Predicting Fault Detection Effectiveness

J. A. Morgan & G. J. Knafl
School of Computer Science,
Telecommunications & Information Systems
DePaul University
243 S.Wabash Ave.
Chicago, IL 60604
{cphdjam,knafl}@cs.depaul.edu

W. E. Wong
Bell Communications Research
Morristown, NJ 07962
ewong@bellcore.com

Abstract

Regression methods are used to model fault detection effectiveness in terms of several product and testing process measures. The relative importance of these product/process measures for predicting fault detection effectiveness is assessed for a specific data set. A substantial family of models is considered, specifically, the family of quadratic response surface models with two-way interaction. Model selection is based on "leave one out at a time" crossvalidation using the predicted residual sum of squares (PRESS) criterion.

Prediction intervals for fault detection effectiveness are used to generate prediction intervals for the number of residual faults conditioned on the observed number of discovered faults. High levels of assurance about measures like fault detection effectiveness (residual faults) require more than just high (low) predicted values, they also require that the prediction intervals have high lower (low upper) bounds.

1. Introduction

Wong et al. [5,6] conduct an experiment in which they seed ten commonly used Unix utilities with faults in order to determine the relative importance of test set size and coverage for assessing fault detection effectiveness, that is, the percentage of faults discovered. Horgan et al. [2] and Piwowarski et al. [4] discuss related issues and suggest that there is a relationship between fault detection effectiveness and coverage measures. They provide data to support their conclusions but do not conduct extensive analyses of their data.

We use the data analyzed in [6] to investigate a related issue. We use regression methods to model fault detection effectiveness as a response variable in terms of several explanatory product and testing process measures. Testing process measures available are test set size and various coverage measures, including block, decision, and all-uses coverage. Product measures available are lines of code as well as blocks, decisions, and all-uses counts, possibly normalized by lines of code. We consider quadratic response surface models with interaction and apply a model selection algorithm based on the predicted residual sum of squares (PRESS) criterion [3]. We conduct a sensitivity analysis as in [3] to assess the relative importance of these various product/process measures for predicting fault detection effectiveness. We also assess the precision of the resulting predictions.

The remainder of this paper is organized as follows. Section 2 provides an overview of basic concepts and terms. Section 3 presents our model selection approach. Section 4 describes the data and the results of our analyses. Section 5 discusses how these methods may be applied in industrial settings. Section 6 discusses the issue of measurement validity versus predictive validity. Section 7 concludes the paper with a summary of our results.

2. Concepts and terms

Details of how fault detection effectiveness is defined, of how test sets are constructed and how blocks, decisions, all-uses, and their corresponding coverage measures are defined and measured are given in [5,6]. For completeness, we provide a brief description of these concepts and terms.

Fault detection effectiveness: For a test set T and a program P the fault detection effectiveness of T for P is defined as the proportion of total faults in P that can be detected by T when T is executed on P.

Test set: A set of test cases. Each test of a program invokes all test cases in a test set. An observation in our data set corresponds to the outcome of a test set applied to one of the ten Unix utilities in the experiment. Test set size is the total number of test cases in each test set.

Block coverage: A block is a code fragment consisting of a sequence of consecutive statements without branches. Block coverage for a test set is the percentage of the total number of blocks in the program exercised by the test set.

Decision coverage: A decision is a possible branch of a decision structure. For example, an if statement will have two decisions associated with it, while a case statement will have as many decisions as there are cases. Decision coverage for a test set is the percentage of the total number of decisions in the program exercised by the test set.

All-uses coverage: The term all-uses refers to either a predicate (p-use) or computational (c-use) definition-use pair. A p-use indicates that the defined variable was used in a predicate and a c-use indicates that it was used in a computational expression. P-use coverage for a test set is the percentage of the total number of p-uses in the program exercised by the test set. Similarly, c-use coverage is the percentage of the total number of c-uses exercised by the test set. All-uses coverage is the percentage of the total number of p-uses and c-uses exercised by the test set.

3. Model selection

We consider models whose response (or dependent) variable is fault detection effectiveness (FDE), that is, models for predicting fault detection effectiveness. We model this variable in terms of basic product and testing process measures as well as in terms of squares and products of the basic measures. Thus, we consider the class of linear and quadratic response surface models determined by our basic measures. The four basic product measures we consider are all-uses per line of code (AUPL), blocks per line of code (BPL), decisions per line of code (DPL), and lines of code (L). The four basic testing process measures we consider are all-uses coverage (AUC), block coverage (BC), decision coverage (DC), and test set size (S).

The data consist of observations $(\vec{x}_i, y_i)$, for $1 \le i \le I$. The subscript i distinguishes between applications of particular test sets to specific Unix utilities. The explanatory variable vector $\vec{x}_i$ is the corresponding vector of observed product and testing process measures. The response variable value y_i is the associated fault detection effectiveness (FDE). A model M specifies the subset of explanatory variables to use in estimating the linear relationship $E(y_i) = \vec{\beta} \cdot \vec{x}_i$ between the expected value of the response variable and the inner product of the model parameters and the explanatory variables. An intercept parameter may be included in the model by adding an initial component value of 1 to all explanatory variable vectors. It is possible that the subset of explanatory variables selected is empty in which case the constant model is most appropriate. The constant model is the model with only an intercept parameter and no explanatory variables, that is, with all vectors $\vec{x}_i$ equal to the 1-dimensional vector with component value 1.

3.1. Crossvalidation

We use the predicted residual sum of squares (PRESS) criterion [3] to distinguish models. In particular, models with lower PRESS scores produce better, that is, more precise overall, predictions of fault detection effectiveness. PRESS is computed as follows. Each observation is deleted in turn and a model M is fit to the remaining observations. The associated prediction $\hat{y}(\vec{x}_i; i; M)$ at the $\vec{x}_i$ value for the deleted observation is compared to the actual y_i value for that observation. The PRESS score for a model M is the sum of squares of the resulting deleted prediction errors, that is,

$$\text{PRESS (M)} = \sum_{i=1}^{I} \left(y_i - \hat{y}\left(x_i; i; M\right)\right)^2.$$

PRESS is a special case of the statistical technique called crossvalidation, in particular, it is a "leave one out at a time" crossvalidation. It is provided as an option by the standard regression procedure of the Statistical Analysis System (SAS®) which we used in our computations.

An alternative crossvalidation approach involves the random choice of one or more (often just one) training sets on which to fit the models M together with the comparison of consequent predictions with actual values for the remaining observations. The advantage of PRESS is that it is based on predictions for all of the data rather than for a small number of randomly chosen subsets.

Crossvalidation provides a measure of the predictive performance of a model in the sense that it measures how well the model predicts selected response values using estimates based on other observations than those that are being predicted. Thus the observations being predicted cannot influence the computation of their predictions as they can in goodness of fit measures like R^2, that is, the proportion of explained variation.

R^2 and related measures of correlation are inappropriate for model selection since they increase with the number of variables in the model. An adjusted R^2 measure, which does not always increase with the number of variables, is sometimes used in model selection but the adjustment does not account for a model's ability to predict individual response values from the other observations as does crossvalidation. Furthermore, R^2, as well as adjusted R^2, cannot be used to distinguish between models with the same variables but with or without an intercept parameter. Crossvalidation may, however, be used to distinguish between such models. Since the crossvalidation score, unlike R^2 and adjusted R^2, is computed in the same way for models with an intercept as for those without one. In any case, we find that models chosen by crossvalidation have relatively large values of R^2.

Stepwise regression procedures are sometimes used for model selection, often by choosing the variables through a comparison of p-values for a series of t tests for zero slope parameters. The computation of these p-values does not account for the fact that multiple comparisons are involved and is based on approximate normality for the data at least for small sample sizes. Crossvalidation is based on minimization of a numerical score over a selection of models and so its inherent meaning is not dependent on distributional assumptions, as is the computation of a p-value, nor does it require adjustment for joint significance. However, model assumptions including distributional assumptions still need assessment, but this is only necessary for competitive models, that is, those with low crossvalidation scores. Models with much higher crossvalidation scores are inappropriate to use, and so there is no need to assess their assumptions. We however do not assess model assumptions for competitive models in this paper.

3.2. Model selection algorithm

We consider the constant model as well as a variety of models M determined by sets of explanatory variables both with and without an associated intercept parameter. Each of these explanatory variables corresponds either to one of the basic product and testing process measures listed earlier or to products or pairs of those measures. We start with an initial current model M_c corresponding to the constant model and an initial pool P_c of explanatory variables corresponding to some subset of basic product and/or testing process measures of interest. Actually, we only consider three possible initial pools: all product measures separately, all testing process measures separately, and all product and testing process measures combined.

At any iteration, the current set M_c of explanatory variables consists of a subset B_c of basic variables included in the currently selected model and a subset Q_c of quadratic and/or two-way interaction terms also included in that model. The currently selected model is based on the variables of M_c and possibly an intercept parameter and has the lowest PRESS score among all models considered so far. The current pool P_c consists of basic variables from the initial pool that are not yet included in the currently selected model.

The current best basic variable x_c from the pool P_c is the one for which PRESS is minimized over the models $M_c \cup \{x\}$ with $x \in P_c$ with or without an intercept included. For the fault detection effectiveness experiment data, including an intercept is almost always better than not, but this will not be the case in general.

The best model for the next iteration is the one with smallest PRESS score over all models of the form $M_c \cup X$ for $X \subset X(x_c, B_c)$ where

$$X(x,B)=\{x,x^2\} \cup \{x \cdot y : y \in B\},$$

considered with and without intercept terms. Thus, the newly selected model is the best model based on the variables of the previously selected model, basic as well as higher order, together with a specific subset of the set of variables that includes the best variable to add to the previously selected model and degree 2 terms based on that variable, possibly interacting with one of the basic variables of the previously selected model.

We stop the algorithm when the PRESS score for $M_c \cup \{x_c\}$ is larger than the PRESS score for M_c or when the pool is empty. It is also possible to stop when the PRESS score of $M_c \cup \{x_c\}$ is no smaller than some large percentage of the PRESS score for M_c, that is, when the improvement, if any, is of limited size, but we do not consider such variations of the algorithm in this paper.

Once a best model M^* is determined by our algorithm, a sensitivity analysis of the relative impact of variables x in that model may be conducted. If the PRESS score for the model $M^* \setminus \{x\}$ based on all the variables of the best model other than x is much higher than the PRESS score for M^*, then the variable x is an important predictor of the response variable. If the change in PRESS score is marginal, less than 10%, then the variable x is of marginal value for predicting the response variable. If the variable with the smallest impact on the predictions has a negligible effect, then its removal results in a suboptimal model with a simpler structure. This process may be iterated as long as the PRESS score of the reduced model is within a small percentage, for example 5%, of the PRESS score for M^*. The choice of a cutoff value of 5% is subjective. Moreover, this quantity should not be confused with a significance level for a hypothesis test. Significance levels require distributional assumptions or large sample sizes for justification. In the regression setting, they also require that other assumptions hold, for example, unbiasedness and constant spread in the population errors. These assumptions are likely to be inappropriate for most models under consideration, but are not required to assess their predictive performance through their PRESS scores.

4. Data analysis

Table 1, reproduced from [5], identifies the ten Unix utilities used to conduct the fault detection effectiveness experiment and lists values for associated product measures.

Table 2 summarizes aspects of the distributions of testing process measures for all these utilities. The data analyzed in this paper were collected by testing these Unix utilities after seeding them with faults as discussed in [5]. A total of 5400 test sets were executed, 540 per utility, 60 each for the 9 test set sizes ranging from 2 to 10 test cases. These data contain information on coverage, size, and fault detection effectiveness. The authors of [6] consider fault detection effectiveness for four categories of faults classified by difficulty, as well as for all types of faults combined. We restrict our attention in this paper to all types of faults combined.

We choose to normalize decisions, blocks, and all-uses, by dividing by lines of code so that they are comparable across all programs. Since lines of code is one of our basic product measures and since we consider products of pairs of basic product measures, all unnormalized product measures are also considered by our model selection procedure. This procedure starts with the basic variables identified in section 3 and searches for a model with best predictive performance within the family of quadratic response surface models with two-way interaction based on those basic variables.

4.1. Modeling product/testing process measures separately

We first apply the algorithm of section 3 to product measures alone and find that the model for FDE with best predictive performance is based on the explanatory variables L, BPL, BPL·L, DPL, DPL·L, AUPL, and AUPL·L. This model has a PRESS score of 232.06. Its predictive performance may be assessed by comparing that score to the PRESS score of 443.09 for the constant model, a 48% decrease. Thus, the above best model provides a 48% better overall job of predicting fault detection effectiveness than no model at all. Furthermore, its PRESS score may be compared to models based on subsets of its variables. For example, the model obtained from it by removing all of its higher order terms, that is, by removing the interaction terms corresponding to the products BPL·L, AUPL·L, and DPL·L, has a PRESS score of 259.22. Since the best model's PRESS score is 10% lower, there is a noticeable benefit to including interaction terms in the model.

We next apply the algorithm to testing process measures alone and discover that the model for FDE with best predictive performance is based on the explanatory variables DC, BC, BC^2, AUC, AUC·DC, S, and S·BC. This model has a PRESS score of 182.47 which is 59% better than the constant model and 21% better than the best model with only product measures. However, it is only 3% better than the model with only its basic process measures, that is, S, DC, BC, and AUC, but no higher order terms (PRESS=188.04). Therefore, there is little benefit to including higher order terms in this model. Also, the best model does not include an intercept term. However, the corresponding model with an intercept has a PRESS score of 182.48 indicating that both models have equivalent predictive performance.

Removing S and its interaction with BC produces a PRESS score of 183.32, an increase of less than 1%, indicating that test set size S is not an important contributor to predictive performance, supporting the findings of [5,6]. This does not mean that fault detection effectiveness does not depend on test set size, but that coverage measures explain all but a minor portion of that dependence. Larger test set sizes need not result in larger fault detection effectiveness unless associated with some indication of being more productive.

Note that we do not report estimated regression functions in this or later sections. Reporting estimated intercept and slope parameter values is likely to give the misleading impression that they may be used in a general setting while it is only our analysis procedure that applies generally (as discussed in section 5).

4.2. Modeling product/testing process measures jointly

We now apply the algorithm of section 3 to product and testing process measures combined and find that the model for FDE with best predictive performance is based on the explanatory variables DC, BPL, BPL^2, DPL, DPL·BPL, L, BPL·L, S, S·DC, APL, APL·L, AUC, APL·AUC, BC, and BC·APL. This model has a PRESS score of 90.49 which is substantially better than the best models based on product measures alone or testing process measures alone. Thus, fault detection effectiveness depends on both product and testing process measures in a dramatically interactive fashion.

Notice once again that the best model includes test set size S as an independent variable. However, removing S and its interaction with DC produces a PRESS score of 91.92, an increase of less than 2%, indicating as before that test set size is not an important contributor to predictive performance.

A sensitivity analysis of the impact of the individual independent variables of the above model indicates that not all of those variables are crucial to predicting fault detection effectiveness well. In particular, removing a set of independent variables (see section 3.2) with combined effect on the PRESS score of less than 5% produces a suboptimal model with PRESS score 94.76. This model is based on the explanatory variables DC, BPL, DPL·BPL, L, BPL·L, APL·L, AUC, APL·AUC, BC, and BC·APL. Thus, the combined effect of the explanatory variables that are removed, that is, S, S·DC, DPL, BPL^2, and APL, is marginal, contributing very little to predictive performance. Removal of any of the remaining variables causes large (well over 15%) increases in PRESS scores.

This latter suboptimal model is a natural model to use for predicting fault detection effectiveness. The resulting 95% prediction intervals have a size approximately $\pm 2 \cdot \hat{\sigma}$, where $\hat{\sigma}$ is the estimated spread about the regression surface. In this case, $\hat{\sigma}=0.13$, about the same value as for the best model. Thus, fault detection effectiveness is predicted to within ±0.26, with 95% confidence. This interval is quite large, indicating that we are unable to predict fault detection effectiveness precisely in this case.

4.3. Modeling product/testing process measures jointly with high levels of coverage

Horgan et al. [2] suggest that highly precise predictions should be possible for high levels of coverage, suggesting at least 70% of block coverage as a rule of thumb. In support of this, they present industrial data for which fault detection effectiveness is 100% whenever block coverage is at least 70%. Piwowarski et al. [4] suggest the same coverage threshold, also basing their suggestion on industrial data. They address function testing and conclude that 70% statement coverage is the minimum threshold required to assure that test cases sufficiently exercise the software.

For this reason, we apply our model selection algorithm to product and process measures when the fault detection effectiveness data is restricted to those test sets which achieve at least 70% block coverage. The best model for this subset of the data, considering both product and process measures, is based on the explanatory variables DC, L, DC·L, BPL, BPL^2, DPL, DPL·BPL, BC, BCV·DPL, APL, APL·DPL, AUC, and AUC^2. The corresponding estimate of standard deviation is $\hat{\sigma}=0.12$, just a little smaller than for modeling all coverage levels. Similar results are obtained for restrictions to at least 70% decision or all-uses coverage. The results for larger coverage thresholds than 70% are not much different.

Unlike Horgan et al., we find that fault detection effectiveness cannot be predicted precisely, even at high levels of coverage. From Figure 1, it is apparent that, for the Wong et al. data, fault detection effectiveness values, for high levels of coverage, span a broad range, from a low of nearly 0% to a high of 100%. This distribution of values may not be typical of industrial settings. The Horgan et al. data is an example of an industrial setting where this is not the case. However, the Wong et al. data indicates the possibility that high levels of coverage may not always result in high levels of fault detection effectiveness.

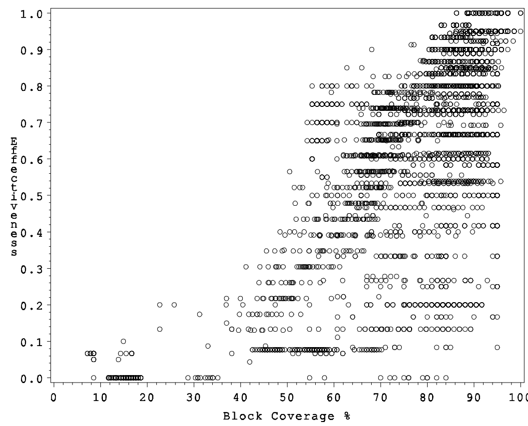

Figure 1: FDE vs. Block Coverage

5. Application to industrial testing situations

In practice, few software development organizations will collect data generated through seeded faults like those analyzed in this paper. However, the methods described in this paper may still be applied in industrial testing situations. Suppose that it is desired to predict the fault detection effectiveness of the system test phase of a particular software product. Assume that historical project data, for example, like that of Horgan et al., are available. Assume also that these data consist of various product/process measures at certain stages of the development cycle as well as the number of faults discovered at those stages. The analysis techniques of this paper may be applied to such data. The resulting model for each such stage may then be used to predict fault detection effectiveness for new software products as long as all the explanatory variables of these models have been recorded for the new software products. Note that for projects where discovered faults (DF) is zero, fault detection effectiveness (FDE) is also zero if residual faults (RF) is greater than zero or else FDE is naturally taken to be one since then both DF and RF are zero. Standard regression procedures will not handle such dichotomous situations very well, and so we assume that DF>0 in what follows.

Suppose the new software product has just undergone system test. Data for past software products may be used to generate a prediction $\widehat{FDE}$ of fault detection effectiveness FDE for the new software product together with a 95% prediction interval [L,U] for that product's FDE. The prediction $\widehat{FDE}$ may then be used to generate a prediction $\widehat{RF}$ for the number of residual faults RF in the product not removed through testing.

The 95% prediction interval for FDE may also be used to generate a 95% prediction interval for RF conditioned on the number of discovered faults DF identified through testing. In particular, since

$$\mathrm{FDE} = \frac{\mathrm{DF}}{\mathrm{DF}+\mathrm{RF}},$$

RF may be expressed in terms of FDE and DF by rearranging terms:

$$\mathrm{RF} = \left(\frac{1-\mathrm{FDE}}{\mathrm{FDE}}\right)\cdot \mathrm{DF}.$$

Thus, predictions of RF may be obtained from predictions of FDE and the observed value DF, that is,

$$\hat{\mathrm{RF}} = \left(\frac{1-\hat{\mathrm{FDE}}}{\hat{\mathrm{FDE}}}\right)\cdot \mathrm{DF}.$$

Furthermore, lower and upper bounds for RF may be obtained in a similar manner from the bounds L, U on FDE producing the 95% prediction interval for RF conditioned on the value of DF:

$$\left[\left(\frac{1-\mathrm{U}}{\mathrm{U}}\right)\cdot \mathrm{DF}, \left(\frac{1-\mathrm{L}}{\mathrm{L}}\right)\cdot \mathrm{DF}\right]$$

Notice that large upper bounds U on FDE allow for the possibility of low values of RF, in particular U=1, corresponds to a lower bound of zero on RF. However, the lower bound L on FDE is the crucial factor in achieving high levels of assurance that FDE is large or that RF is small. Notice also that the upper bound on RF depends on both the lower bound L on FDE as well as the actual number of discovered faults DF. This upper bound on RF accounts for variability in FDE but is conditioned on the observed value of DF.

To justify the above prediction interval for RF, assume DF>0 and let $\vec{\mathrm{X}}$ denote a vector of product and testing process measures and note that the probability that RF is in that interval, conditioned on the observed values for $\vec{\mathrm{X}}$ and for DF, is

$$\begin{aligned} \mathrm{P}\left(\left(\frac{1-\mathrm{U}}{\mathrm{U}}\right)\cdot \mathrm{DF} \le \mathrm{RF} \le \left(\frac{1-\mathrm{L}}{\mathrm{L}}\right)\cdot \mathrm{DF} \,\middle|\, \vec{\mathrm{X}}, \mathrm{DF}\right) &= \mathrm{P}\left(\frac{1-\mathrm{U}}{\mathrm{U}} \le \frac{\mathrm{RF}}{\mathrm{DF}} \le \frac{1-\mathrm{L}}{\mathrm{L}} \,\middle|\, \vec{\mathrm{X}}, \mathrm{DF}\right) \\ &= \mathrm{P}\left(\frac{1-\mathrm{U}}{\mathrm{U}} \le \frac{1-\mathrm{FDE}}{\mathrm{FDE}} \le \frac{1-\mathrm{L}}{\mathrm{L}} \,\middle|\, \vec{\mathrm{X}}, \mathrm{DF}\right) \\ &= \mathrm{P}\left(\mathrm{L} \le \mathrm{FDE} \le \mathrm{U} \,\middle|\, \vec{\mathrm{X}}, \mathrm{DF}\right). \end{aligned}$$

Assume that, if $\vec{\mathrm{X}}$ predicts the fault detection effectiveness FDE of test sets well, it will also determine the number of discovered faults DF resulting from those test sets to the extent that it is reasonable to assume that $\mathrm{DF}|\vec{\mathrm{X}}$ is a deterministic function of $\vec{\mathrm{X}}$ or at least has relatively negligible variability. Under this assumption, the above probability equals $\mathrm{P}(\mathrm{L} \le \mathrm{FDE} \le \mathrm{U}|\vec{\mathrm{X}})$ which is 0.95 by choice of the bounds L and U. This assumption is not required to apply the above argument to conclude that the induced interval for the ratio of RF to DF is a 95% prediction interval, since then the conditioning would be on $\vec{\mathrm{X}}$ by itself and not on DF.

For industrial data sets, it may be reasonable to include DF within the vector $\vec{\mathrm{X}}$ of predictor variables, and then the calculation of the prediction level for RF does not require any assumptions about DF. The reason we do not include DF as a predictor of FDE in our analysis is that its inclusion produces an essentially deterministic prediction system [3]. While our data set has 5400 observations, these correspond to only 10 programs and hence to only 10 distinct fault counts DF+RF, thus FDE/DF is only one of 10 distinct values which may be fit perfectly by including enough covariates in $\vec{\mathrm{X}}$. It may be that this would not be a problem when analyzing testing data for larger numbers of industrial systems.

6. Measurement vs. predictive validity

Fenton [1] distinguishes between the validity of a measure and the validity of a prediction system, that is, how well a measure accurately characterizes its associated attribute versus how precisely the measure may be used to predict measures of related attributes. For example, testing effort is an attribute of the testing process. Test set size and various coverage metrics are measures of that attribute. Their validity as measures of testing effort is separate from their validity for predicting fault detection effectiveness, a measure of the usefulness of the testing process. Fenton asserts that the measurement validity of measures like test set size and coverage is a necessary prerequisite for their validity as predictors of other measures such as fault detection effectiveness.

A more valid measure for an explanatory attribute may in fact provide relatively more precise predictions of a measure for a related response attribute than do other less valid measures of the same explanatory attribute. However, this is no guarantee that sufficiently precise predictions of the response measure will be produced. Measurement validity alone is not sufficient for high levels of predictive validity. Test set size and coverage measures seem to be reasonably valid measures of testing effort. However, for the data we analyze, these measures, even when combined with an extensive set of product measures, produce fairly imprecise predictions of fault detection effectiveness.

Such low levels of precision may be an indication that the test cases used in the Wong et al. experiment represent a fairly wide spectrum of testing situations, quite possibly much broader than most industrial testing situations. On the other hand, our observations are consistent with one of the case studies of Horgan et al. [2], the one for which the same test set is executed on twelve versions of an autopilot system developed by different programming teams. For that case study, they report a weak correlation between block coverage and faults detected for each program version and conclude that, since each version has a different fault distribution, coverage measures may not be good predictors of the absolute number of program faults. In any case, high levels of assurance about predictions of fault detection effectiveness may require additional measures that capture other facets of testing effort than do test set size and coverage.

7. Summary and conclusions

Models based on testing process measures alone are better at predicting fault detection effectiveness than models based on product measures alone. Also, substantial gains in predictive performance result from models with both product and testing process measures. Thus, testing process measures are more important for predicting fault detection effectiveness than product measures are, but testing process measures do not fully explain the aspects of fault detection effectiveness that are explained by product measures.

On the other hand, we find that fault detection effectiveness cannot be predicted with a great deal of precision. When block coverage exceeds 70%, the 95% prediction interval for fault detection effectiveness, using the best model of section 4.3, spans a range of about ±24%. For example, suppose that a software product with specific product characteristics is tested to a certain level of coverage with a predicted fault detection effectiveness of 80%, that is, the percentage of residual faults is predicted to be 20%. With 95% confidence, the true level of fault detection effectiveness may be as low as 56%, that is, the percentage of residual faults could be as much as 44%, more than double its prediction. This suggests that high levels of assurance about measures like fault detection effectiveness will require that the lower bounds of associated prediction intervals, rather than the predictions themselves, be above a target level.

These results hold for the fault detection effectiveness data analyzed in this paper. It is not certain that they hold in more general settings, but it is reasonable to expect them to apply to programs with product/process measure values within the span of values for these data and with faults similar in nature to those seeded by the authors of [5,6]. Other product/process measures may be useful to include in general. For example, faults discovered during testing and process measures related to other development phases than testing are not considered in this paper, but are likely beneficial. Moreover, other measures of product size than lines of code, for example, function points, may be more natural to use to normalize the other product measures. In any case, the methods described in this paper may be employed in predicting fault detection effectiveness from historical data, but the appropriate models will depend on the number of tested systems as well as the available explanatory measures and perhaps also on the testing organization.

References

[1] N. E. Fenton, "*Software Metrics: A Rigorous Approach*," Chapman & Hall, London, 1991.

[2] J. R. Horgan, S. London, and M. R. Lyu, "Achieving Software Quality with Testing Coverage Measures," *Computer*, September 1994, pp. 60-69.

[3] J. A. Morgan and G. J. Knafl, "Residual Fault Density Prediction using Regression Methods," *Proceedings of the Seventh International Symposium on Software Reliability Engineering*, White Plains, NY, October 1996, pp. 87-92.

[4] P. M. Piwowarski, M. Ohba, and J. Caruso, "Coverage Measurement Experience During Function Test," *Proceedings of the Fifteenth International Conference on Software Engineering*, Los Alamitos, CA, May 1993, pp. 287-301.

[5] W. E. Wong, J. R. Horgan, S. London, and A. P. Mathur, "Effect of Test Set Size and Block Coverage on the Fault Detection Effectiveness," *Proceedings of the Fifth International Symposium on Software Reliability Engineering*, Monterey, CA, November 1994, pp. 230-238.

[6] W. E. Wong, J. R. Horgan, S. London, and A. P. Mathur, "Effect of Test Set Minimization on Fault Detection Effectiveness," *Proceedings of the Seventeenth International Conference on Software Engineering*, Seattle, WA, May 1995, pp. 41-50.

Table 1. Characteristics of Unix utilities [5]

Utility	# Faults	LOC†	Blocks	Decisions	All-Uses
CAL	20	163	96	50	180
CHECKEQ	20	90	74	69	170
COL	29	274	154	104	374
COMM	15	144	100	69	110
CRYPT	17	121	69	39	113
LOOK	13	135	88	52	110
SORT	25	842	508	394	1638
SPLINE	14	289	193	121	400
TR	12	127	101	82	248
UNIQ	18	125	84	58	124

†Lines of code

Table 2. Summary Statistics - Coverage

	Min	Median	Mean	Max
Decision Coverage	3.25	73.21	80.56	94.23
Block Coverage	7.19	80.20	86.17	100.00
All-Uses Coverage	2.87	63.93	68.42	91.13