The reliability growth models utilize the data collected here, usually in the form of successive execution times between failures (or, sometimes, numbers of failures in successive fixed time intervals; see Chap. 1 for details), to estimate the current reliability and predict the future development of the growth in reliability.

It is important to realize that all questions of practical interest involve *prediction.* Thus, even if we want to know the *current reliability* at a particular point in this process, we are asking a question about the *future:* in this case about the random variable, T, representing the time to the next failure. However we care to express our questions concerning the current reliability—as a rate of occurrence of failures, as a probability of surviving a specified mission time without failure, as a mean time to next failure, or in any other convenient way—we are attempting to predict the future. Longer-term prediction might involve attempting to estimate the (distribution of) time needed to achieve some target reliability, or the reliability that might be expected to be achieved after a certain duration of further testing.

The important point here is that when we ask, rather informally, whether a model is giving accurate reliability measures, we are really asking whether it is *predicting* accurately. This is something that is sometimes overlooked even in the technical literature; there are several examples of authors "validating" a model by showing that it can accurately explain past failure behavior and claiming thereby that it is "accurate." It is a simple matter to demonstrate that such ability to accurately capture the past does not necessarily imply an ability to predict accurately. The point is nicely expressed in a quotation of Niels Bohr, one of the greatest physicists of the 20th century: "Prediction is difficult, especially of the future."

4.2 Examples of Model Disagreement and Inaccuracy

4.2.1 Simple short-term predictions

Perhaps the simplest prediction arises when we ask what is the *current* reliability of a system. This will be a statement about the distribution of the time to the next failure and it could be expressed in several ways, as discussed earlier: the current *mean* or *median time to next failure* (MTTF), the *hazard rate function* of the distribution, or the *reliability function.*

It is as well to start rather informally with some examples that show how seriously the models can disagree in the answers they give to this most simple of all questions: namely, how reliable is the system now? We shall first show this disagreement between models, and then show how some of the results are also clearly *objectively* wrong.

Figure 4.1 shows plots of the successive current median times to next failure for a set of data, SYS1 from [Musa79], as calculated by some of the popular models: Jelinski-Moranda (JM) [Jeli72], Littlewood (LM) [Litt81], and Littlewood-Verrall (LV) [Litt73]. Thus, in this plot at stage j the predicted median of T_j is calculated for each model based upon all the data that has been observed prior to this stage, i.e., interfailure times, $t_1, t_2, \ldots, t_{j-1}$. We chose medians here for no particular reason

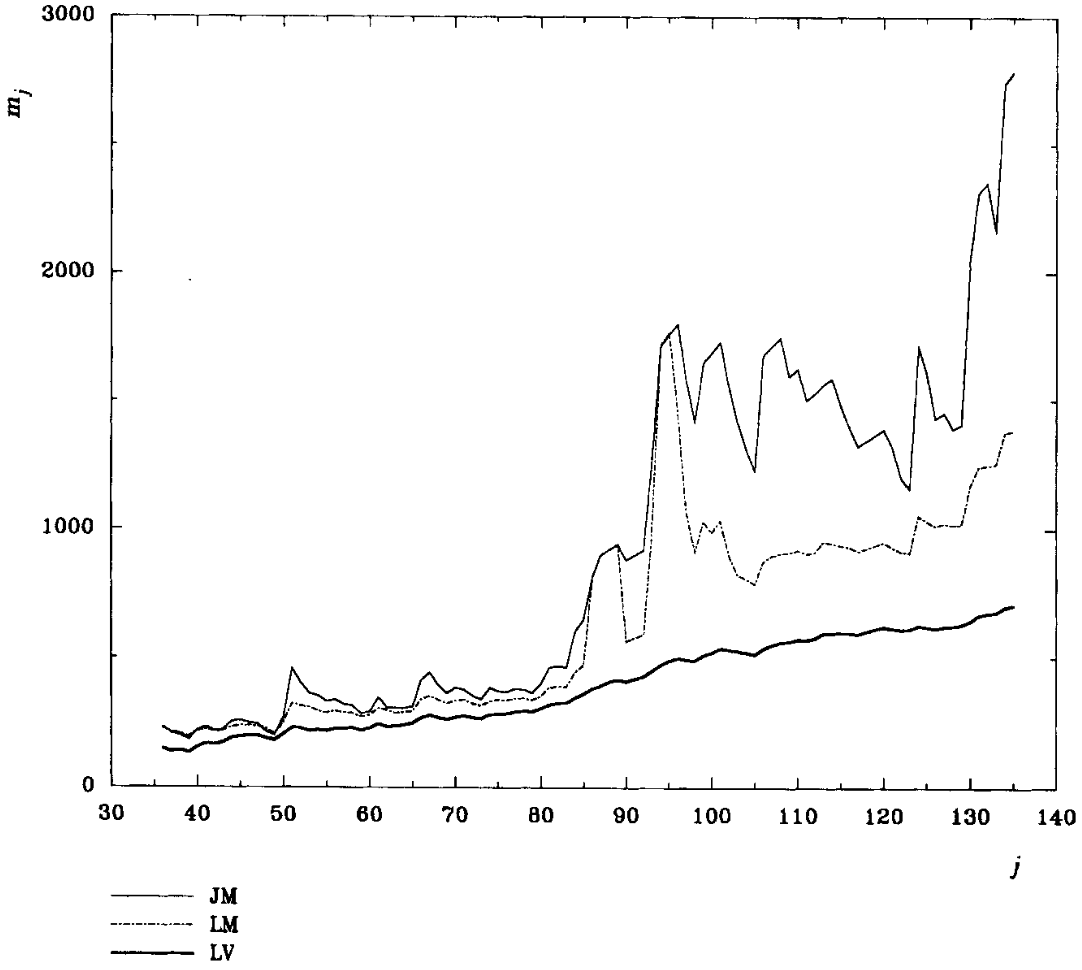


Figure 4.1 Successive one-step-ahead median predictions from models JM, LM, and LV of the time to next failure, T_j, plotted against j for $j = 36, \ldots, 136$, for data set SYS1. Notice the disagreement in these median predictions in the later part of the data.

other than convenience—the conclusions we draw will apply to other measures such as MTTF or *hazard rate.*

In the early stages of the plot there is reasonably close agreement between the three different models in how they predict the medians. This agreement disappears after about stage 85, when the medians begin to disagree. The first point to make, then, is that for this data the different models are giving quite markedly different numerical predictions in this very simple case of one-step-ahead median prediction. The fact of disagreement does not, of course, mean that all the predictions on this plot are inaccurate; on the contrary, it may be the case that one of the models is approximately correct and the other two are wrong (or even that some more complex reversals of fortune among the models are occurring as the data vector grows larger).

We can conduct quite crude investigations to examine this question of *absolute* accuracy. Figure 4.2 shows the actual data superimposed upon the median predictions of Fig. 4.1. At each stage j we can thus compare the three median predictions of T_j with the actual observed time to failure t_j. A very crude test of a certain type of accuracy would be to count the proportion of times the predicted median exceeded the later-observed time to failure. For accurate median predictions this proportion should be about one-half, and any significant departure from this would be evidence of some kind of bias in that sequence of predictions. If we look at the median predictions for JM, m_j^{JM}, we can see that from about $j = 90$ the proportion of times $m_j^{JM} > t_j$ is about 0.8. This suggests that the later predictions arising from the JM model are too large—i.e., in some sense the results from this model are too *optimistic* for this data set. Applying the same test to the LV predictions, there is evidence that the medians are too *pessimistic.* The LM model, on the other hand, passes this test quite well, with $m_j^{LM} > t_j$ about 57 percent of the time between stage 90 and the end of the data set.

Figure 4.3 shows a similar analysis using eight different models, JM, LM, and LV, as before, and Goel-Okumoto (GO) [Goel79], Musa-Okumoto (MO) [Musa84], Duane (DU) [Crow77, Duan64], Littlewood nonhomogeneous Poisson process (LNHPP) [Mill86], and Keiller-Littlewood (KL) [Keil83] models, on the data set SS3 from [Musa79], with once again the actual interfailure times superimposed. Again there is great disagreement between the eight different models, but, interestingly, they fall into two different groups of six and two models, respectively, with quite close agreement *within* each group. You might naively hope that the group of six models that agree with one another might be closer to the truth than the other pair. In fact, a comparison (as above) between the plots and the observed data shows that *none* of the eight models is getting close to the truth. The group of six are grossly optimistic in their median predictions, with a very high proportion of median predictions exceeding the later-observed times between failures; the other pair are grossly pessimistic.

4.2.2 Longer-term predictions

If these results were not discouraging enough, the problems of prediction inaccuracy become even more serious when we consider further-ahead prediction. In Fig. 4.4 we return to the SYS1 data to show median predictions 20 steps ahead, i.e., using only the data $t_1, t_2, \ldots, t_{j-20}$ to predict T_j. The performance of the JM model is extremely poor, with occasional excursions to infinity where it

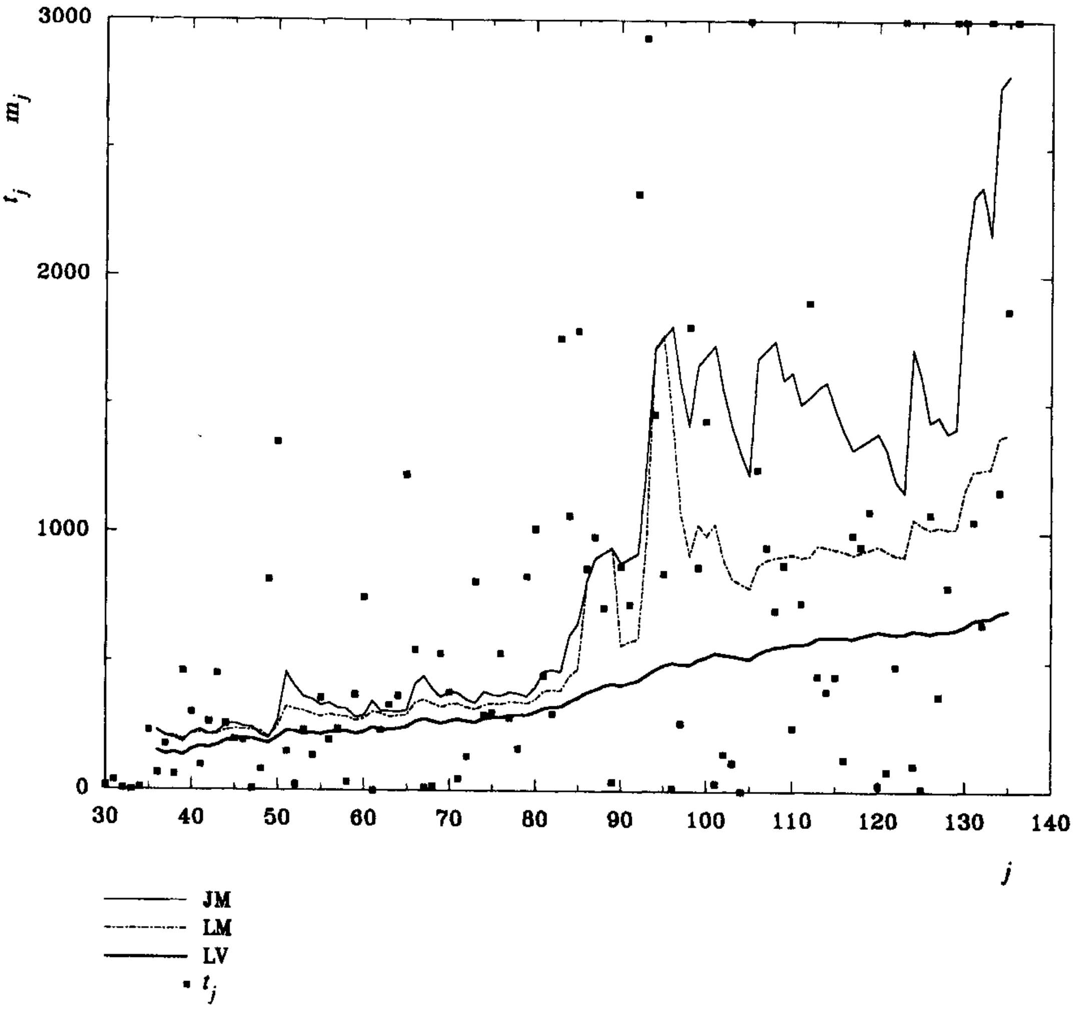


Figure 4.2 Observed times between failures, t_j, and successive one-step-ahead median predictions (as in Fig. 4.1) from models JM, LM, and LV of the time to next failure, T_j, plotted against j for $j = 36, \ldots, 136$, for data set SYS1. Comparison of the median predictions at each stage with the actual times between failures indicates the bias in these predictions; those from the JM model are too optimistic and those from the LV model too pessimistic, whereas those from the LM model would appear to be, on average, unbiased.

"believes" that the program under investigation will *never* fail again! This result can arise here because an intermediate parameter in this model is the number of remaining faults, which can be estimated to be zero. Clearly, the behavior shown by JM in this figure is very far from the truth. For example, each time the software is declared perfect it in fact disgraces itself by promptly failing again! Worse, the model does not even agree with the results that it produces itself when more data (the intervening 20 data points) are available (see Fig. 4.1).

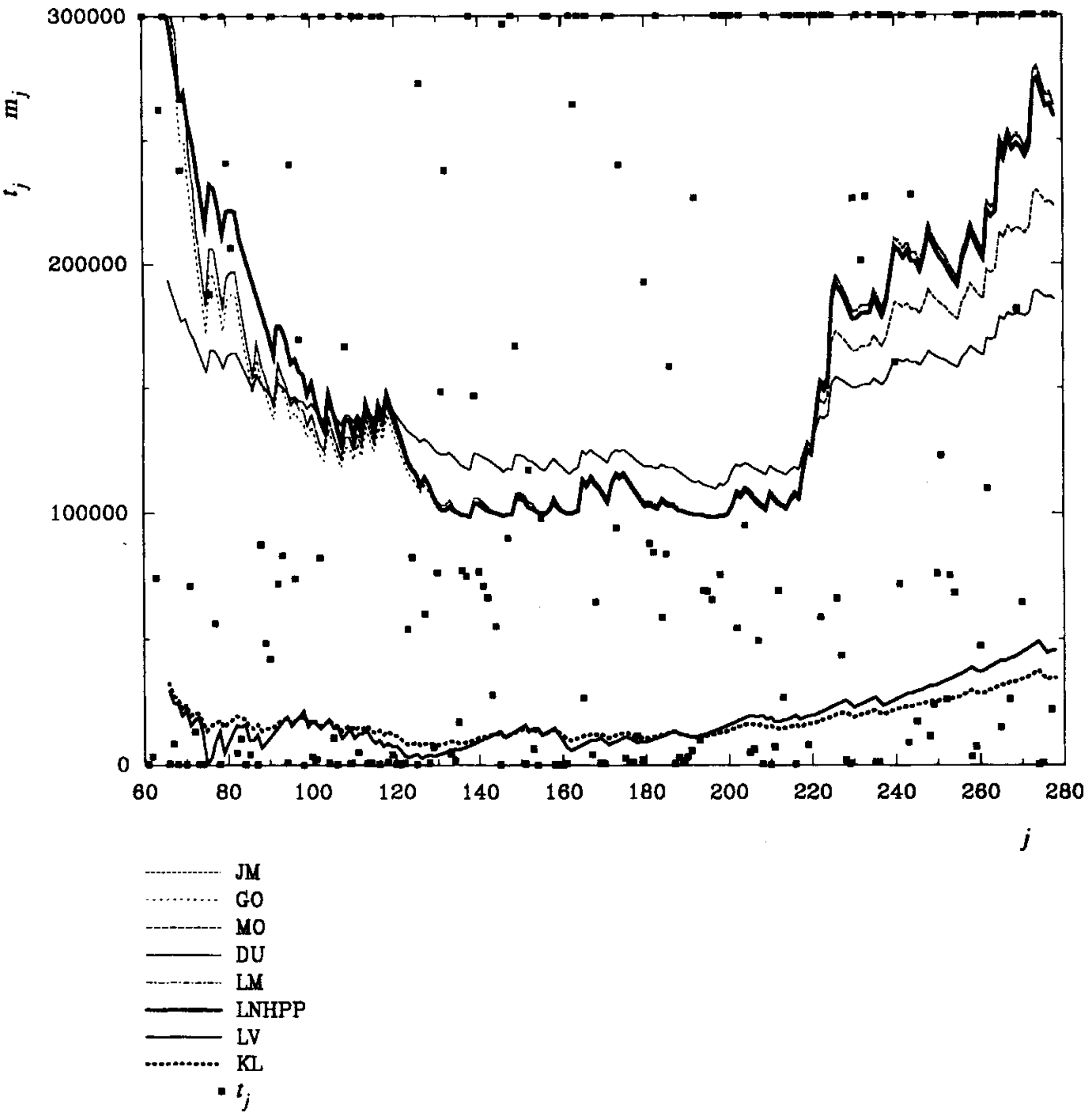


Figure 4.3 Observed times between failures, t_j, and successive one-step-ahead median predictions from eight models of the time to next failure, T_j, plotted against j for j = 66, . . . , 278, for data set SS3. These median predictions fall into two distinct groups, and the disagreement in the predictions for these two groups is great throughout the data set. The median test indicates that the predictions from LV and KL are grossly pessimistic, while the remaining median predictions are grossly optimistic.

The LV model performs better in this simple example of longer-term prediction. In Fig. 4.4 we show the 20-step-ahead median predictions with, for purposes of comparison, the one-step-ahead predictions of the same medians made at a later stage—these latter are the same as those shown earlier in Fig. 4.1. While we know from our earlier analysis that these are somewhat pessimistic, at least this model is exhibiting a reasonable self-consistency inasmuch as the predictions made earlier on a smaller data set are in good agreement with those made later. The JM model cannot even satisfy this minimal condition.

The results here are worrisome partly because they concern a very simple longer-term prediction. If we wished to predict much further into the future or to make more complex predictions of a different

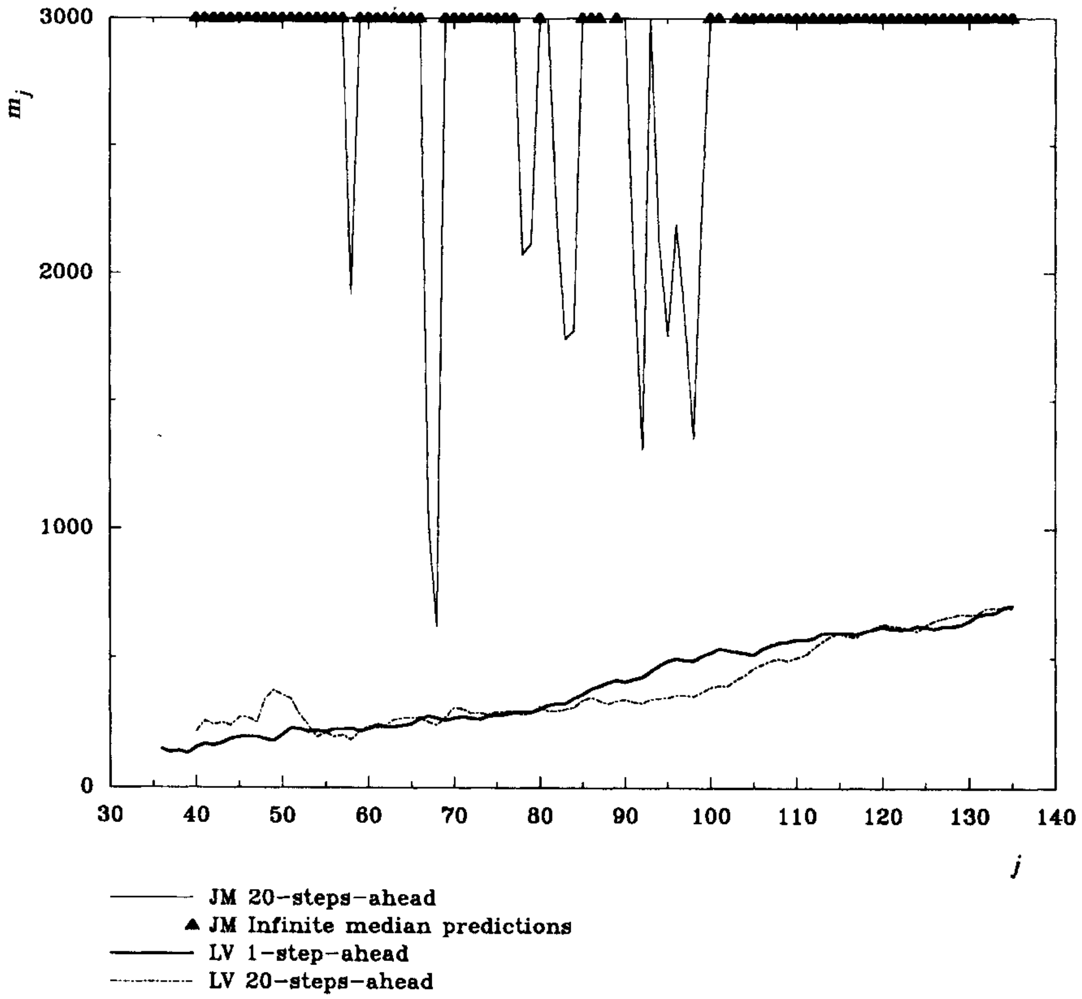


Figure 4.4 Successive 20-step-ahead median predictions from the JM model and successive one-step-ahead (as in Fig. 4.1) and 20-step-ahead median predictions from the LV model, of T_j, plotted against j for $j = 36, \ldots, 136$, for data set SYS1. The 20-step-ahead predictions from the JM are far too optimistic (compare with the times between failures in Fig. 4.2) and are much more optimistic than the one-step-ahead predictions from the same model (compare with Fig. 4.1). For LV, the one-step-ahead and 20-step-ahead median predictions are only marginally different.

nature (for example, predicting the time at which a prespecified reliability target will be reached), then it would be rash to assume that even *good* performance on some easier predictive task would allow us to conclude that other predictions would also be accurate. In fact, this observation can be made more generally: the fact that a model can give accurate predictions of one type for a particular data source does not allow us to conclude that predictions of a different type will also be accurate, even on the same data.

4.2.3 Model accuracy varies from data source to data source

Even in the results that we have shown already, which only concern two data sets, it is clear that the accuracy of some of the models varies considerably. For example, on the SYS1 data, the median predictions for the LM model pass the simple test of being in some sense "unbiased"—about 50 percent of the predicted medians exceed the later-observed times to failure, as should be the case—but on the SS3 data the same model gives grossly optimistic predictions.

The results of the more detailed analysis on the SYS1 and SS3 data sets in Sec. 4.3 will reveal even more serious differences between the behavior of a model on different data sources. In all the analyses that we have carried out over many data sets, we have found that the accuracy of the different models varies greatly from one data set to another [Abde86b].

It is certainly true that some models seem to be able to produce accurate predictions more consistently than others. The JM model, for example, appears to be fairly consistently *inaccurate,* and there are reasons for this in its unreasonable assumption that there are a finite number of faults contributing to the overall unreliability *and that these are all equal in their effect.* But even other, more plausible, models cannot be guaranteed to produce accurate results.

The conclusion seems to be that, even for a model which has given good results on a number of previous data sets, it would be unwise simply to *assume* that it will give good results on a novel data source.

4.2.4 Why we cannot select the best model a priori

Faced with this impasse—that models cannot be trusted to give accurate answers on all data sources—we might try to identify those characteristics of data sources that allow particular models to be accurate.

Unfortunately this does not seem to be possible. There is clearly great variation from one program to another: in the problem being solved, in the development practices, in the architecture, in the operational environment. No one has succeeded in identifying a priori those characteristics of a program that will ensure that a particular model can be trusted to produce accurate reliability predictions. In fact, this is not surprising, since the models involve rather crude assumptions about what may be a quite complex underlying failure process. There are many things that might impact upon the properties of the failure process that are simply ignored by the models. Examples include the nature of the operational environment, the internal fault-handling procedure (e.g., whether the software is fault-tolerant), etc. Such factors represent a source of uncontrolled variability in the properties of the failure process that is not treated by any of the models. In the absence of specific ways of taking account of such factors, we can expect the models to vary in their performance as the factors vary from one data source to another.

4.2.5 Discussion: a possible way forward

The results here are presented only to show that great disagreement between model predictions can and does exist and that we can show objectively that some very simple predictions can be extremely poor. It is worrisome that a model sometimes cannot even make one-step-ahead median predictions accurately, particularly if a potential user wishes to use the model for much more ambitious purposes. In fact, there is a sense in which results like this are only the tip of the iceberg. Models can go wrong in many different ways which might not be detected by the crude techniques used above. Even if a model were to pass our simple one-step-ahead median test, for example, this would not be a reason to trust its ability to produce accurate one-step-ahead predictions of a different nature—hazard rates, say, or reliability functions.

These observations show how important it is to devise more general ways in which the predictive accuracy of these models can be evaluated. We have shown that it is not possible to trust a particular model to give accurate results all the time or to select a model a priori that will give accurate results for a hitherto unseen data set. We therefore seem to have no alternative but to try to evaluate each model's predictive accuracy upon each new data set that is analyzed. The principle will be the same as the one that has been illustrated by the simple examples above: we must compare a prediction with the actual observation (when this is later made), and recursively build up a sequence of such prediction/observation comparisons. From this sequence we should be able to gain information about the accuracy of past predictions, and so make decisions about the current prediction (i.e., which model to trust, if any).

4.3 Methods of Analyzing Predictive Accuracy

4.3.1 Basic ideas: recursive comparison of predictions with eventual outcomes

Consider again, for simplicity, the simplest prediction problem of all: that of estimating the current reliability. Let us assume that we have observed the successive times between failures $t_1, t_2, \ldots, t_{j-1}$, and we want to predict the next time to failure T_j. We shall do this by using one of the models to obtain an estimate, $\hat{F}_j(t)$, of the true (but unknown) distribution function $F_j(t) \equiv P(T_j < t)$. Notice that if we knew the true distribution function then we could calculate any of the measures of current reliability, q_j, mean or median time to next failure or the rate of occurrence of failures (ROCOF), and so on, that may be appropriate for a particular application.

We now start the program running again, and wait until it next fails; this allows us to observe a realization t_j of the random variable T_j. We shall repeat this operation of *prediction* and *observation* for some range of values of j. In this way we can generate a *sequence*, $\hat{q}_j, j = s, \ldots, i$, say, of one-step-ahead predictions of interest. Table 4.1 shows an example of some times between failures and two prediction sequences, which we shall use to illustrate some of the techniques described in this chapter.

There are a number of ways suggested in the literature in which the accuracy of such a sequence of point predictions may be investigated. For example, the *variability* [Abde86b] may be examined,

$$\text{Variability}\{\hat{q}_j; j = s, \ldots, i\} = \sum_{j=s+1}^{i} \left| \frac{\hat{q}_j - \hat{q}_{j-1}}{\hat{q}_{j-1}} \right|$$

TABLE 4.1 Time Between Failures Data, $t_{12}, t_{13}, \ldots, t_{20}$ and Two Sequences, *A* and *B*, of Rate Predictions, $\hat{\lambda}_j^A$ and $\hat{\lambda}_j^B$, of T_j, $j = 12, \ldots, 20$

For illustrative purposes we shall assume that each prediction of T_j is based on previous data $t_1, \ldots, t_{j-1}$, and that the predictive distributions are exponential, with the predicted rates shown, for example, $\hat{F}_j(t) = 1 - e^{-\hat{\lambda}_j t}$ and $\hat{f}_j(t) = \hat{\lambda}_j e^{-\hat{\lambda}_j t}$. Predictions of mean time to failure $\hat{\text{MTTF}}_j = 1/\hat{\lambda}_j$ and median $\hat{m}_j = \ln(2)/\hat{\lambda}_j$ for these two prediction sequences are also shown.

j	t_j	$\hat{\lambda}_j^A$	$\hat{\text{MTTF}}_j^A$	$\hat{m}_j^A$	$\hat{\lambda}_j^B$	$\hat{\text{MTTF}}_j^B$	$\hat{m}_j^B$
12	105	0.010	100	69	0.0028	357	248
13	137	0.0077	130	90	0.023	43	30
14	125	0.0048	208	144	0.0071	141	98
15	161	0.0044	225	156	0.0020	500	347
16	162	0.0031	323	234	0.018	56	39
17	153	0.0029	345	239	0.0021	476	330
18	179	0.0028	357	248	0.0022	455	315
19	201	0.0017	603	418	0.0070	143	99
20	220	0.0013	769	533	0.0010	1000	693

This measure will detect whether a sequence of predictions is unduly *noisy*. Returning to our example earlier where we counted the proportion of the actual t_j exceeded by their predicted medians, $\hat{m}_j$, and asked if this proportion were very different from ½, it is clear that a sequence of predictions may pass this test, but still be very inaccurate. In other words, the predictions may *on average* be good but the *individual* median predictions may still be inaccurate. Using the above variability measure we might compare sequences of predictions from different models and reject one in favor of another on the basis that the predictions from the latter are more smooth, indicated by a smaller value of this variability measure. The obvious shortfall of such a measure is that reality may *itself* be noisy, and so we should not necessarily favor a predictor with a smaller variability measure.

It is clear that what we really need to examine, in order to assess the accuracy of a sequence of predictions, is the departure between the predictions and the truth. We would say, informally, that a model was giving good results if what we *observed* tended to be in close agreement with what we had earlier *predicted*. The approach we shall describe is based upon formal ways of comparing prediction with observation.

Of course, our problem would be easier if we could observe the true $F_j(t)$ so as to compare it with the prediction, $\hat{F}_j(t)$. Since this is not possible, we must somehow use the t_j, which is all the information that we have. Clearly this is not a simple problem, and it is compounded by its being nonstationary: we are interested in the accuracy of a sequence of *different* distributions, for each of which we see only one observation. However, it is possible to think of simple comparisons we can make such as the crude median test we discussed earlier. Other ways of comparing point predictions with observations are suggested in the literature. For example, in [Musa87], the *relative error* is used,

$$\text{Relative error} = \frac{\hat{\mu}(\tau_n) - n}{n}$$

Here, n failures have been observed by the total elapsed time τ_n, and $\hat{\mu}(\tau_n)$, the expected total number of failures by time τ_n, is a prediction made at an earlier time, $\tau(\tau < \tau_n)$, say. Plots of the relative error for different values of τ and for different values of τ_n may be examined, and these plots indicate the nature of the inaccuracy in the predictions of the expected number of failures, i.e., whether they are optimistic or pessimistic. They may also be used to *compare* two predictions or prediction systems, since the smaller the relative error the more accurate the predictions. Examination of the relative error, however, does not provide an *absolute* measure of accuracy of a single prediction system since tests to decide whether the error is significantly large do not exist.

A number of measures for which there are significance tests have been proposed. For example, the Braun statistic [Abde86b] may be used to see how accurate MTTF predictions are for those models for which the MTTF exists.

$$\text{Braun statistic}\{\widehat{\text{MTTF}}_j; j=s,\ldots,i\} = \frac{\sum\limits_{j=s}^{i}(t_j - \widehat{\text{MTTF}}_j)^2}{\sum\limits_{j=s}^{i}(t_j - \bar{t})^2}\left(\frac{i-s}{i-s-1}\right)$$

This statistic may also be calculated when the available data is *failure-count data* [Abde86a] as opposed to time-between-failures data. This is where the observations are numbers of failures, n_k, within successive time intervals, x_k, $k = 1, \ldots, r$, say.

$$\text{Braun statistic}\{\hat{E}[N_k]; k=s,\ldots,r\} = \frac{\sum\limits_{k=s}^{r}(n_k - \hat{E}[N_k])^2 x_k}{\sum\limits_{k=s}^{r}(n_k - \bar{n})^2 x_k}$$

Another statistic which can be used to investigate the accuracy of predictions of the expected number of failures within successive time intervals is the χ^2 statistic,

$$\chi^2 \text{ statistic}\{\hat{E}[N_k]; k=s,\ldots,r\} = \sum_{k=s}^{r}\left(\frac{(n_k - \hat{E}[N_k])^2}{\hat{E}[N_k]}\right)$$

Unfortunately the analyses of the accuracy of point predictions such as those described above, regardless of whether they provide absolute measures of accuracy or merely comparative measures of one prediction system versus another, do not tell us a great deal. Even if a series of point predictions was found to be accurate based on these various criteria we would only acquire confidence in these point predictions. This would not tell us whether other reliability measures were accurate. What we really need is to be able to detect *any* kind of departure between prediction, $\hat{F}_j(t)$, and truth, $F_j(t)$. We shall proceed with a discussion of various techniques which can be used to assess the accuracy of predictive *distributions,* $\hat{F}_j(t)$, by comparing them with the (later) observed times between failures, t_j. Although the discussion of these techniques in this chapter is limited to time-between-failures data, extensions to failure-count data exist.

4.3.2 The prequential likelihood ratio (PLR)

The first technique we consider is a very general means of *comparing* sequences of predictions for accuracy. It will show (at least asymptotically) which of a pair of predictions is most accurate in a very general sense. It does not, however, provide direct evidence of *absolute* accuracy.

An intuitive and informal explanation of the prequential likelihood ratio approach is shown in Fig. 4.5, where there are two ways, A and B, of making a prediction at stage j. Here we see the *true* distribution (in fact the probability density function, pdf, $f_j(t) \equiv F_j'(t)$) of the next time to failure, T_j, together with estimates of this (i.e., predictions, $\hat{f}_j^A(t)$ and $\hat{f}_j^B(t)$) coming from two different models, A and B. In practice, of course, we shall not be able to see the true distribution, which is unknown. If we could see it, as here, we might be able to decide readily which is the best of the two predictions: clearly, here A is better than B.

After making these two predictions, which are based only upon the data we have seen prior to stage j, we wait and eventually see the next failure occur after a time t_j. Since this is a realization of a random variable whose distribution is the true one, we would expect t_j to lie in the main body of this true distribution, as it does here: that is, it is more likely to occur where $f_j(t)$ is larger. If we evaluate the two predictive pdf's at this value of t, there will be a tendency for $\hat{f}_j^A(t_j)$ to be larger than $\hat{f}_j^B(t_j)$. This is because the A pdf tends to have more large values close to the large values of the true distribution than does the B pdf. In fact, this is what we mean when we say informally that "the A predictions are closer to the truth than the B predictions"—that the value of the A pdf tends to be everywhere closer to that of the true pdf than is the value of the B pdf.

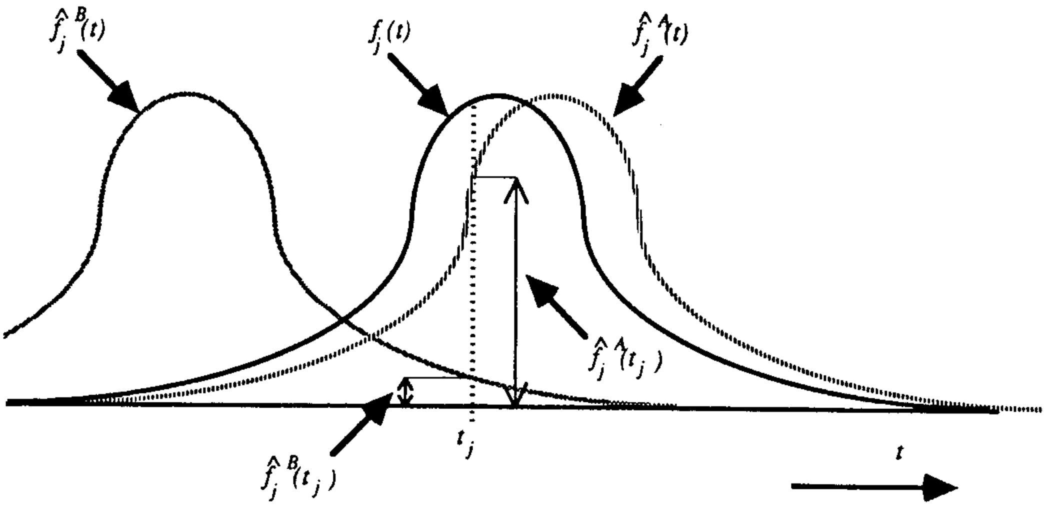


Figure 4.5 *True* predictive pdf, $f_j(t)$, of the next time to failure, T_j, together with estimates of this pdf, $\hat{f}_j^A(t_j)$ and $\hat{f}_j^B(t_j)$, from two models, A and B. A is clearly a better predictor of the truth than is B.

Thus if the predictions from *A* are more accurate than those from *B*, $\hat{f}_j^A(t_j)/\hat{f}_j^B(t_j)$ will tend to be larger than 1. The PLR is merely a running product of such terms over many successive predictions:

$$\text{PLR}_i^{AB} = \prod_{j=s}^{j=i} \frac{\hat{f}_j^A(t_j)}{\hat{f}_j^B(t_j)}$$

and this should tend to increase with *i* if the *A* predictions are better than the *B* predictions. Conversely, superiority of *B* over *A* will be indicated if this product shows a decreasing trend.

You should note that, even if *A* is performing consistently more accurately than *B*, we cannot guarantee that $\hat{f}_j^A(t_j)/\hat{f}_j^B(t_j)$ will always be greater than 1. Thus, typically in a case where *A* is better than *B*, we would expect the plot of PLR_i^{AB} (or, more usually for convenience, the *log* of this) to exhibit overall increase, but with some local random fluctuations. We are looking for consistent upward or downward *trend* in the PLR_i^{AB} as we make successive predictions.

Table 4.2 shows how to do this PLR analysis for the simple predictors, *A* and *B*, mentioned earlier in Sec. 4.3.1. The corresponding plot of the $\log(\text{PLR}^{AB})$ for model *A* versus model *B* is shown in Fig. 4.6. The fairly steady upward slope in this plot indicates that prediction sequence *A* is generally better than *B* over the range of predictions examined, although toward the end of this range, performance between the two prediction sequences would seem to be leveling out.

We are usually interested in comparing the accuracy of more than two sequences of predictions. To do this we select one, quite arbitrarily, as a reference and conduct pairwise comparisons of all others against this, as above. As an example, in Fig. 4.7 we show a PLR analysis of the SS3 data. Recall that, in Sec. 4.2.1, we saw an analysis of the one-step-ahead *median* predictions for this data, and established via a simple informal analysis that none of these models could be trusted to give accurate medians for this data. Nevertheless, six of

TABLE 4.2 $\hat{f}_j^A(t_j)$ and $\hat{f}_j^B(t_j)$ ($\hat{f}_j(t_j) = \hat{\lambda}_j e^{-\hat{\lambda}_j t_j}$) for Prediction Sequences *A* and *B* Shown in Table 4.1, Together with the $\log(\text{PLR}_i^{AB}) = \sum_{j=12}^{j=i} \log(\hat{f}_j^A(t_j)/\hat{f}_j^B(t_j))$ Evaluated for Prediction Sequence *A* versus Prediction Sequence *B*.

j	t_j	$\hat{f}_j^A(t_j)$	$\hat{f}_j^B(t_j)$	$\log(\hat{f}_j^A(t_j)/\hat{f}_j^B(t_j))$	$\log(PLR_i^{AB})$
12	105	0.00350	0.00209	0.516	0.516
13	137	0.00268	0.000985	1.00	1.52
14	125	0.00263	0.00292	−0.105	1.41
15	161	0.00236	0.00145	0.487	1.90
16	162	0.00187	0.000975	0.651	2.55
17	153	0.00186	0.00152	0.202	2.75
18	179	0.00170	0.00148	0.139	2.89
19	201	0.00121	0.00171	−0.346	2.54
20	220	0.000977	0.000803	0.196	2.74

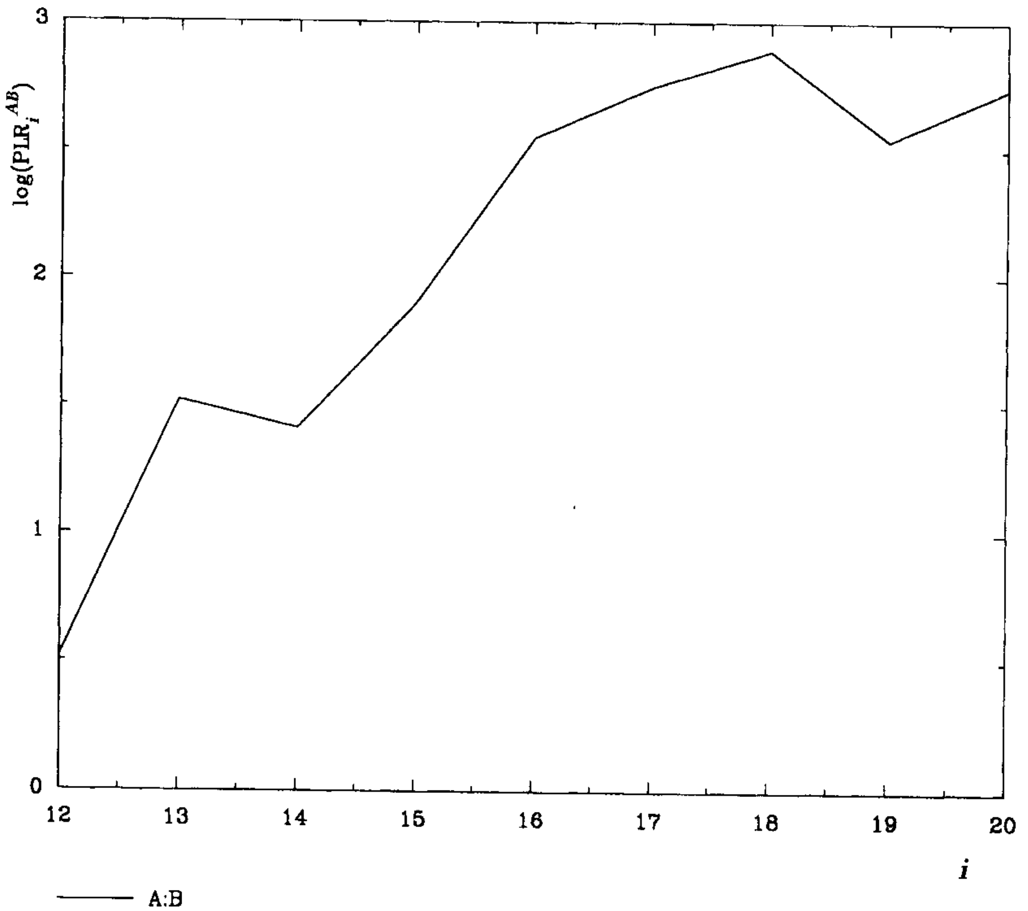


Figure 4.6 Log(PLR) plot for the prediction sequences shown in Table 4.1, that is, log(PLR$_i^{AB}$) versus i for prediction sequence A versus prediction sequence B as calculated in Table 4.2. The increase in the slope of this plot indicates that prediction sequence A is generally better than prediction sequence B.

the models gave median predictions that were in close agreement, and it might be thought that these would at least be *more* accurate than the other two. In fact this is not the case, and on the contrary the other two models (KL and LV) perform very much better on the PLR criterion.

In the figure, the DU model has been chosen as the reference model, so that all comparisons are pairwise with respect to this. It can be seen that for the LV and KL models the PLR plots against DU exhibit a clear upward trend (notice that the plots are of the log of PLR here), indicating their superiority over DU. The plots of the other models are similar to one another, exhibiting neither upward or downward trend and thus no superiority over DU. The evidence here, then, is that the six models that were in agreement on the earlier median plots are shown by the PLR analysis to be giving *general* one-step-ahead predictions that are of similar accuracy, and that this accuracy is much less than that given by LV and KL, which are themselves similar to one another in their accuracy.

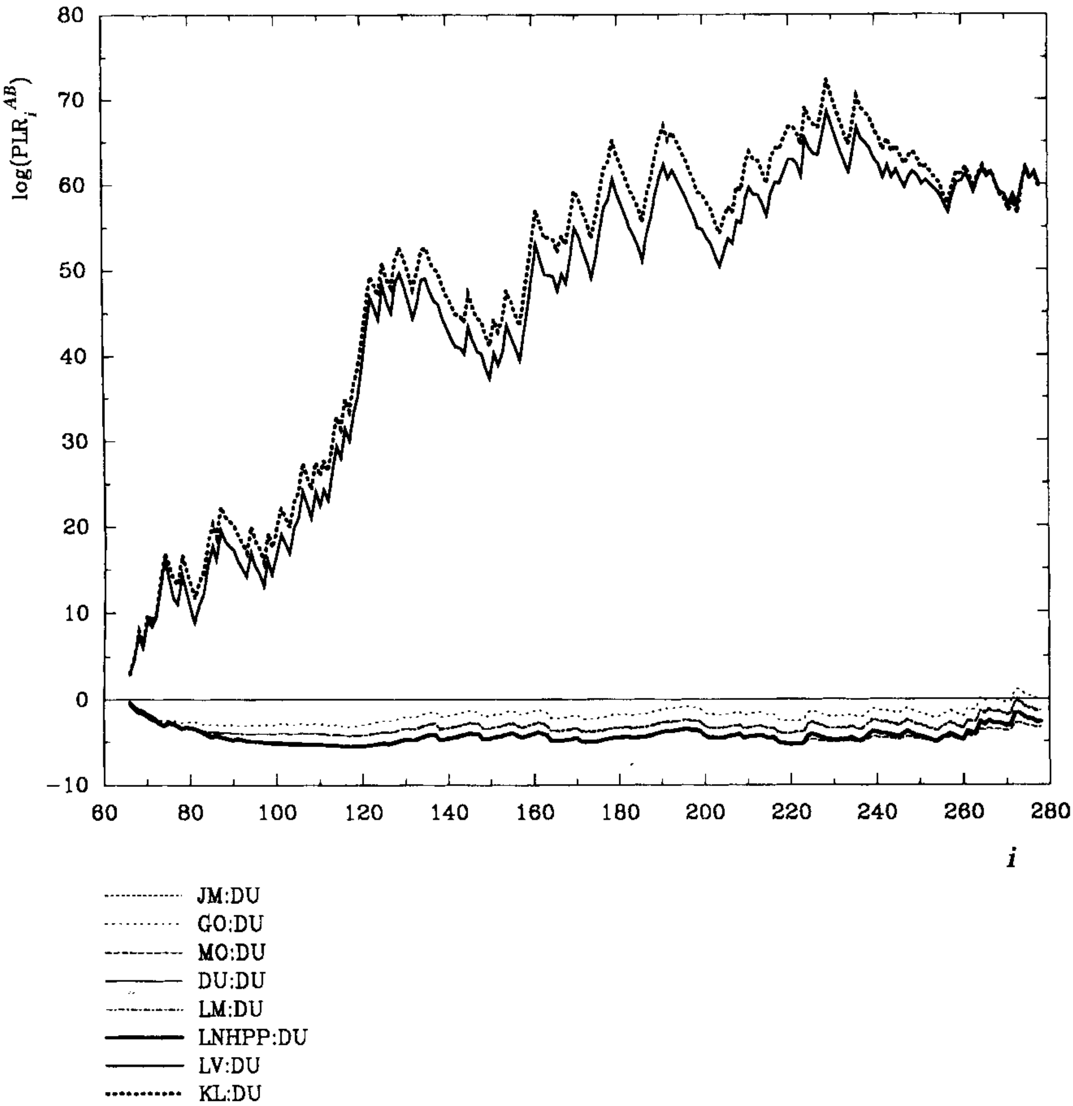


Figure 4.7 Log(PLR) plots for one-step-ahead predictions of T_i, $i = 66, \ldots, 278$, from eight models with the DU model chosen as the reference model against which to compare, for data set SS3. These plots indicate that, within the two groups of models which were previously identified as giving similar median predictions, there is indeed similar accuracy in the predictions and that the LV and KL groups are giving much more accurate predictions than the remaining six models.

The justification we have given here for the PLR is informal and intuitive. There is, however, a more formal asymptotic theory. If, as $i \to \infty$, the PLR$_i^{AB}$ above tends to infinity, it can be shown that "we shall be . . . justified in regarding B as discredited, in favor of A . . ." [Dawi84]. If the ratio tends to neither infinity nor zero, then we cannot make a choice between A and B and *they will deliver indistinguishable predictions.* These results are completely general and concern circumstances (admittedly asymptotic) where we can be sure that A is a "completely better" predictor than B. In other words, they relate to any predictions, however expressed.

4.3.3 The *u*-plot

The PLR is a completely general technique for making *comparisons* of the accuracy of different competing predictions. It does not, however, allow us to say whether any of the predictions are *objectively* accurate. Our first general technique for detecting systematic objective differences between predicted and observed failure behavior is called the *u-plot,* and it is based on a generalization of the simple median check described above.

The purpose of the u-plot is to determine whether the predictions, $\hat{F}_j(t)$, are on average close to the true distributions, $F_j(t)$. It can be shown that, if the random variable T_j truly had the distribution $\hat{F}_j(t)$—in other words, if the prediction and the truth were *identical*—then the random variable $U_j = \hat{F}_j(T_j)$ will be uniformly distributed on (0,1). This is called the *probability integral transform* in statistics [DeGr86]. If we were to observe the realization t_j of T_j, and calculate $u_j = \hat{F}_j(t_j)$, the number u_j will be a realization of a uniform random variable. When we do this for a sequence of predictions, we get a sequence $\{u_j\}$, which should look like a random sample from a uniform distribution. Any departure from such uniformity will indicate some kind of deviation between the sequence of predictions, $\{\hat{F}_j(t)\}$, and the truth $\{F_j(t)\}$. Table 4.3 shows $\{u_j\}$ sequences for the simple predictors A and B mentioned earlier.

One way of looking for departure from uniformity is by plotting the *sample distribution function* of the $\{u_j\}$ sequence. This is a step function constructed as follows: for a sequence of predictions $\hat{F}_j(t)$, $j = s, \ldots, i$ on the interval (0,1), place the points $u_s, u_{s+1}, \ldots, u_i$ (each of these is a number between 0 and 1); then from left to right plot an increasing step function, with each step of height $^1/_{(i-s+2)}$ at each u on the abscissa, as shown in Fig. 4.8. The range of the resulting monotonically increasing function is (0,1), and we call it the u-plot. Figure 4.9 shows the u-plots based on the $\{u_j\}$ sequences shown in Table 4.3 for the two predictors A and B.

TABLE 4.3 u_j^A **and** u_j^B ($u_j = \hat{F}_j(t_j) = 1 - e^{-\hat{\lambda}_j t_j}$) **for Prediction Sequences** A **and** B **Shown in Table 4.1**

j	t_j	u_j^A	$u_{(j)}^A$	u_j^B	$u_{(j)}^B$
12	105	0.650	0.249	0.255	0.197
13	137	0.652	0.289	0.957	0.255
14	125	0.451	0.358	0.588	0.275
15	161	0.518	0.394	0.275	0.275
16	162	0.395	0.395	0.946	0.326
17	153	0.358	0.451	0.275	0.588
18	179	0.394	0.518	0.326	0.755
19	201	0.289	0.650	0.755	0.946
20	220	0.249	0.652	0.197	0.957

$u_{(j)}^A$ and $u_{(j)}^B$ are these same u sequences reordered in ascending order of magnitude.

If the $\{u_j\}$ sequence were truly uniform, this plot should be close to the line of unit slope. Any serious departure of the plot from this line is indicative of nonuniformity, and thus of a certain type of inaccuracy in the predictions. A common way of testing whether the departure is significant is via the Kolmogorov-Smirnov (KS) distance, which is the maximum vertical deviation between the plot and the line of unit slope (see, for example, Fig. 4.9) [DeGr86]; there are readily available tables for this. However, a formal test is often unnecessary: for many of the examples in this chapter it is clear merely from an informal perusal of the plots that the predictions are poor.

Figure 4.10 shows a u-plot analysis of predictions from the previous eight models on the SS3 data. Remember that the informal median analysis showed that none of the eight could be trusted. Recall that the group of six models was very optimistic (i.e., the models were underestimating the chance of the next failure occurring before t), while the other two were pessimistic, although PLR analysis showed that these latter two were in fact less inaccurate than the six. From Fig. 4.10 we can now see the reason for these results: all the models have extremely bad u-plots, with *KS* distances so large that they are well beyond the values that are tabulated. However, while very bad, the LV and KL pair have *KS* distances that are smaller (and so, less bad) than those of the other six—which confirms the PLR analysis.

What is so striking about Fig. 4.10 is that there is such a marked difference in *shape* in the two groups of plots. In fact, informal inspections of u-plots can tell us quite a lot about the *nature* of the prediction errors. The number u_j is the estimate we would have made, before the event, of the probability that the next failure will occur before t_j, the

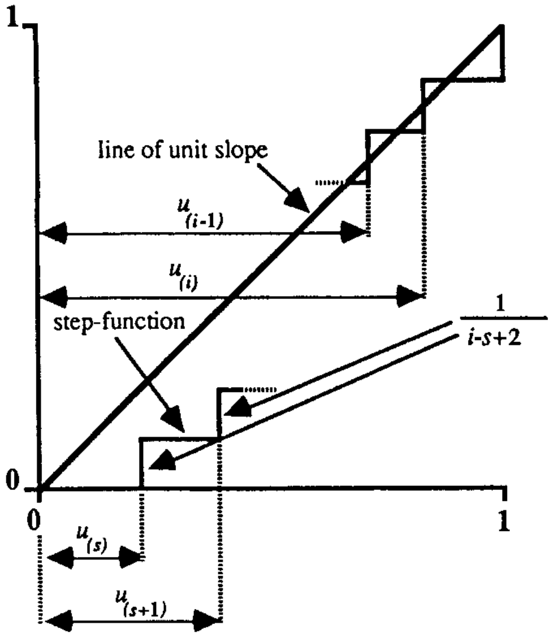


Figure 4.8 How to draw the u-plot, for predictions of $T_s, \dots, T_i$. Here, $\{u_{(s)}, u_{(s+1)}, \dots, u_{(i)}\}$ are the original set of u's $\{u_s, u_{s+1}, \dots, u_i\}$ reordered in ascending order of magnitude.

time when it *actually does* eventually occur. In the case of consistently too optimistic predictions, this number would therefore tend to be smaller than it would be if the predictions were accurate. That means the u_js will tend to bunch too far to the left in the (0,1) interval, and the resulting u-plot will tend to be *above* the line of unit slope. A similar argument shows that a u-plot which is entirely below the line of unit slope indicates that the predictions are too pessimistic. In Fig. 4.10 the plots for LV and KL are almost everywhere below the line of unit slope, indicating that these predictions are objectively too pessimistic; similarly the other six are generally too optimistic. It is sometimes even possible to explain u-plot shapes in terms of inaccuracies more general than simple optimism and pessimism. In Fig. 4.10, for example, there is evidence that the optimism/pessimism argument is a slight oversimplification. Thus, the six models which are generally optimistic seem to be *pessimistic* for predictions associated with the right-hand tail of the

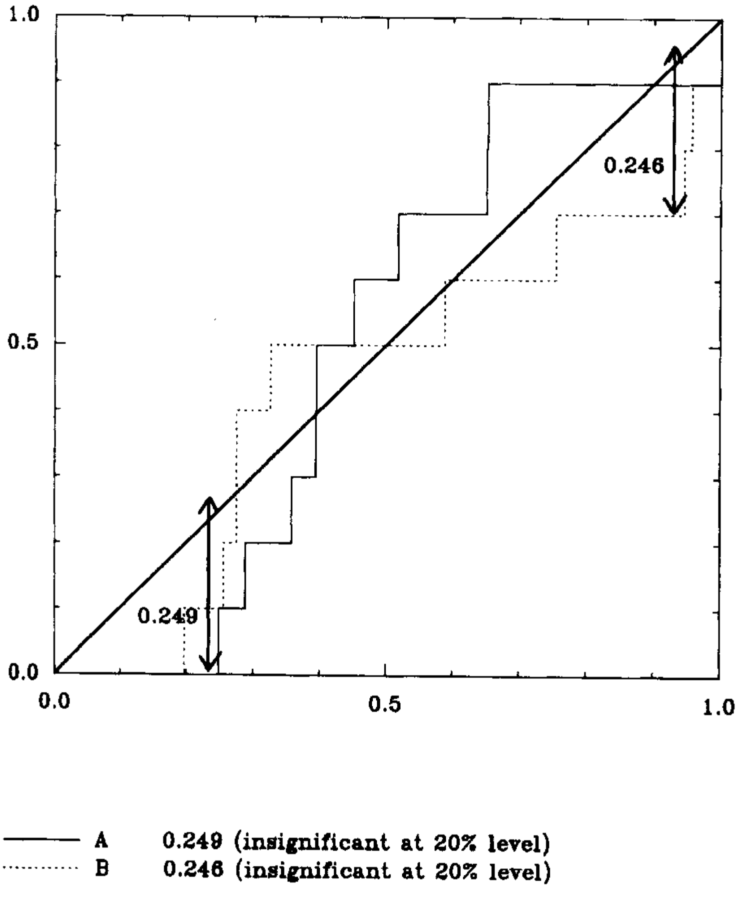


Figure 4.9 u-plots as calculated in Table 4.3 for prediction sequences A and B in Table 4.1. These plots are step functions with step size 1/10. The *KS* distances, indicated on the plots by the arrows, are 0.249 for prediction sequence A, and 0.246 for prediction sequence B, and these values are statistically insignificant at the 20 percent level, indicating that neither prediction sequence is significantly biased.

distribution of time to next failure (i.e., predictions associated with *high* reliability) as evidenced by the plots' crossing of the line of unit slope on the right. The LV and KL predictions, on the other hand, can be seen to be pessimistic in general, but slightly optimistic for predictions of high and low reliability.

Figure 4.11 shows a u-plot analysis of the JM, LM, and LV models on the SYS1 data. The plot for JM is everywhere above the line of unit slope, and its *KS* distance is highly statistically significant. This confirms that the predictions from this model are too optimistic, as we suspected from the earlier simple median analysis. Similarly, LV is too pessimistic, but less dramatically so. The plot of LM is not statistically significant: it thus passes this test, but may, of course, be deficient in some other way.

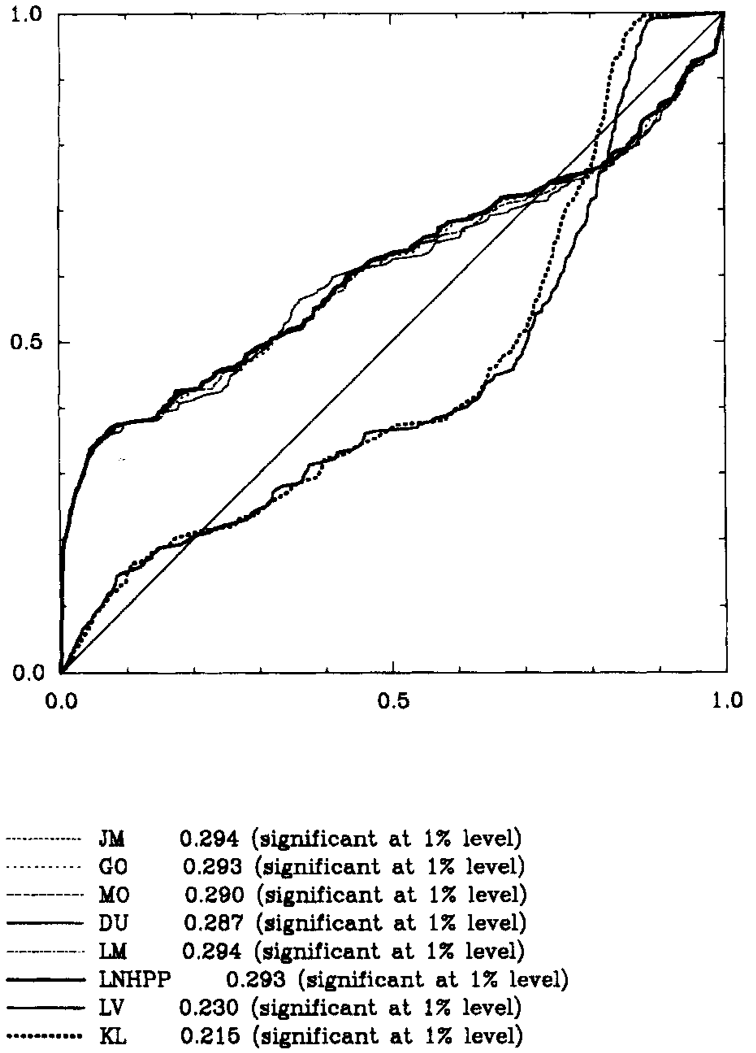


Figure 4.10 u-plots and *KS* distances and significance levels for predictions of T_i, $i = 66, \ldots, 278$, from eight models for data set SS3. The departure of these plots from the line of unit slope indicates that predictions from all eight models are significantly inaccurate for this data set, with LV and KL giving generally pessimistic predictions and the remaining six models giving generally optimistic predictions.

These results for SYS1 and SS3 confirm and explain the earlier results dealing with the medians alone. But it must be emphasized that the u-plot approach is much more general than the analysis we conducted earlier; it relates to the whole shape of the predictive distribution rather than merely to one point (the median) on this distribution. The u-plot can be thought of as detecting a *systematic* difference between the predictions and the truth. This is very similar to the notion of *bias* in statistics: there we use the data to calculate an *estimator* of a population parameter, and this estimator is called *unbiased* if its average value is equal to the (unknown) parameter. Of course, our case is more complex since at each stage we wish to estimate a *function,* rather than merely a number; furthermore, we can only detect prediction error over a *sequence* of *different* predictions because of the inherent nonstationarity of the problem.

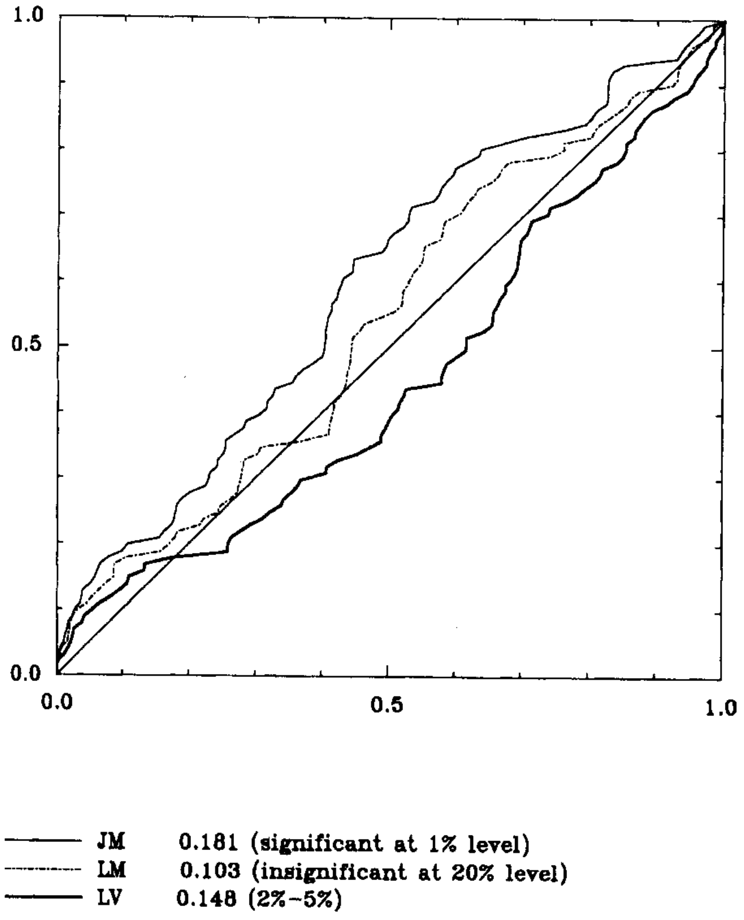


Figure 4.11 u-plots and KS distances and significance levels, for predictions of T_i, $i = 36, \ldots, 136$, from the JM, LM, and LV models for data set SYS1. These plots indicate that the JM model is giving significantly optimistic predictions, the LV model is giving significantly pessimistic predictions, and the LM predictions have, on average, no significant bias. (Note that 2%–5% means that the u-plot for the LM model is significant at the 5 percent level and insignificant at the 2 percent level.)

An interesting special case arises when the prediction errors are completely stationary, i.e., the nature of the error is the same at all stages. There will then be a constant (functional) relationship between $\hat{F}_j(t)$ and $F_j(t)$, and the u-plot is an estimate of this functional relationship. It turns out, in fact, that there is often *approximate* stationarity of errors of this kind. We shall show later in Sec. 4.4 that in such cases it is possible to recalibrate the model—essentially allowing it to "learn" from past mistakes—and obtain more accurate predictions.

4.3.4 The *y*-plot

The u-plot treats one type of departure of the predictors from reality—namely, a kind of reasonably consistent bias. There are other departures from reality which cannot be detected by the u-plot. For example, in one of our investigations we found a data set for which a particular prediction system had the property of optimism in the early predictions and pessimism in the later predictions. These deviations were averaged out in the u-plot, in which the temporal ordering of the u_j's disappears, so that a small *KS* distance was observed. It is necessary, then, to examine the u_j's for *trend.*

There is no obvious standard statistical test for this situation. One way to proceed is as follows, and has the advantage that it results in a plot that is visually similar, and is interpreted similarly, to the u-plot. Remember that the u_j sequence should look like a sequence of independent, identically distributed uniform random variables on (0,1). Since the range, (0,1), remains constant, any trend will be difficult to detect in the u_j sequence, which will look very regular. If, however, we make the transformation $x_j = -\ln(1 - u_j)$, we produce a sequence of numbers that should look like realizations of independent, identically distributed unit *exponential* random variables. That is, the sequence should look like the realization of the successive interevent times of a homogeneous Poisson process; any trend in the u_j's will show itself as a nonconstant rate for this process. There are many tests for trend in a Poisson process. We begin, as in [Cox66], by normalizing the whole transformed sequence onto (0,1). That is, for a sequence of predictions from stage s through stage i, we define

$$y_k = \frac{\sum_{j=s}^{k} x_j}{\sum_{j=s}^{i} x_j} \qquad \text{where } k = s, \ldots, i-1$$

A step function with steps of size $1/(i-s+1)$ at the points $y_s, y_{s+1}, \ldots, y_{i-1}$ is drawn from the left on the interval (0,1), exactly as in the case of the

u-plot. Table 4.4 and Fig. 4.12 show how to construct the y-plot for the two predictors, A and B, considered earlier.

Figure 4.13 shows an example using the same range of predictions as before from the same eight models on the SS3 data. Again, the results divide into the same two groups of six and two models, respectively. The six models have highly significant *KS* distances, so there is evidence that there is trend in the errors being made in the predictions; the results from LV and KL, the other two models, are not statistically significant. This means that the LV and KL predictions, while clearly shown to be in error by our previous analyses, are producing errors that are in some sense stationary. In a case like this, when the error being made remains constant, there arises the possibility of estimating its nature and using this to correct for the error in future predictions (on the assumption that its nature will continue unchanged into the future). This idea will form the basis of our recalibration technique described below.

4.3.5 Discussion: the likely nature of prediction errors, and how we can detect inaccuracy

With the techniques described above we have the beginnings of a framework for making decisions about which model to use within a particular context, and whether the predictions should be trusted to be accurate. It is important to emphasize the differences between, on the one hand, the PLR approach and, on the other, devices such as the u-plot and y-plot. PLR will only tell us about *relative* performance among competing models, but it will do this in the most *general* way possible, with the underlying theory [Dawi84] providing an assurance that all deficiencies have been taken into account. The u-plot and y-plot, on the other hand, give us some *absolute* information, but only about certain *specific* ways in which predictions can differ from the truth.

TABLE 4.4 x_j^A, y_j^A, x_j^B and y_j^B, $x_j = -\ln(1 - u_j)$ and $y_k = \Sigma_{j=12}^{k} x_j / \Sigma_{j=12}^{20} x_j$ for Prediction Sequences A and B shown in Table 4.1.

j	t_j	x_j^A	y_j^A	x_j^B	y_j^B
12	105	1.05	0.191	0.294	0.297
13	137	1.06	0.382	3.15	0.347
14	125	0.600	0.491	0.887	0.437
15	161	0.730	0.624	0.322	0.469
16	162	0.503	0.714	2.92	0.764
17	153	0.443	0.795	0.322	0.796
18	179	0.501	0.886	0.395	0.836
19	201	0.341	0.948	1.41	0.978
20	220	0.286	1.00	0.219	1.00

What this means is that, if we want to ask which of a set of alternative models should be preferred in the analysis of a particular set of data, we should use the PLR. When this gives dramatic evidence of an increasing trend for a pairwise comparison, then we should strongly believe that one model is giving more accurate results than the other. For example, Fig. 4.7 indicates that LV is clearly superior to DU for the SS3 data. However, even when a particular model is clearly superior to others for a particular set of data, it is not necessarily the case that it is giving accurate results: in the case of the SS3 data, according to the u-plots in Fig. 4.10, *all* models were giving results which were inaccurate.

It is important, therefore, after picking out the one (or more) model that performs best on the PLR, to investigate further via u-plot and y-plot analysis. A good u-plot (accompanied by a good y-plot) will tell us that a particular type of consistent bias is absent in the predictions (the good y-plot being needed to ensure that the errors in prediction are at least approximately stationary, so that the u-plot result can be trusted).

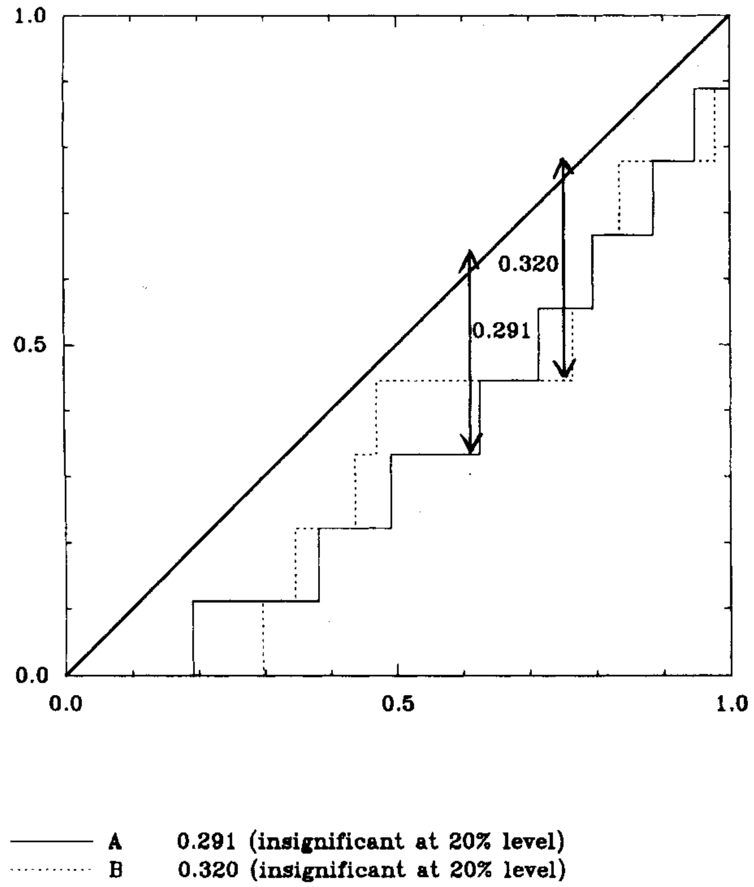


Figure 4.12 y-plots as calculated in Table 4.4 for prediction sequences A and B in Table 4.1. These plots are step functions with step size 1/9. The KS distances, again marked by the arrows, indicate that both prediction sequences A and B are capturing the trend in the failure data.

Of course, predictions can be in error in ways other than the bias that the u-plot detects. Consider the analogy of estimating a population parameter from a random sample in statistics. Even if we have an estimator that is unbiased we may still prefer on other grounds to use a biased one. For example, the unbiased estimator may have a large variance, so that although its expected value is equal to the unknown parameter, any particular calculated value of the estimator may be very far from this. This is the difference between what happens *on average* and what happens at *a particular instance.* Similar argu-

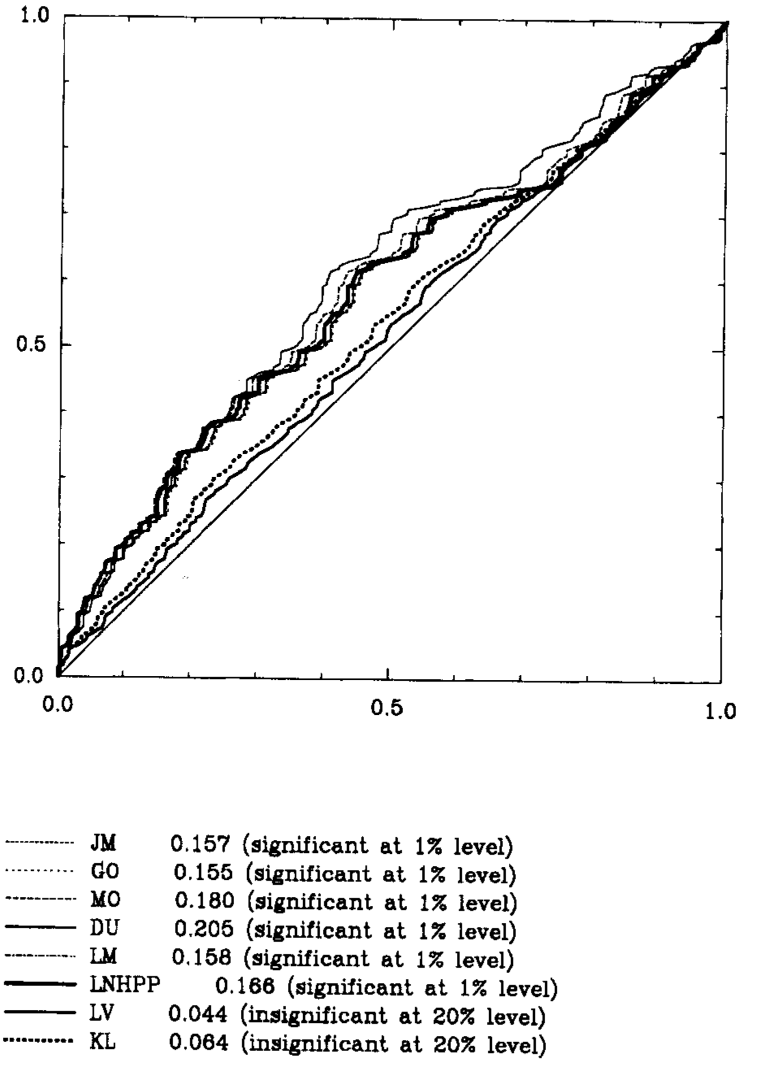


Figure 4.13 y-plots and KS distances and significance levels for predictions of T_i, $i = 66, \ldots, 278$, from eight models for data set SS3. The y-plots for LV and KL show no significant departure from the line of unit slope, indicating that the prediction errors, which we know to be present from the u-plots in Fig. 4.10, are stationary, while for the remaining six models, significant departure in the y-plots is shown, indicating that for these models the prediction errors are not stationary.

ments apply to a good u-plot, which also tells us something about average behavior, but which can mask large inaccuracies on particular predictions. The analogy with variance in our case is a kind of unwarranted noisiness in a sequence of predictions, e.g., predictions that are randomly alternatively too optimistic and too pessimistic, but whose average is close to the truth. Such predictions might exhibit a good u-plot, but any individual prediction could be very inaccurate and hence useless.

It has not been possible to find a way of testing for this kind of inappropriate noisiness in predictions. The problem is that we are considering a much more complicated problem than the simple statistical estimation of a constant parameter from a random sample—in our case we *know* that what we are estimating is nonstationary. Indeed, it is precisely the nonstationarity (the reliability growth) that is of interest to us. It may be the case, then, that this nonstationarity is of a complex form. In particular, there may be genuine reversals of fortune within a general picture of average reliability growth: there may be bad fixes among the good ones. In other words, apparently invalid noisiness in a sequence of prediction may simply be reflecting the *true behavior* of the reliability. The difficult trick is to distinguish noisiness that is merely an artifact of the prediction technique from such real noisiness.

Although there is no direct method of detecting unwarranted noisiness in predictions, this may not be a serious problem. In the first place, it seems unlikely that the evolution of the true reliability will be very noisy in practice. Second, we can get some indirect evidence of inappropriate noisiness from the PLR analysis, since this is sensitive to *all* departures from predictive accuracy. For example, if a model appeared to differ from others in analysis of a particular data set only in its noisiness, *and* its PLR was inferior to others, it would be reasonable to infer that its noisiness was the cause of this poor performance and was therefore unwarranted.

In the next section, where we show how it is possible in some cases to remove the bias errors that are detected by the u-plot, we shall see that the nonstationarity in the prediction errors indicated by a poor y-plot does not in fact appear to be a problem in many cases.

Our own experience, then, is that the PLR and the u-plot alone can be quite powerful tools in deciding whether particular competing predictions should be trusted to be accurate. Certainly their use cannot guarantee that there are no subtle departures between predicted and actual failure behavior, but we believe that the most important and most likely problems are *bias* and *noise,* and that these are usually handled adequately. In Sec. 4.5 we shall work through some examples completely to show how all the different facets of our analytical approach fit together; before that, we complete our description of these new techniques by introducing the idea of recalibration.

4.4 Recalibration

4.4.1 The *u*-plot as a means of detecting bias

We now need to describe carefully what we mean by the notion of *bias* that has so far been discussed quite informally. One way of expressing the notion of *prediction error* more formally is to say that at stage i there is some function G_i which relates the predicted to the true distribution of the time-to-next-failure random variable, i.e., $F_i(t) = G_i[\hat{F}_i(t)]$. Such a function, if we knew it, would tell us everything there is to know about the error in the predictions being made at a particular stage. In particular, if we knew G_i we could recover the true distribution, $F_i(t)$, from the inaccurate prediction, $\hat{F}_i(t)$. In practice, of course, we do not know this function.

However, if we say that a model is *merely* biased in its predictions, then we are asserting not only that there is a difference between what is predicted and the underlying truth, *but that this relationship is constant* so that the sequence G_i is (approximately) stationary, i.e., $G_i \cong G$, say, for all i. In such a case, when there is only a single G function for the whole sequence of predictions, we might try to estimate it and thus provide a means of recalibrating future inaccurate predictions to produce better ones.

The point here is that there is always an unknown function that will transform the predicted distribution into the true distribution, but it is only *sometimes* the case that this function is approximately the same for all i. When this occurs, we have the opportunity of *estimating* this error function from the earlier predictions we have made by comparing these with the observed outcomes. In fact, it can be shown that the u-plot based upon these earlier predictions is a suitable estimator of G [Broc90].

You might reasonably ask whether the condition of stationary errors described above ever applies in real life. In fact such complete stationarity does seem rather implausible. However, as we shall show, this appears not to be critical for our recalibration technique to provide predictions with improved accuracy. And of course, it is not necessary to trust such an approach to be effective, since any recalibrated predictions can be evaluated for accuracy just like any other set of predictions.

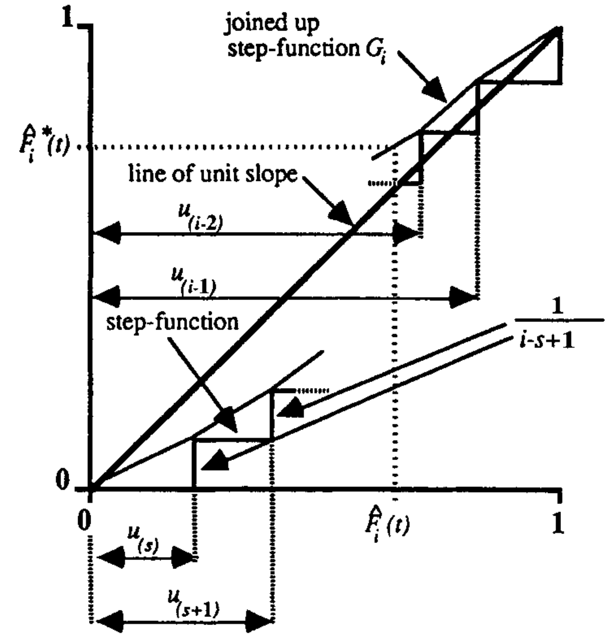


Figure 4.14 The joined-up step function, G_i, of the u-plot of predictions of $T_s, \ldots, T_{i-1}$. Again, $\{u_{(s)}, u_{(s+1)}, \ldots, u_{(i-1)}\}$ are the original set of u's, $\{u_s, u_{s+1}, \ldots, u_{i-1}\}$, reordered in ascending order of magnitude. In fact it is a smoothed version of this step function, G_i^*, which we shall be using to recalibrate predictions, and not simply the joined-up step function, as shown here.

4.4.2 The recalibration technique

The steps of the recalibration procedure are as follows:

1. Obtain the u-plot, say G_i^* based upon the raw[†] predictions, $\hat{F}_s(t), \ldots, \hat{F}_{i-1}(t)$, that have been made before stage i[‡] (see Fig. 4.14). This can be thought of as an estimate of the function G which is assumed to represent the (approximately) constant relationship between prediction and truth.
2. Obtain $\hat{F}_i(t)$, the raw prediction at stage i.
3. Calculate the recalibrated prediction, $\hat{F}_i^*(t) \equiv G_i^*[\hat{F}_i(t)]$ (see Fig. 4.14).
4. Repeat this at each stage i. In this way a sequence of recalibrated predictions will result.

The most important point to note about this procedure is that it is truly predictive, inasmuch as only the past is used to predict the future. This means that it is not necessary to believe a priori that the recalibrated predictions will be better than the raw ones, since the various techniques for comparing and analyzing predictive accuracy can

[†] We use *raw* here to indicate the predictions before recalibration has taken place. Although we usually think of these predictions as coming directly from a reliability model, this is not obligatory; it is possible, for example, that the initial raw prediction sequence is *itself* the result of recalibration.

[‡] For technical reasons, which do not detract from the general explanation given here, it is desirable for G_i^* to be a *smoothed* version of the joined-up step-function u-plot, G_i, shown in Fig. 4.14; a spline-smoothed version, see [Broc90], has been used in the examples that follow.

be used. In particular, the PLR will tell us whether recalibration has produced better results than a simple use of the raw model.

This is a particularly important point in view of the apparently strong assumption of stationarity of errors that underlies the recalibration idea. However, we can obtain some idea of whether there is nonstationarity here by examination of the y-plot. In fact, and quite surprisingly, it turns out that *even in those cases where the y-plot gives evidence of nonstationarity* the recalibration procedure can be shown to give significantly improved accuracy over the raw model. We shall see this in the following examples.

4.4.3 Examples of the power of recalibration

We have already seen that when we apply any of our eight models to the SS3 data, we obtain results that are extremely inaccurate. The u-plots (see Fig. 4.10) are highly statistically significant in all cases. Analysis of the y-plots (see Fig. 4.13) shows that for LV and KL these errors might be stationary, and thus these models are possible candidates for recalibration; the other six have highly significant y-plots and would not at first be thought able to benefit from recalibration. In fact we have applied the recalibration procedure to all eight models and Figs. 4.15 to 4.17 show the results.

Figure 4.15 shows the plots of recalibrated medians, i.e., the medians of the recalibrated versions of the successive predictive distributions.* Comparing this with the plots of the medians from the raw models (Fig. 4.3), we can see that the eight models are now producing median predictions that are in much closer agreement. In fact, the six models that were shown in the earlier u-plots (Fig. 4.10) to be grossly optimistic in their predictions now have much smaller predicted medians; similarly, LV and KL, which were grossly pessimistic, now have larger predicted medians. This might indicate that there is some objective sense in which the recalibrated predictions really are better than the raw ones, and in fact this is shown to be the case in the u-plots of the recalibrated predictions (Fig. 4.16). In comparison with the raw u-plots, the improvement is dramatic in all eight cases, as shown by the KS distances, but this is obvious even from a cursory glance.

What is surprising is that the improvement is so marked in the cases of those six models for which the y-plots tell us that the errors are not stationary. Since stationarity of the underlying sequence of errors is needed to justify the assumption that a single G function can be used

* An S appended to a model name is used to denote the recalibrated version of the model, so JMS is the recalibrated version of the JM model, and so on.

for the recalibration, this is very surprising. We can, however, confirm that there has been objective improvement in all eight sets of predictions by examining Fig. 4.17, which shows plots of the PLR, comparing for each model the recalibrated predictions with the raw predictions. In fact these plots show that there has been *greater* improvement for the six models than for LV and KL—but of course it must be remembered that there was more *room* for improvement in these cases, as shown in the original PLR plots of Fig. 4.7.

In fact, in other analyses we have carried out [Broc87] we have found that in general it does not seem to be necessary to pass the y-plot test in order for recalibration to be effective. In any case, since we have the general procedures of the previous section for analyzing the accuracy of any

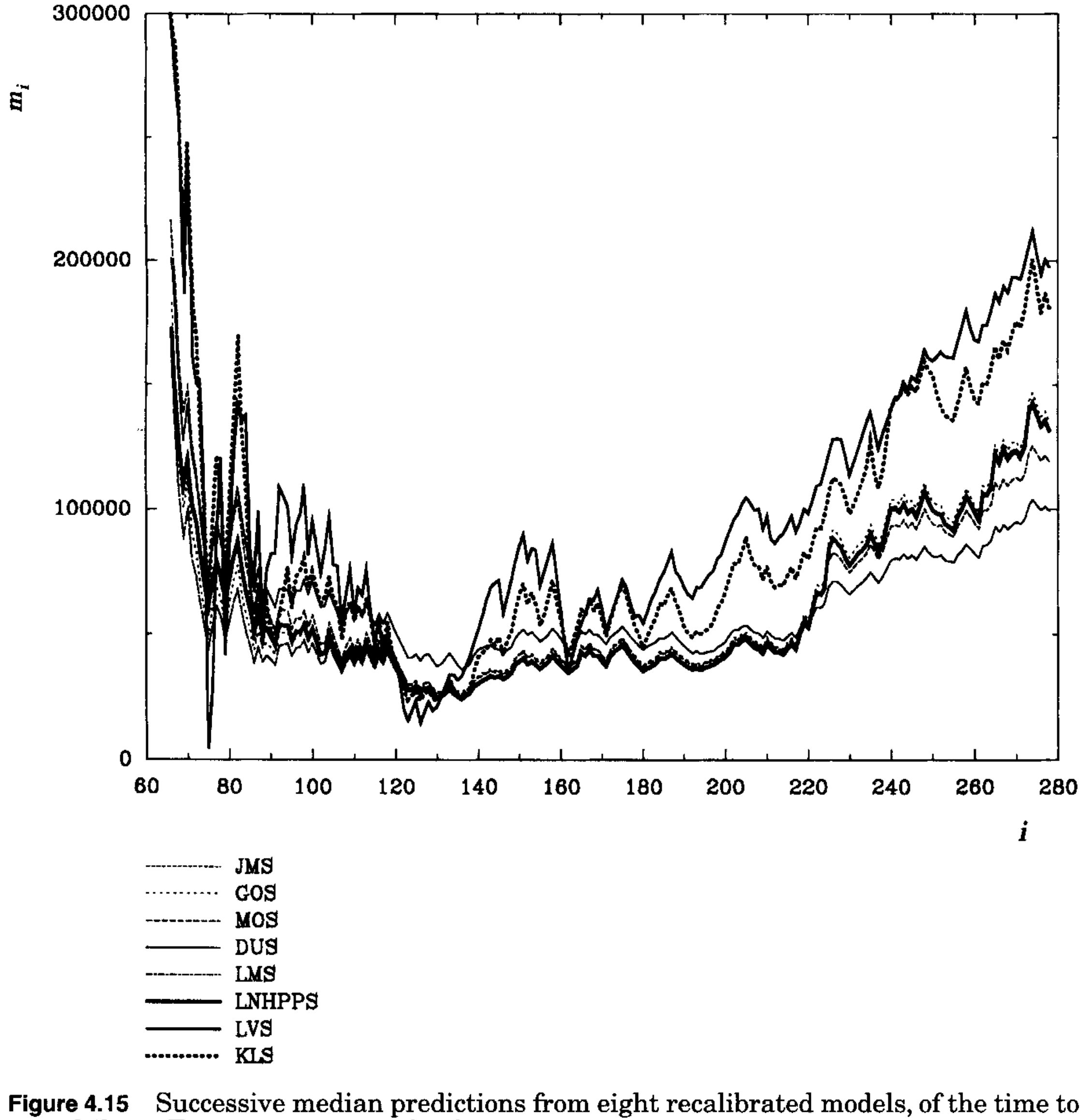
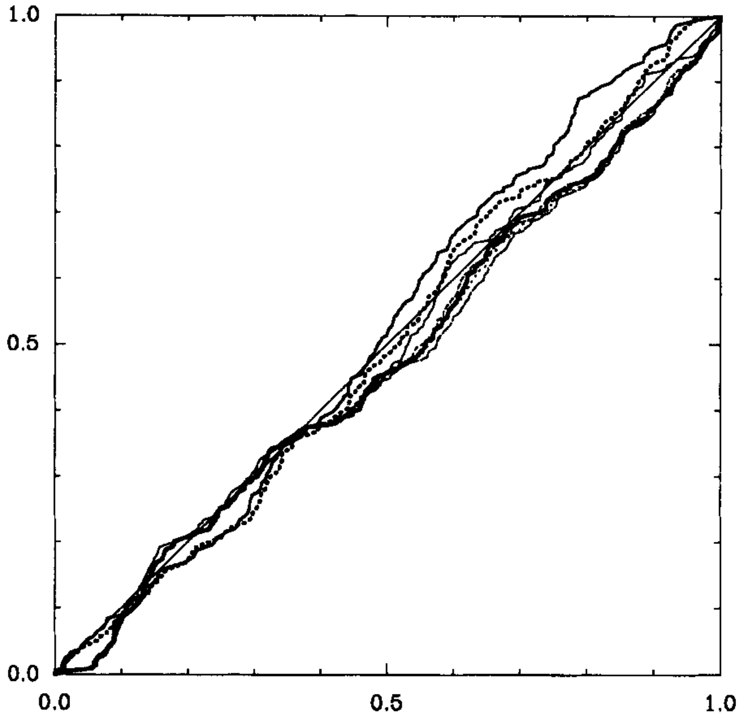


Figure 4.15 Successive median predictions from eight recalibrated models, of the time to next failure, T_i, $i = 66, \ldots, 278$, for data set SS3. Notice how much closer in agreement these recalibrated predictions are when compared with the corresponding raw predictions in Fig. 4.3.



-----------	JMS	0.084 (5%–10%)
···········	GOS	0.073 (10%–20%)
-----------	MOS	0.063 (insignificant at 20% level)
———	DUS	0.056 (insignificant at 20% level)
-·-·-·-·-	LMS	0.085 (5%–10%)
━━━	LNHPPS	0.068 (insignificant at 20% level)
———	LVS	0.087 (5%–10%)
••••••••••	KLS	0.064 (insignificant at 20% level)

Figure 4.16 *u*-plots and *KS* distances and significance levels for predictions of T_i, $i = 66, \ldots, 278$, from eight recalibrated models for data set SS3. These plots are now much closer to the line of unit slope than where the *u*-plots for the corresponding raw predictions (see Fig. 4.10), and the departure is now statistically insignificant, indicating that recalibration has removed bias in the raw predictions.

sequence of predictions, it is not really necessary to know beforehand whether suitable conditions for recalibration exist—we can merely check after the event to see whether there has been an overall improvement.

Recalibration looks like a powerful general technique for improving on the predictive accuracy of any* software reliability growth model. Indeed, it may have applications in other areas of forecasting.

* In this chapter, recalibration is applied only to continuous-time models, but it should be noted that it is also possible to apply recalibration to discrete-time models [Wrig88; Wrig93], where the observations and predictions to be made relate to the number of failures observed in the next period of time.

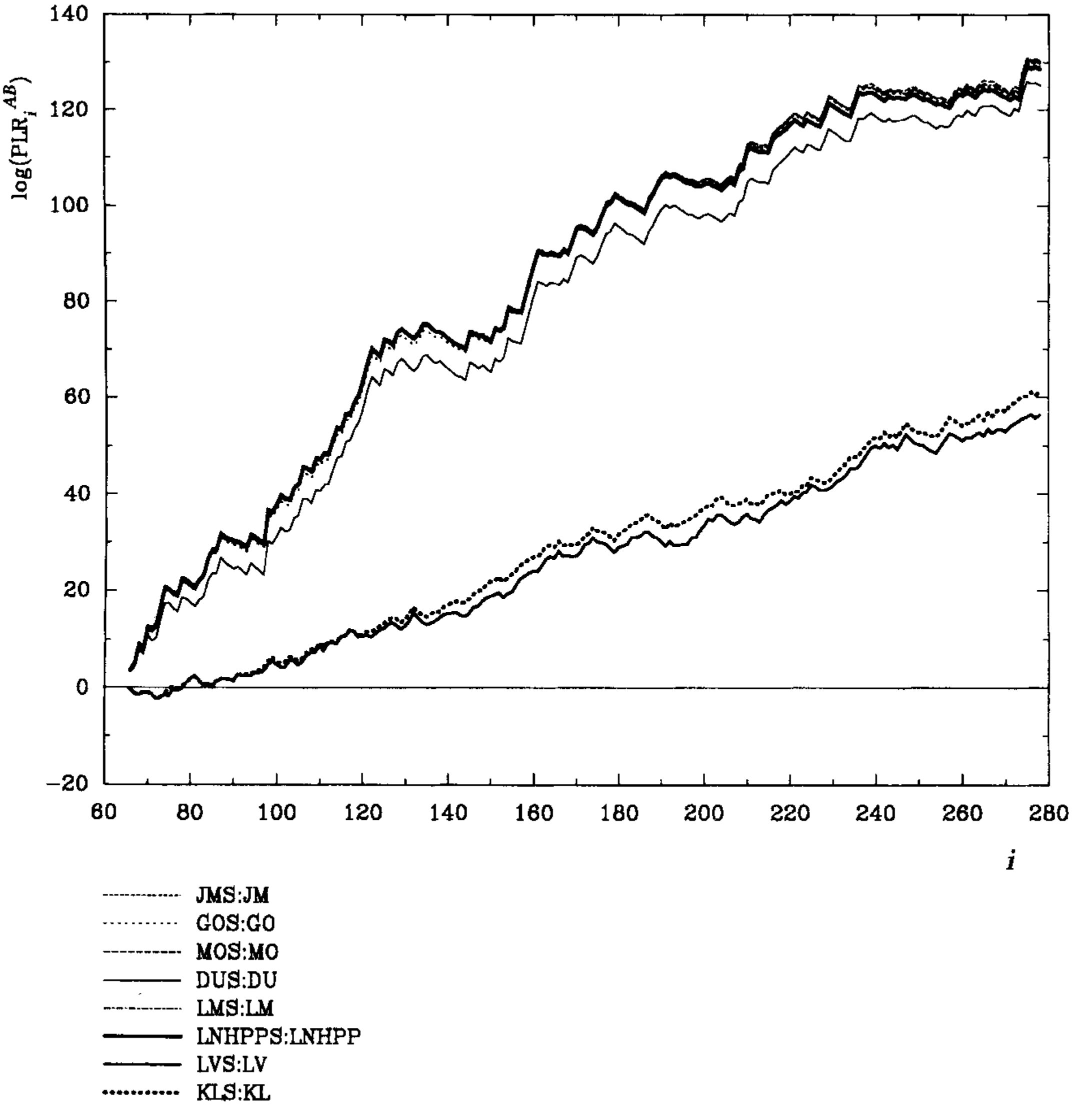


Figure 4.17 Log(PLR) plots for the recalibrated predictions versus the corresponding raw predictions of T_i, $i = 66, \ldots, 278$, for data set SS3. These plots indicate that the recalibrated predictions are much more accurate than the raw predictions; the least improvement is shown for LV and KL, but from Fig. 4.7 it can be seen that these models were initially better than the others and so there was less room for improvement.

4.5 A Worked Example

We have seen in previous sections of this chapter how the different techniques for analyzing predictive accuracy and for recalibrating predictions work on some data sets, SYS1 and SS3. We now present another worked example in which the techniques are used in the way in which we recommend they be used in practice.

Our new data set, CSR1, was collected from a single-user workstation at the Centre for Software Reliability (CSR), and represents some 397 user-perceived events: genuine software failures, together with events arising, for example, from usability problems and inadequate documentation. Figure 4.18 shows the data, and Fig. 4.19 shows a succession of median predictions from the same eight models used previously. There are two striking things to note here: first, there is little evidence of reliability growth until about halfway through the data set; and second, there is again quite marked disagreement between the different models when this growth does start. The *u*-plot of Fig. 4.20 shows that all models are performing very badly, since all the *KS* distances are highly significant. More to the point, there are great differences in the *nature* of the prediction errors being made. Thus JM, GO, LM, and LNHPP are too optimistic (the plot is almost everywhere above the line of unit slope) while LV and KL are pessimistic (the plot is below the line of unit slope). MO and DU, on the other hand, have a pronounced S-shaped *u*-plot, intersecting the line of unit slope at about (0.5,0.5). This indicates that their *medians* are quite accurate, but that estimates of other points on the distribution of time to next failure will be inaccurate: estimates of probabilities of *small* times to failure will be too optimistic, those of *large* times will be too pessimistic.

The PLR analysis in Fig. 4.21 shows that KL is performing best overall, with LV second. The relatively poor performance of the other mod-

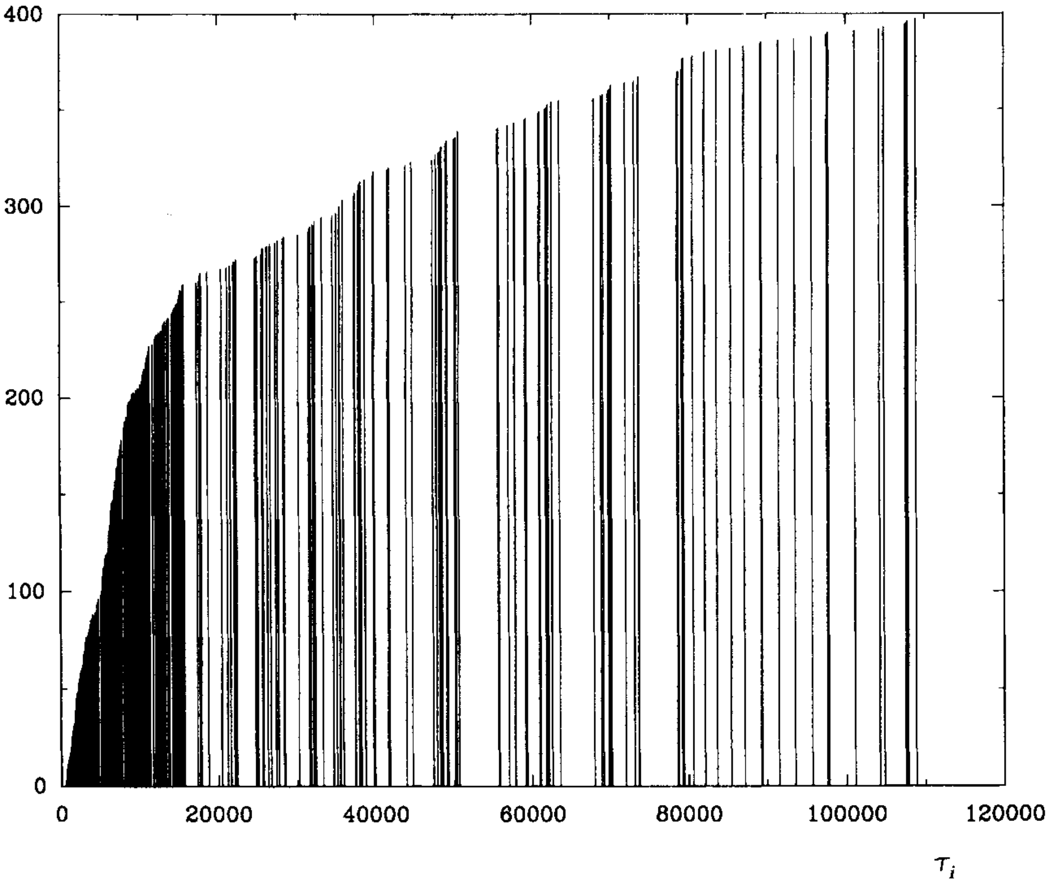


Figure 4.18 Data set CSR1 shown as the cumulative number of failures, *i*, versus the total elapsed hands-on time measured in minutes, $\tau_i = \sum_{j=1}^{j=i} t_j$, $i = 1, \ldots, 397$. This data, collected from a single-user workstation at CSR, represents some 397 user-perceived failure events.

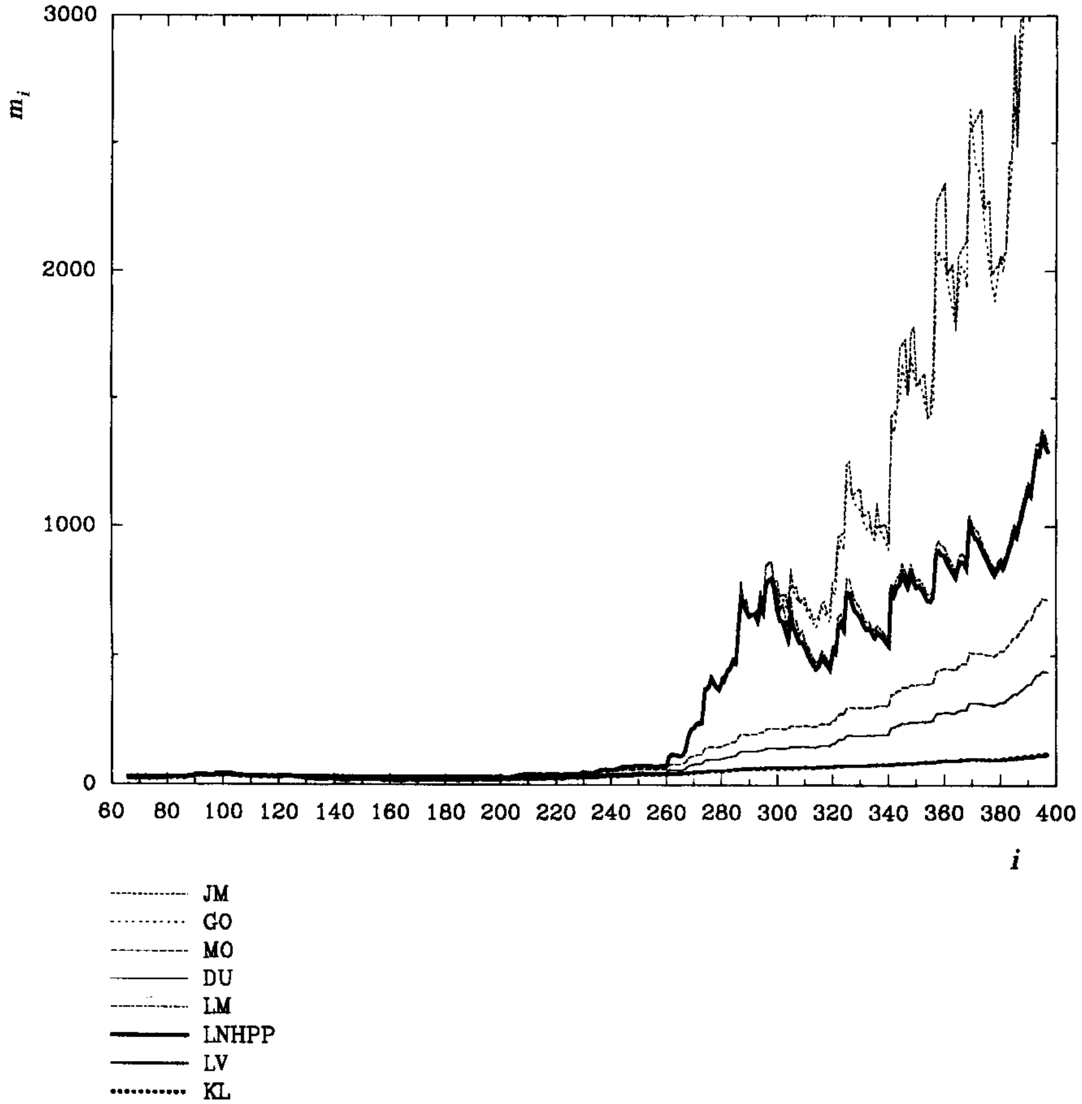


Figure 4.19 Successive median predictions from eight raw models, of T_i, plotted against i for $i = 66, \ldots, 397$ for data set CSR1. Notice how these median predictions are in close agreement in the first half of the data set where there is little evidence of reliability growth, but that they diverge, and increase, in the second half.

els is partly due to bias, as shown by the u-plots, and in some cases by their being too noisy (see the great fluctuations in the medians, for example, in Fig. 4.19). Once again, none of the raw predictions can be trusted according to the u-plot analysis, and these models are thus candidates for recalibration. Figure 4.22 shows the effect of this upon the median predictions: there has been some change in the medians from those obtained from the raw models, and it is in the right direction in view of the original u-plot indications of pessimism or optimism. The u-plot of the recalibrated predictions (Fig. 4.23) confirms that there has indeed been an improvement in comparison with Fig. 4.20. However, only KLS has a plot that does not significantly differ from the line of unit slope (although MOS, DUS, LNHPPS, and LVS are only just sig-

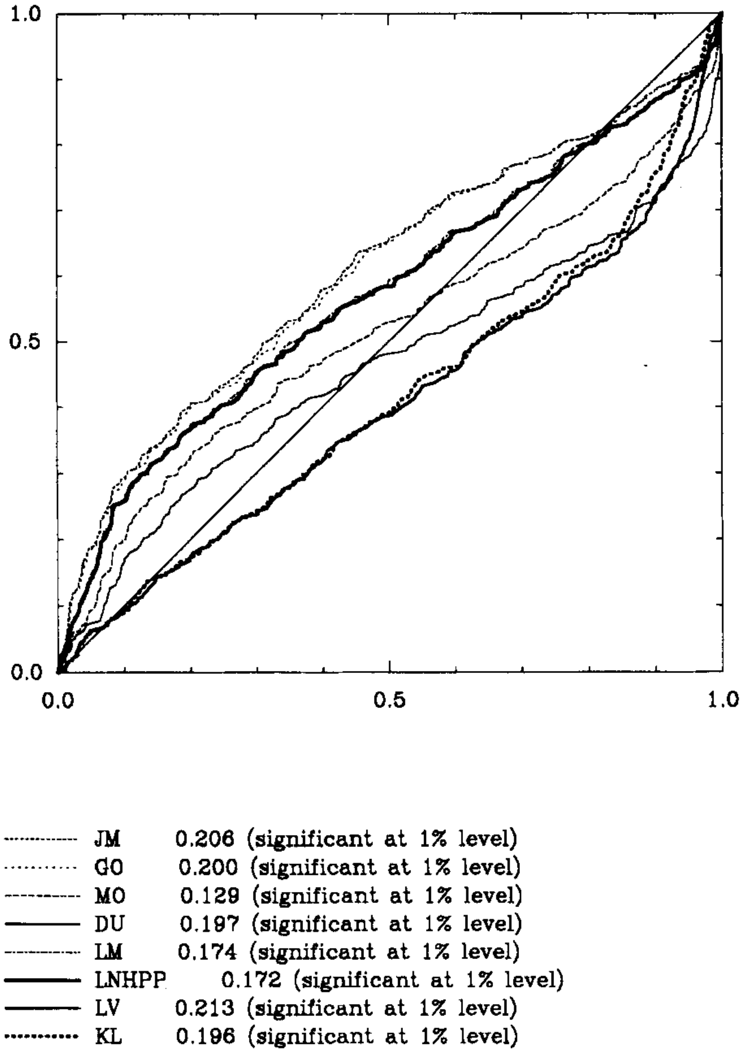


Figure 4.20 u-plots and KS distances and significance levels for predictions of T_i, $i = 66, \ldots, 397$ from eight raw models for data set CSR1. These plots indicate that all these predictions are significantly inaccurate for this data set, with some (e.g., JM and GO) being grossly optimistic and some (e.g., LV and KL) being grossly pessimistic, while others (e.g., MO) have more complicated departures of prediction from the truth than simple optimism or pessimism.

nificant at the 5 percent level). Notice that, in the case of MOS and DUS, while the u-plots have improved a great deal, there is little change in the medians (Fig. 4.19 and 4.22). This is expected, since the raw medians are quite accurate; however, other points on the raw predictive distributions are not accurate, and these will have been improved by the recalibration. Figure 4.24 shows a steady increase in all PLR plots and confirms that, in all cases, the recalibrated predictions are superior to the raw ones. The greatest improvement arising from recalibration is in DU, but this is largely because this model was so bad originally (see Fig. 4.21).

Figure 4.25 shows that after recalibration the best predictions are coming from DUS, with KLS and LVS next best. Thus in this case a user who wished to make further predictions on this data set would be advised to use the recalibrated DU model, bearing in mind, though, that this predictive analysis should be repeated at future stages in case there should be a reversal in fortunes between the various raw and recalibrated predictions from the different models. It is notable that here the recalibration has turned the *worst-performing* model, DU, into the *best*, DUS.

In the analysis of this data set we have deliberately taken no account of the fact that there seems to be little evidence of reliability

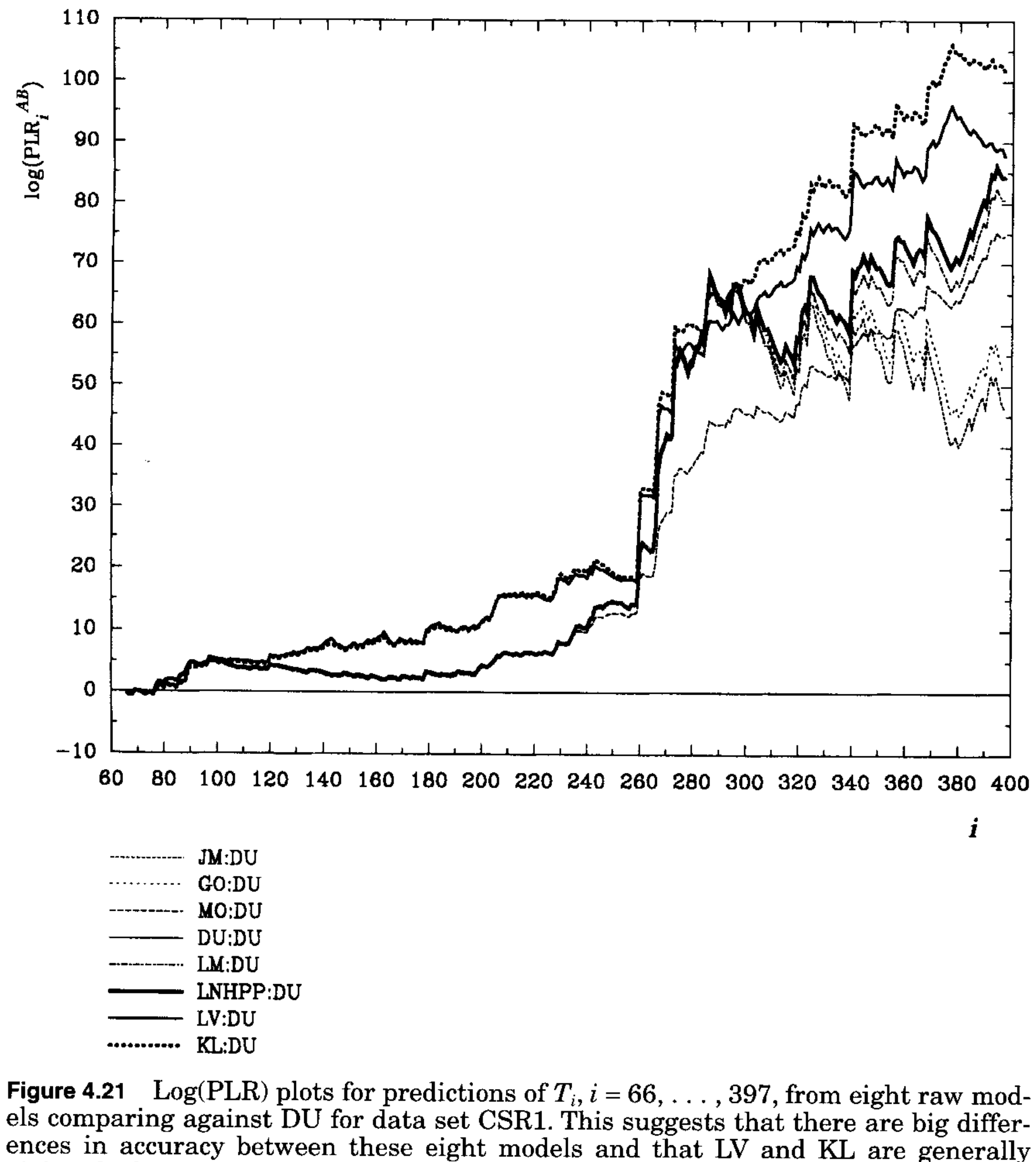


Figure 4.21 Log(PLR) plots for predictions of T_i, $i = 66, \ldots, 397$, from eight raw models comparing against DU for data set CSR1. This suggests that there are big differences in accuracy between these eight models and that LV and KL are generally giving the best predictions, and DU is generally giving the worst.

growth until quite late—rather, we have blindly applied the models and the recalibration procedure as would a naive user. Clearly, it would be a trivial matter to carry out some simple preprocessing of the data to detect the early stationarity (for example, applying simple tests for trend). In the event that there is no growth indicated in the early part on the data, it would be sensible to exclude this data and apply the growth models only to the *later* stages where growth *is* present.

For a similar analysis considering only failures that are *known* to be due to software faults in CSR1 data set, see Prob. 4.7.

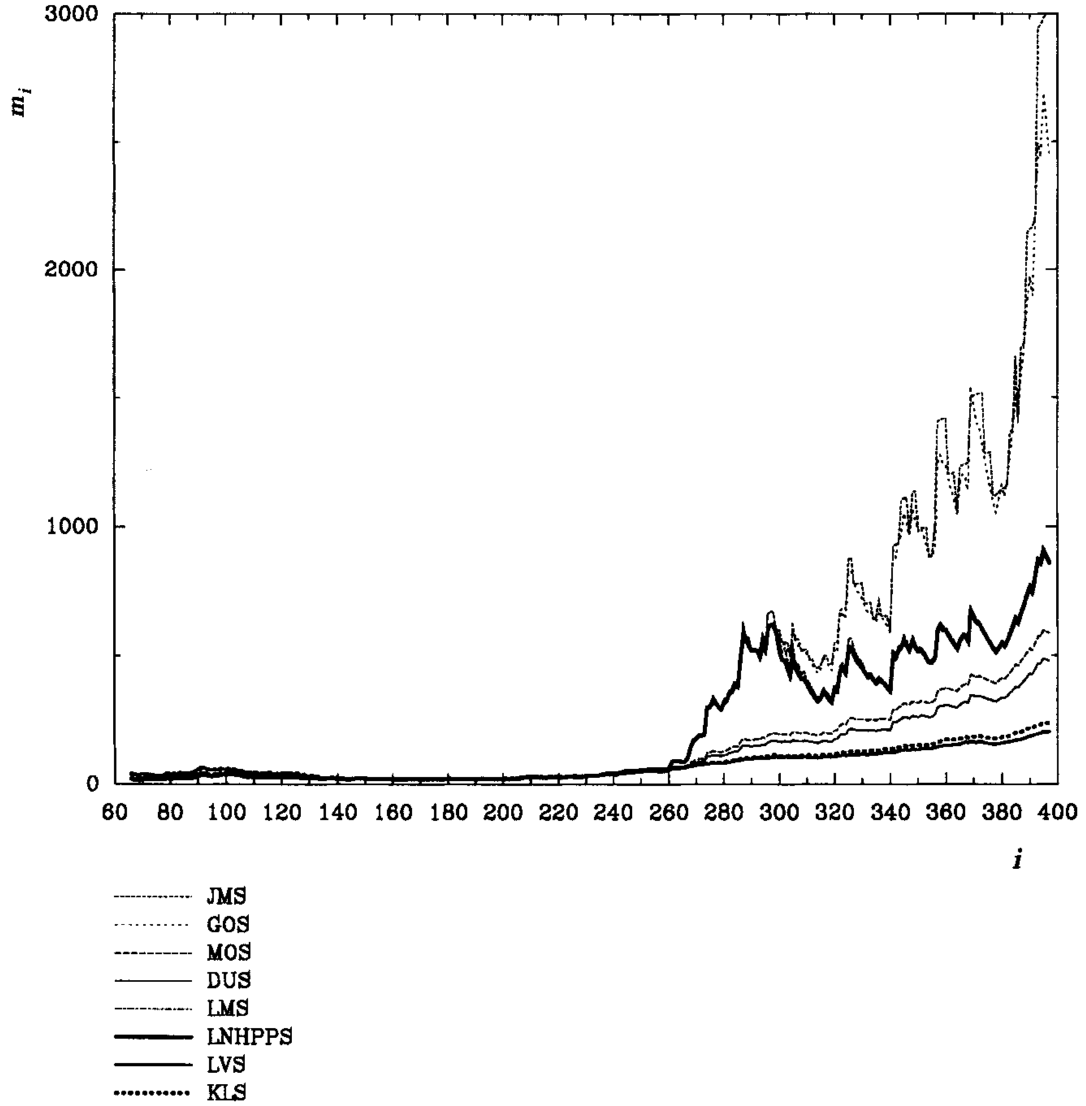
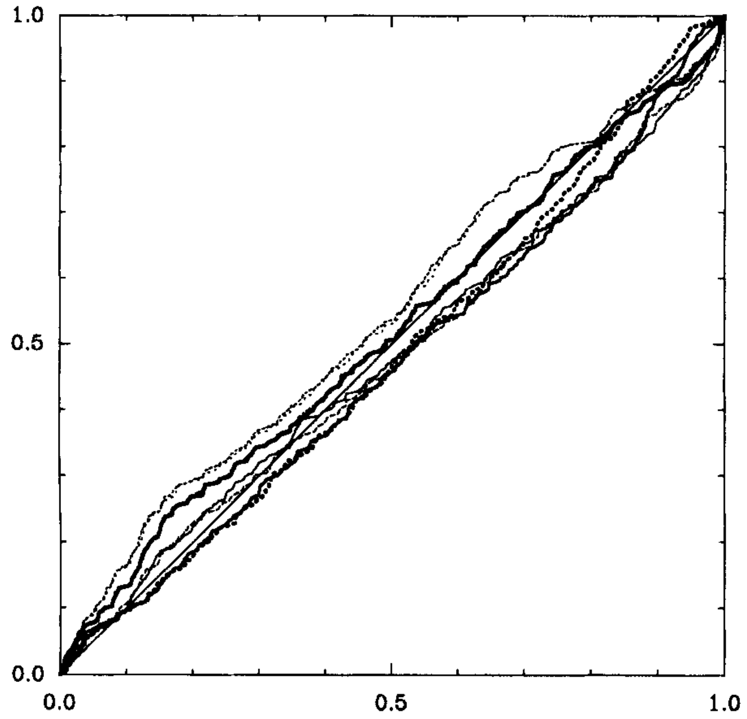


Figure 4.22 Successive median predictions from eight recalibrated models of T_i, plotted against i for $i = 66, \ldots, 397$ for data set CSR1. Comparing with the raw medians in Fig. 4.19, it can be seen that these are in closer agreement than before, but that they still diverge in the second half of the data set.



- - - - -	JMS	0.110 (significant at 1% level)
·········	GOS	0.107 (significant at 1% level)
— - — -	MOS	0.078 (2%–5%)
———	DUS	0.075 (2%–5%)
— — —	LMS	0.083 (1%–2%)
━━━	LNHPPS	0.081 (2%–5%)
———	LVS	0.080 (2%–5%)
●●●●●●	KLS	0.055 (insignificant at 20% level)

Figure 4.23 u-plots and KS distances and significance levels for predictions of T_i, $i = 66, \ldots, 397$ from eight recalibrated models for data set CSR1. Notice how these have improved when compared with the raw u-plots in Fig. 4.20, indicating that the bias in the raw predictions has been reduced by recalibration.

4.6 Discussion

4.6.1 Summary of the *good* news: where we are now

In this chapter we hope we have convinced you of two things.

First, there are serious problems that need to be addressed concerning the accuracy of reliability growth models. There is no universally acceptable model that can be trusted to give accurate results in all circumstances; users should not trust claims to the contrary. Worse, we cannot identify a priori for a particular data source the model or models, if any, that will give accurate results; we simply do not understand which factors influence model accuracy.

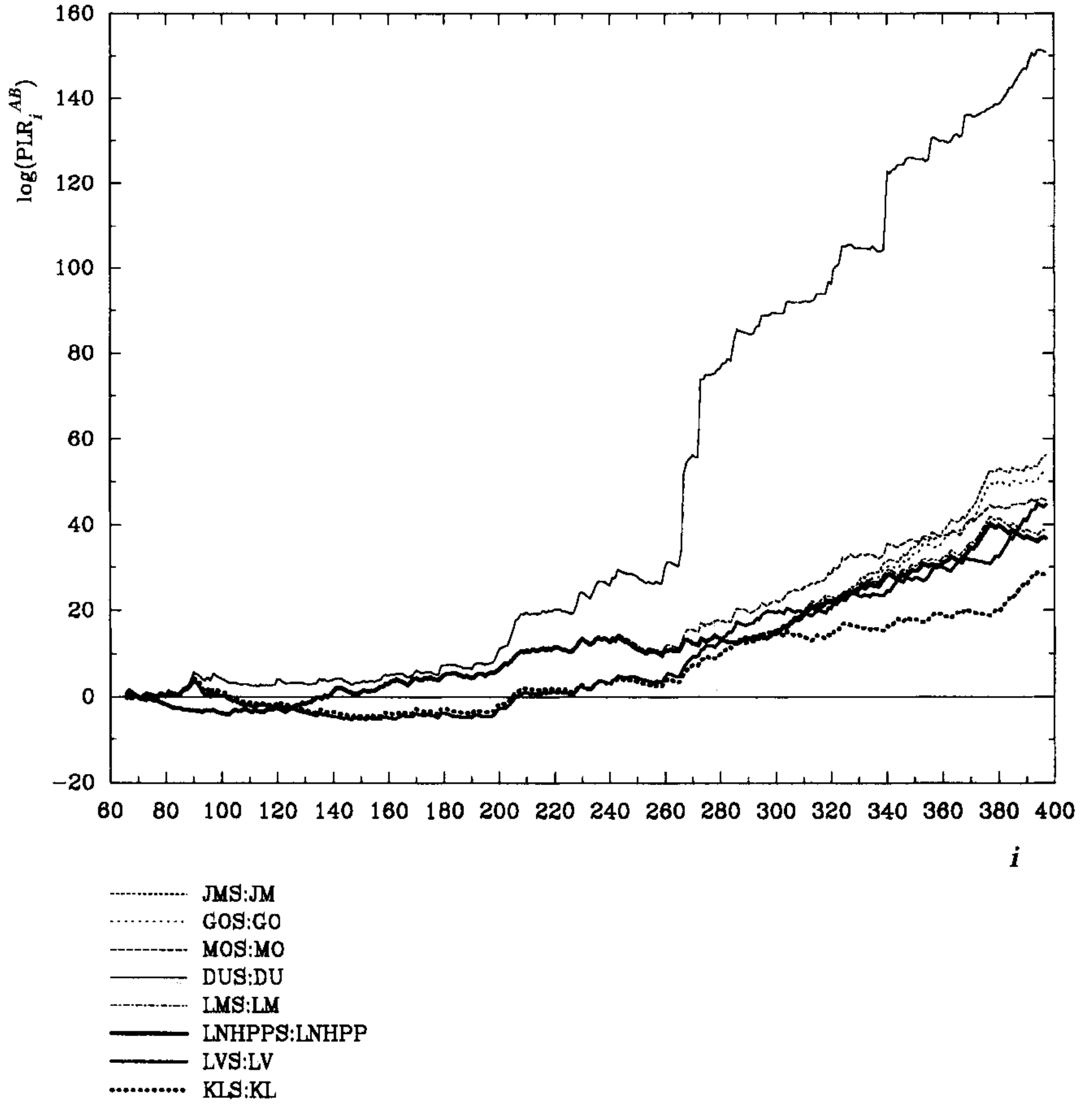


Figure 4.24 Log(PLR) plots for the recalibrated predictions versus the corresponding raw predictions of T_i, $i = 66, \ldots, 397$ for data set CSR1. These plots indicate that the recalibrated predictions are more accurate than the raw; the most dramatic improvement is shown for DU, but from Fig. 4.21 it can be seen that this model was most in need of improvement in the first place.

Second, and more hopefully, there are techniques which can rescue us from this apparent impasse; they allow the accuracy of the actual predictions being obtained on a particular data source to be analyzed. One of these techniques for analyzing the accuracy of predictions also brings with it a bonus: it is possible to use it to assess the errors in past predictions made by any raw model and hence to recalibrate future raw predictions in order to eliminate such errors. With this new approach to software reliability prediction, we believe that users will normally be able to obtain reliability measures and predictions in which they can have confidence, and that this confidence will be justified. In those situations where it is not possible to obtain accurate results from any of the models, even with recalibration, users will get a warning that this is the case.

It must be admitted that the ways of examining the accuracy of predictions that we have described are nontrivial, and users may find them at first quite unfamiliar. This is not surprising, since traditional statistical methods have tended to neglect the problem of prediction in favor of estimation. It is only recently that techniques such as PLR analysis have become available. However, the *use* of the techniques is really quite straightforward, normally involving nothing more than the simple graphical analysis we have seen in the examples. Further, these new measures are implemented in some of the current software reliability tools (SRMP, SMERFS, and CASRE) which are discussed in App. A.

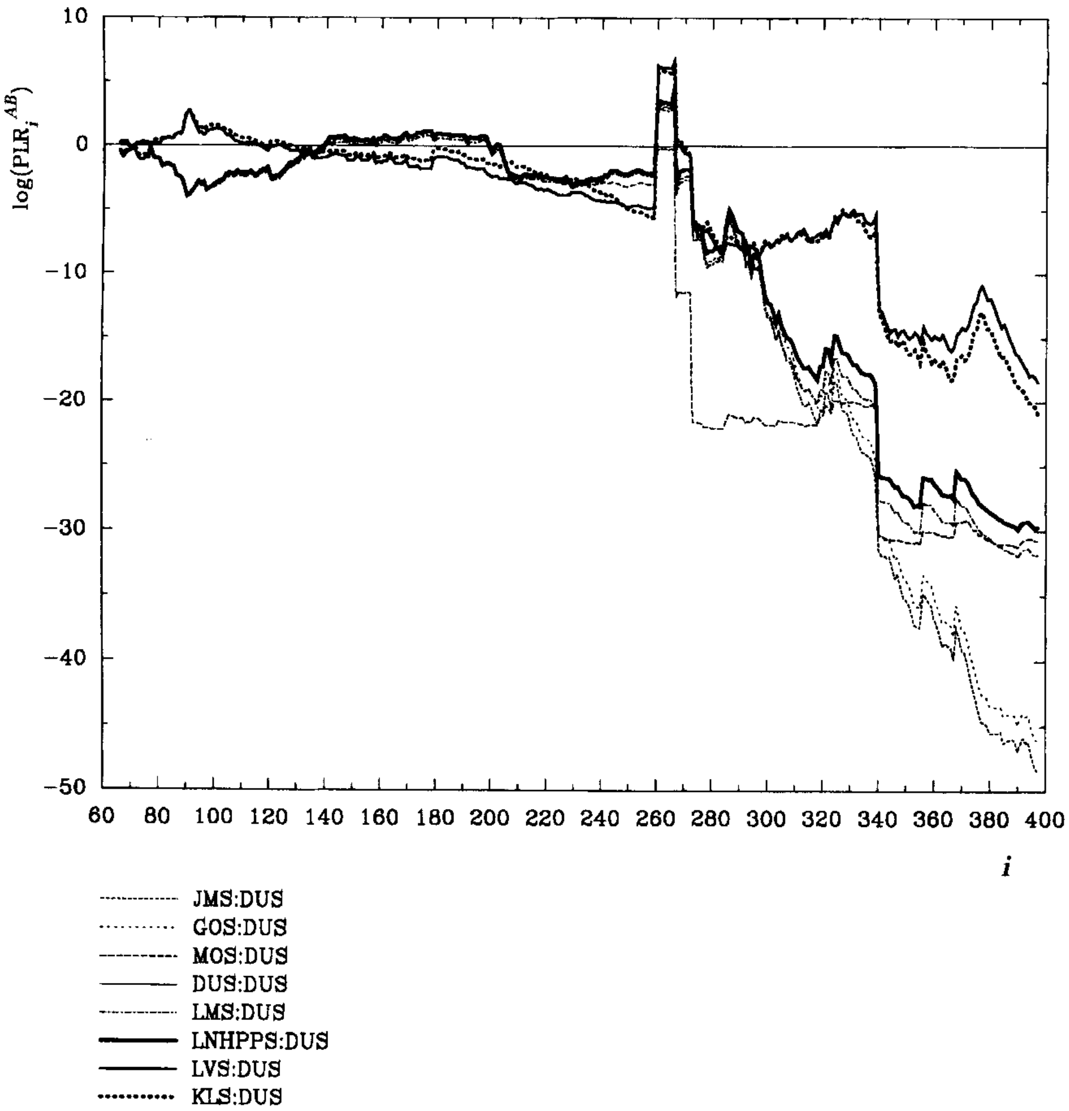


Figure 4.25 Log(PLR) plots for predictions of T_i, $i = 66, \ldots, 397$ from eight recalibrated models comparing against DUS for data set CSR1. This suggests that the best recalibrated predictions are coming from DUS, which we know from Fig. 4.21 was the worst before recalibration, with KLS and LVS next best.

4.6.2 Limitations of present techniques

One major limitation of the techniques for prediction evaluation and recalibration described in this chapter relates to the earlier discussion in Sec. 4.2.2 on long-term predictions. It was noted that prediction errors are likely to be different depending on the nature of the predictions being made and, in particular, that the nature of the error in one-step-ahead predictions is likely to be different than for longer-term predictions, for example, as we have seen, 20 steps ahead. For simplicity, the techniques described here have concentrated upon one-step-ahead predictions only. There are clearly practical restrictions when we try to extend these techniques to predictions further than one step ahead. Theoretically, an extension to n-steps-ahead predictions is fairly obvious; we could make many such predictions, and then evaluate their accuracy by comparison with the corresponding observations when they are finally later observed. The practical limitation of course lies with the value of n. The larger the value of n, the less likely it is that we will ever have enough data to make many such observations and thus to conduct such an analysis of accuracy, or to recalibrate such predictions. This problem becomes even worse when we consider predictions such as estimating how long it will take to achieve a target reliability.

Although these problems due to sparseness of data apply to a greater extent to the topics of evaluation and recalibration discussed in this chapter, they also apply to software reliability modeling in general. A major limitation of the whole software reliability growth approach is that it is really only practicable for those situations in which rather modest reliability levels are involved. If we require ultrahigh reliability, and need to evaluate a particular system in order to have confidence that such a target reliability has in fact been achieved, these techniques will not be sufficient. The problem, of course, lies not in a deficiency in the reliability growth approach itself, but in the fact that the amounts of data needed are prohibitively large. It has been shown in [Litt93] that the length of time needed for a reliability growth model to acquire the evidence that a particular target mean time to failure has been achieved will be *many times* that target. Worse, this multiplier tends to become larger for a higher target reliability—there is a law of diminishing returns operating in software reliability growth. This kind of result seems to occur whatever model we use and whatever the nature of the reliability measure adopted, and effectively precludes these kinds of techniques from providing evidence that a system has achieved ultrahigh reliability.

Another serious restriction to the usefulness of all these techniques lies in their need for the inputs to be selected in the same way during the collection of data as it will be in the period to which the predictions relate. Thus, if we wish to predict the reliability of a program in its operational environment, we need to base our predictions upon failure data collected when the software was operating in such an environment (or good simulated approximation of it). In this sense, the models (and recalibration) work by a sophisticated form of extrapolation, and we can expect the results to be accurate only if past and future are similar. For some applications, constructing a test environment that is a realistic replica of an operational one is not too difficult, but it must be admitted that sometimes this is a difficult task. On the other hand it could be said that some reasonable approximation to the intended operational profile should be considered to be part of the specified requirements for a system, rather than this just being limited to the functional requirements. There is a sense in which it could be said that not knowing the expected operational profile is like not having completely specified the requirements of the real-world problem to which the software is intended to be the solution.

Furthermore, the notion of a *single* operational environment can be too restrictive. Some programs go out into the world and are used in different ways by many users. These users will then often experience different reliabilities for the same program. Of course, we could construct many different test environments to try to reproduce the different types of operational use, but this would be expensive. Ideally, we would like to be able to take the failure data from a (nonrepresentative) test environment and use this together with information about the operational environment to predict operational reliability. Chapter 5 discusses operational profile techniques in detail.

4.6.3 Possible avenues for improvement of methods

It has been our experience from using the techniques for evaluating predictive accuracy and recalibration that with the many current models it is usually possible to obtain trustworthy results from one of them—either before or after recalibration. It has also been our experience that having more sophisticated models does not necessarily lead to predictions (either before or after recalibration) which are more robust or applicable over a wider range of data sets than those with simpler assumptions. Unless there are pressing reasons to the contrary, we believe that research would be better conducted in areas other than model building.

Having said that, some of the present models and the analysis techniques could benefit from further work. For example, it is hard to predict far into the future with some models, and exact results are not available. Similarly, as discussed in the previous section, there has not been much work on the analysis of the accuracy of such predictions. Depending upon their precise nature, such predictions may require us to develop more advanced versions of the analysis techniques described here.

In addition to this there are other possible ways in which these methods for improving raw predictions can themselves be improved. For example, as we observed earlier with the CSR1 data in Sec. 4.5, there are some data sets for which, even though recalibration gives dramatic improvement over the raw predictions, there is still room for further improvement. In these cases it is apparent that the recalibrated predictions are still biased because there is nonstationarity in the raw prediction errors. There are several possible methods which could be investigated in such cases. We could apply the recalibration method again to those recalibrated prediction sequences which are still biased. Alternatively, in the presence of nonstationary raw prediction errors it is reasonable to assume that the most recent prediction errors reflect more accurately the current prediction error than those further into the past; it would thus seem sensible to use only these most recent predictions in recalibrating the current prediction. This, in turn, naturally leads us to the possibility of investigating methods which formally test for changes in the prediction errors so that we can decide which of the past raw predictions to use in recalibrating the current prediction. Investigations so far indicate that applying recalibration using only very recent predictions tends to eliminate bias successfully (i.e., good u-plots result) but sometimes this decrease in bias is outweighed by an increase in noise in the resulting recalibrated predictions, and so there is a trade-off to be made here. An alternative to the search for an optimum window of predictions in such cases (i.e., a window which results in bias reduction that is not outweighed by increased noise) might be to investigate the possibility of direct methods for the elimination of noise—i.e., smoothing techniques.

A related subject is the investigation of other techniques for improving raw reliability predictions, such as the combining techniques which are considered in Chap. 7. Here any group of predictors (raw or recalibrated) may be combined to form a new predictor. Like recalibration, the new predictors generated from these combination techniques are genuinely predictive (being based only on past data) and so the analysis techniques discussed in this chapter can be used to assess the benefit gained from combination. Various combination techniques have been investigated and the most promising seem to be those where the combination depends on past predictive performance of the initial predictors; for example, where the combined predictor is a combination with more weight given to those initial predictors which have performed the best in the (recent) past. Investigation so far indicates that the main benefit of these combination techniques is that they result in a new predictor comparable with the best of the initial predictors. The main advantage of these techniques is that the result is *automatic* selection of a best predictor for a particular data set. This is important, since one major criticism of reliability modeling is that the user of such techniques needs to be reasonably expert. There is much further work to be done in this area: for example, trying more sophisticated combination methods, testing for appropriate past intervals of predictions on which to base combination at each prediction stage, and so on.

The problem outlined in the previous section of predicting the reliability of a program in a different operational environment from the one in which the failure data has been collected is an important one that might benefit from research. There has been some work on this problem [Cheu80, Litt79b], based on a structural decomposition of the software, where the operational profile of a program is characterized by the Markovian exchanges of control between its modules. The idea here is that the reliability estimation for the modules could be performed once and for all in a testing environment, using the reliability growth models, and then the reliability of the overall program could be predicted for any new operational environment merely by estimating the parameters of the Markov process for the new environment. This latter task should be much easier than the reliability estimation. There seems to be no experience of using these ideas, however, and it might be questioned whether some of the modeling assumptions are realistic.

A criticism that is often made of the reliability growth models is that they give their answers far too late—what is needed, it is stated, is a means of estimating and predicting the reliability of a system at a much earlier stage in its development so that corrective action can be made if necessary. We are skeptical about being able to make genuine predictions of final system reliability at an early stage, but it may be possible to identify attributes of the early development process that will indicate potential future problems.

A more promising approach might be to try to identify some attributes of process and product that can be used *with* the later failure data in order to obtain more accurate models. There has been considerable interest in the statistical literature over recent years in stochastic models with such explanatory variables; the problem in this case seems to be that of identifying variables with genuine explanatory power.

4.6.4 Best advice to potential users

The first and most important advice we would give if you are setting out to measure and predict software reliability is: be skeptical. There is no model that can be relied upon to be accurate under all circumstances (although there do seem to be some models that are inaccurate on most data sources). Nor can we identify a priori those circumstances where a particular model *is* appropriate and *will* give accurate results.

In the face of these difficulties, we believe that there is no alternative but to adopt the eclectic approach that we have described here. Many models should be used simultaneously, and their output compared with the actual failure times using the techniques we have described. The result in most cases will be that reliability predictions will be identified that are trustworthy with respect to certain important types of possible error, and this trustworthiness will be *demonstrated.* The latter point is particularly important—with our approach there is no need to appeal to dubious arguments such as model plausibility, or past good performance on other data sets.

Finally, we have a bonus in our recalibration technique, which seems to work in a high proportion of cases, giving results that are better than those of the corresponding raw models. Experience of applying this recalibration technique has shown that it often gives dramatic improvement over the raw predictions, and only in rare circumstances will marginally worse predictions result.* Once again, as a user you do not need to, and should not, *trust* our claims for the efficacy of this technique—rather you should treat it as another source of competing predictions that need to be analyzed for accuracy on your data, using the methods we have described, just as with any other predictions.

Although the techniques we have described depend upon rather novel and subtle statistical methods, we think that their actual use and interpretation from the graphical presentations are comparatively straightforward. This is aided by the use of some of the software reliability tools which are discussed in App. A. Our advice if you are contemplating measuring and predicting software reliability is to go ahead and try our approach. Most times you will get results you can trust. In those rare cases where none of the raw or recalibrated models work, our techniques will give you a warning.

4.7 Summary

The techniques we have described here are important because they largely resolve a basic dilemma of software reliability modeling: a user is now faced with a plethora of models, but no one of them can be rec-

* This sometimes occurs when the raw model predictions are already unbiased and so there is no room for further improvement. In such circumstances the recalibrated and raw predictions are approximately the same, although recalibration may add some noise to the predictions.

ommended for universal use. Indeed it is our belief that the relatively poor take-up of software reliability modeling techniques has been a result of certain models being sold as universal panaceas. Users rightly adopt an attitude of "once bitten, twice shy" when they see these models occasionally giving ludicrous results.

We think that the techniques we have developed provide a means to overcome these difficulties and that it is now possible to measure and predict software reliability for the relatively modest levels that are needed in the vast majority of applications. Most important, the techniques provide a means whereby the user can be *confident* that the results are sufficiently accurate for the particular program under examination. There is thus no need to subscribe to dubious claims about the inherent plausibility of a particular model in order to have some assurance that the reliability figures can be trusted.

One of the analysis techniques described in this chapter also brings with it a bonus: it allows us to assess the nature of the inaccuracy of past predictions and to recalibrate future predictions in order to improve predictive accuracy. The examples we chose for this chapter are ones in which the raw models perform rather badly. We did this deliberately to show the power of the recalibration technique, but it is often the case that some individual raw models will perform reasonably well even before recalibration. From a user's point of view, however, this is immaterial. The recalibration procedure is easy to use and is genuinely predictive, so it should be applied as a matter of course; then it is easy to use the analytical methods to find which of the many different (raw and recalibrated) versions is performing best for the data of interest.

Problems

4.1 *a.* Give two reasons why techniques for analyzing the results of applying software reliability models are needed for use with each new data set to which they are applied.

b. State the main objective of practical interest of these techniques.

c. Briefly describe the general approach to predicting reliability that we are advocating in this chapter.

4.2 Explain what is meant when we say that a sequence of reliability predictions is

a. Optimistic

b. Pessimistic

c. Consistently biased

d. Noisy (compared with the truth)

4.3 Discuss all the relative advantages and disadvantages of the following techniques for analyzing predictive accuracy:

a. The variability as defined in Sec. 4.3.1.
b. Techniques which compare sequences of point predictions of the time between failures T_j with the (later observed) time between failures t_j.
c. Prequential likelihood ratio
d. The u-plot
e. The y-plot

4.4 Consider the following two prediction systems. Assume that the time to next failure, T_j, is exponentially distributed with failure rate

$$\hat{\lambda}_j = \frac{n}{\sum_{r=j-n}^{j-1} t_r}$$

and that for prediction sequence A, $n = 1$, and prediction sequence B, $n = 20$.

In the presence of data for which each time between failures is exponentially distributed and which exhibits reliability growth, discuss what you think the nature of the errors would be in the two prediction sequences suggested here. Describe how these errors are likely to be shown in the u- and y-plots and the prequential likelihood ratio.

4.5 Briefly describe the recalibration technique. Under what circumstances would you expect this technique to *eliminate* inaccuracies in a sequence of raw model predictions?

4.6 Briefly discuss some limitations of the techniques for analyzing accuracy and the recalibration technique, as described in this chapter.

4.7 Data set CSR2 is a subset of CSR1 where only failures that are known to be due to software faults are considered. Perform an analysis on CSR2 similar to the one applied to CSR1 in Sec. 4.5.

4.8 Data set CSR3 is another subset of the data previously analyzed in this chapter (CSR1), but this time failures related only to Pascal programming are included. Tables 4.5, 4.6, and 4.7 (in the Data Disk) show raw and recalibrated one-step-ahead predictions of T_i, $i = 66, \ldots, 104$ that result from applying three models, JM, DU, and KL.

a. Draw plots of the *raw* median predictions against the prediction stage i for these three models. Discuss these plots.
b. Draw the u-plots, and calculate the *KS* distances, for the three *raw* prediction sequences. From Table 4.8, in the Data Disk, say which of these plots is significantly far from the line of unit slope according to these *KS* distances. Based on these u-plots *only,* state which model is giving the most accurate *raw* predictions and which is giving the least accurate predictions. Comment on what the shape of the u-plots for the models which are giving inaccurate predictions tells us about the *nature* of the *raw* prediction errors in each case.

c. Choosing the raw DU model as the reference against which to compare, draw the log(PLR) plot, as shown in the previous examples for these raw prediction sequences (i.e., JM versus DU and KL versus DU). According to this PLR analysis, which raw model is the most accurate, and which is the least accurate? Does this analysis confirm the previous u-plot analysis?

4.9 For the data set CSR3 and the predictions in Tables 4.5, 4.6 and 4.7:

a. Draw the median plot (again against i) for the three sequences of *recalibrated* predictions. Comment on these plots in comparison with the equivalent raw median plots.

b. Draw the u-plots, and calculate the *KS* distances, for the three *recalibrated* prediction sequences. As before, from Table 4.8, say which of these plots is significantly far from the line of unit slope according to these *KS* distances. Comment on these plots in comparison with the equivalent raw u-plots. Discuss whether recalibration has effectively eliminated bias initially present in the raw prediction sequences.

c. Choosing the DUS model as the reference against which to compare, draw the log(PLR) plot as shown in the previous examples for these recalibrated prediction sequences (i.e., JMS versus DUS and KLS versus DUS). According to this PLR analysis, which recalibrated prediction sequence is the most accurate, and which is the least accurate? What does a comparison of this PLR analysis with the equivalent PLR analysis for the raw models suggest?

d. Draw log(PLR) plots for the recalibrated versus the raw prediction sequences (i.e., JMS versus JM, DUS versus DU, and KLS versus KL). Discuss whether these plots show that recalibration has made the predictions more accurate. According to these plots, which model shows the most improvement via recalibration, and which shows the least?

4.10 According to the analyses and plots in Probs. 8 and 9, which of the six predictions shown in Tables 4.5, 4.6, and 4.7 would you choose to use for the next one-step-ahead prediction (i.e., of T_{105} for the data set in CSR3)? Discuss why.