Making inferences with small numbers of training sets

C. Kirsopp and M. Shepperd

Abstract: A potential methodological problem with empirical studies that assess project effort prediction system is discussed. Frequently, a hold-out strategy is deployed so that the data set is split into a training and a validation set. Inferences are then made concerning the relative accuracy of the different prediction techniques under examination. This is typically done on very small numbers of sampled training sets. It is shown that such studies can lead to almost random results (particularly where relatively small effects are being studied). To illustrate this problem, two data sets are analysed using a configuration problem for case-based prediction and results generated from 100 training sets. This enables results to be produced with quantified confidence limits. From this it is concluded that in both cases using less than five training sets leads to untrustworthy results, and ideally more than 20 sets should be deployed. Unfortunately, this raises a question over a number of empirical validations of prediction techniques, and so it is suggested that further research is needed as a matter of urgency.

1 Introduction

Software project prediction, typically meaning effort prediction, is an important but unfortunately challenging problem for software engineers. Thus it is no surprise that there has been considerable research activity in this area. A lot of this research activity has taken the form of comparing different prediction techniques on data that have been collected from real completed software projects. The goal has then been to try to establish which technique, or techniques, are the most accurate. Over the last ten years or so, most interest has centred around prediction systems that are in some sense local. Such systems are developed, or calibrated, for a particular environment and there is no expectation that they will provide accurate results for other environments or situations. Examples include developing local models using ordinary least squares regression (LSR) [1, 2], artificial neural nets (ANN) [3], case-based reasoning (CBR) [4], rule induction (RI) [5], and fuzzy rule induction [6].

A strength of the effort prediction research community is that workers have not been content merely to propose new techniques, but there has also been significant effort to empirically validate them as well. Validation provides certain challenges, not least the need to both train and validate the prediction system on representative data.

Typically this is accomplished by splitting the available data into two subsets. In this paper we show that although it is widely used, there are potentially serious problems with this procedure. Specifically, the confidence limits that can be attached to a measure of prediction system accuracy may be unacceptable and prevent meaningful comparisons between competing prediction techniques, particularly when the size of the effect is small.

The case study in this paper considers the problems associated with the common practice of splitting a data set into training and validation sets, and illustrates this with a publicly available data set provided by Desharnais [7] and the ANGEL prediction system [8].

2 Short review of software effort estimation

As suggested by the introduction, there has been substantial interest, and consequently research, into the problem of predicting software costs, principally effort, at an early stage in a project. Early work included attempts to fit simple nonlinear models to data collected, such as the research by Walston and Felix at IBM in the mid 1970s [9]. Also at this time, various general purpose prediction systems were popularised, the best known being COCOMO. An important development was the work carried out by Kitchenham and Taylor [10] and Kemerer [11]. In both cases the researchers sought to independently assess various general purpose prediction techniques such as COCOMO [12] and Function Points [13] on data sets other than those on which they had been developed. Kemerer, in particular, endeavoured to establish some sort of order of preference between the four techniques under investigation using the mean magnitude of relative error (MMRE) as an accuracy indicator. Many other studies followed, all with the same general objective of providing evidence to show which, of many, prediction techniques were the most accurate.

More recently, however, the majority of prediction techniques have focused on building local systems that are fitted to a particular dataset. This is largely in response to the considerable difficulties of successfully using more universal approaches without substantial adaptation or calibration activity. See, for example, the study conducted by Miyazaki and Mori [14] who demonstrated the positive

IEE Proceedings online no. 20020695

DOI: 10.1049/ip-sen:20020695

Paper first received 19th April 2002 and in revised form 30th July 2002

The authors are with the Empirical Software Engineering Research Group, School of Design, Engineering & Computing, Bournemouth University, Royal London House, Bournemouth BH1 3LT, UK

Table 1: Summary statistics of prediction techniques (data taken from [5])

Technique	Sample count	MMRE*			
		Mean	Minimum	Middle	Maximum
ANN	3	47	21	53	66
CBR	3	57	43	49	80
LSR	3	62	38	47	100
RI	3	104	86	87	140
RI (with pruning)	3	90	41	89	141

*Mean magnitude of relative error (MMRE) is an accuracy indicator and is calculated as: $MMRE = 1/n\sum_{i=1}^{n}|(actual_i - predicted_i)/actual_i|$

effect of calibrating the COCOMO model to a local environment, in their case that of Fujitsu. This study has one drawback in that the researchers used the entire dataset, in other words it was a model fitting exercise. This tends to lead to optimistic results, since if the prediction technique were to be used in practice, not all the data would be available as one would be predicting for some future incomplete project. Indeed, building any local prediction system unfortunately has major repercussions on how we evaluate it; namely, we need to be careful not to use the same data for building and for evaluating.

Most empirical research into prediction systems uses some kind of hold-out strategy. Hold-out strategies work by simulating the problem of predicting some future, unknown project by dividing the data set into a training set (that is, data points which are assumed to be known and can therefore be used to develop the prediction system) and a validation set (that is, data points to assess the accuracy of the prediction system). Usually the data points are selected randomly from the underlying data set. Two other techniques to achieve the same aim are the jack-knife and bootstrap. The jack-knife differs from a random hold-out in two ways. First, only one case is placed in the validation set at a time. Second, this is done systematically so that all cases are held-out once. In general this is not widely used as it requires considerable computational effort, since a data set of n cases will require n prediction systems to be developed. The bootstrap differs from a simple hold-out strategy only in that the random sampling is done with replacement, and consequently, the training set may contain multiple copies of the same case. This can be useful in situations where n is small and there is a need to generate many samples. Again, a disadvantage is that this does not seem to fit well with the real-world use of a prediction system, where clearly there will not be multiple copies of the same project (for more details see Efron and Gong [15]).

An example of evaluating prediction systems by randomly splitting the training set into a prediction and training set is a study we conducted, when we sought to compare a number of linear regression models for predicting the size of a 4 GL system using simple measures derived from a data model [2]. The data set comprised 77 complete cases or software projects, which was then randomly divided into a training set of 50 cases and a validation set of 27 cases. The question arises as to what extent did our findings depend upon the random allocation of cases; in other words, suppose a specific case had been differently allocated—would this have made any difference to our conclusions? This question is the focus of our paper.

Because of concern about the sampling process, more recent work from our group [5], aimed at comparing the performance of four different prediction techniques on the same dataset (Desharnais), repeated the sampling process three times. 'The procedure adopted was to randomly partition the dataset into a training set of 67 projects and validation sets of 10 projects. This was performed three times yielding validation sets 1, 2 and 3 so as to help assess the stability of any prediction systems generated' (Mair *et al.* [5]).

Unfortunately, the results from Table 1 indicate a great deal of variability depending upon the choice of training set and the random variability of that choice. In particular, note the large range between maximum and minimum MMRE value. This leaves us vulnerable to rank reversal problems. In other words, depending upon which sample we used, we could conclude that different prediction systems yielded the most accurate results. Clearly this is not a very satisfactory state of affairs. For each technique there is a probability distribution of accuracy values, each value coming from one of the possible combinations of cases in the training set. Rather than comparing values from 1 (or a small sample) of these possible training sets, we should endeavour to identify the 'best' prediction technique based on properties of the distribution of results from all possible training sets.

A related observation derives from a systematic exploration of the interaction between data set properties and prediction system accuracy that we had previously conducted [16]. As part of this work we repeated all sampling processes twice to randomly construct, in each experiment, two different training sets. We then formally tested for significant differences between the pairs of residuals from the validation sets using a Wilcoxon signed rank test and $\alpha = 0.01$. For the small training sets ($n = 20$), 27 out of 32 tests showed significant differences (and note the conservative value of α). For the larger training sets the situation improved to 14 out of 32 differences, although even this is quite alarming. In other words, the results depend upon a random sampling process. This leads us to conclude that there is a need for considerable caution when interpreting results from empirical comparisons of prediction systems.

3 Introduction to case study

We have shown that results from prediction system studies can be highly dependent on the particular training set selected. Results seem to vary significantly from one sampled training set to another. If the dispersion of accuracy values is very large then it is likely that a single sample may wrongly estimate the centre [Note 1] and lead to problems of incorrect inference. Effectively, we may erroneously prefer prediction system A to prediction

Note 1: Whether measured as mean or median.

system B. This means that to have any confidence in inferences made from prediction systems, it must be shown that' outcomes are the result of the underlying property being studied and not just an artefact of the particular training set. The case study provides an empirical exploration of the scale of the problem and of the utility of deploying multiple sampled training sets in validation studies. The example used for the case study is the configuration of the CBR effort prediction system ANGEL [16]. We chose this example not only because it is of some practical interest, but also to illustrate some of the wider methodological issues of how one empirically validates a prediction system.

The ANGEL prediction systems operate as follows. We have n projects or cases, each of which needs to be characterised in terms of a set of p features. In addition, for each project, we must also know the value of the feature that is to be predicted (in this case study, effort). Features can either be continuous (e.g. experience of the project manager), discrete (e.g. number of interfaces) or categorical (e.g. development environment). Historical project data is collected and added to the case base. When a prediction is required for a new project, this case is referred to as the target case. The target case is also characterised in terms of the p features. This imposes a constraint on the feature set in that it should only contain features for which the values will be known at the time of prediction. The next step is to measure the similarity between the target case and other cases in the p-dimensional feature space. The most similar k cases or projects are then used, possibly with adaptation [Note 2], to generate a prediction for the target case. Where CBR is used without any adaptation, of the cases retrieved, this is referred to as a k-nearest neighbour (k-NN) technique. Multiple cases are used to make a prediction because it improves accuracy by averaging-out the variation in similar projects. However, as more projects are included in this average the projects added become less similar to the target case and the average value tends towards the sample mean. This leads to a trade-off in the selection of the k value to use. For all the results described in this case study the prediction is obtained by taking the mean of the target feature values from the k most similar cases.

It has been argued [17], that as the size of the training set increases the optimum number of cases on which to base a prediction will also increase. The argument for k increasing with n is that larger training sets will have larger numbers of cases that will be acceptably close to the target to include in making the prediction. In this case study we are interested in systematically exploring the relationship between n and k to see whether the rational argument given above actually holds empirically.

In order to explore this relationship between k and n we used the Desharnais data set [7]. After cases with missing values are removed, the data set contains 77 cases (or projects). Another issue with case-based prediction is that not all features are necessarily helpful towards the task of prediction, and consequently, using the entire feature set can adversely affect the results. It is common practice, therefore, to pre-test the data in order to select a suitable subset of features that will actually be used to build the prediction system. This procedure resulted in the removal of five features. This leaves us with a data set of 77 cases each with five features.

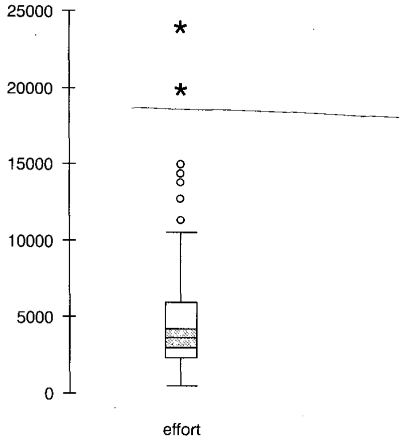

Fig. 1 *Boxplot of project effort for the Desharnais data set*

One characteristic of the Desharnais data set is the presence of a small number of extreme outliers, denoted by stars in Fig. 1. These are defined as exceeding (since the distribution is positively skewed)

$$\text{the upper hinge} + 3.0\ (\text{upper hinge} - \text{lower hinge}).$$

This is a common characteristic of software engineering data sets and clearly leads to vulnerability in the sampling process. Training sets that contain such outliers may generate very different accuracy results from those not containing any of these outliers.

4 Small samples and uncertainty

To empirically assess the optimum value of k to use for CBR-based effort prediction, the accuracy of prediction systems built using the same training set but different k values could be measured. A plot could then be made of k against accuracy using mean absolute residual [Note 3]. However, the results produced by this approach proved to be highly dependent on the particular training set chosen. To demonstrate this, a set of 100 training sets were generated (with $n = 20$) by randomly sampling without replacement from the entire dataset of 77 cases. The results from each of these 100 training sets were plotted. Fig. 2 shows some example forms (or shapes) of results generated from different training sets. This reveals a wide variety of apparently clear functional forms, rather than repeated similar forms or simply random results.

All 100 plots were then classified into the various categories shown in Fig. 2. The categories are somewhat arbitrary as it is only intended to give an idea of the possible variation. The results from this classification are given in the pie chart of Fig. 3. Only around one, sixth of the samples showed no discernible trend. This means that there is a five in six chance of randomly sampling a training set that will show any one of a number of distinct functional forms.

Note 2: Adaptation is the way in which the actual prediction is calculated based on the analogies found. Example adaptation strategies might be inverse distance weighting or inverse rank weighting. More complex rule-based adaptation strategies could also be devised.

Note 3: We choose the mean absolute residual as our accuracy indicator as we do not distinguish between under- and over-estimates, and wish to use a symmetric rather than a relative measure. For a more detailed discussion of the merits and disadvantages of various accuracy indicators see Kitchenham *et al.* [18].

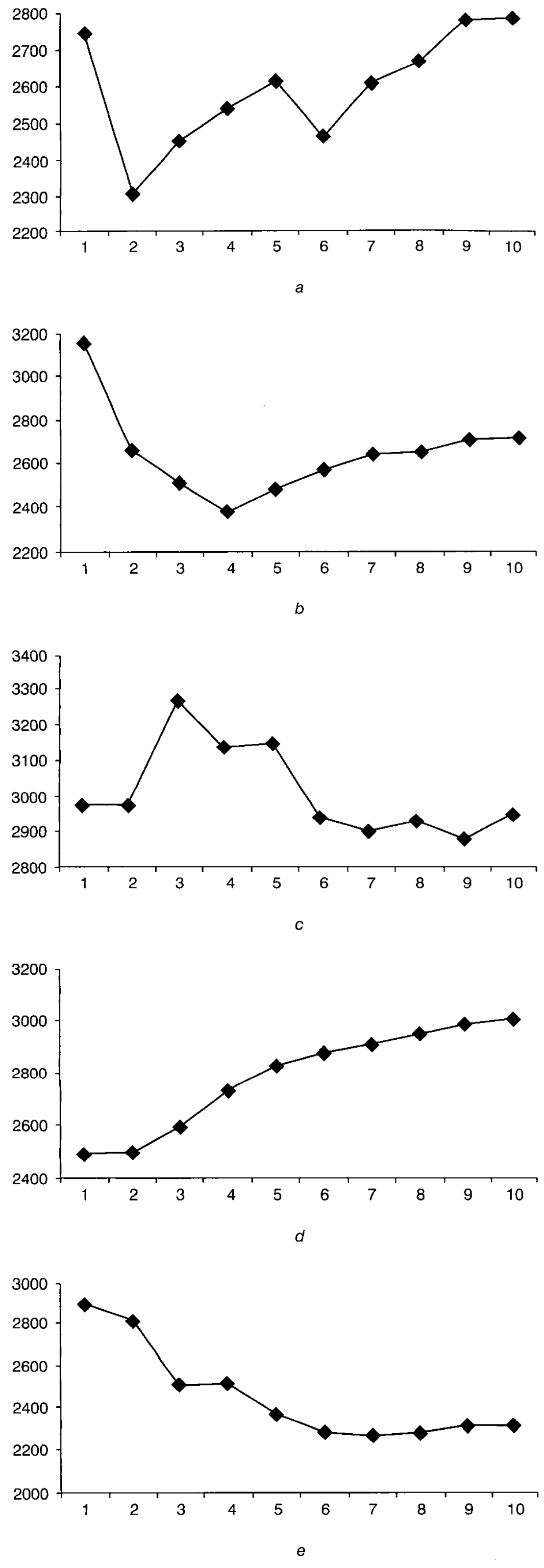

Fig. 2 *Example 'shape types' for mean(|r|) vs k of different samples*

a No clear trend
b Single minima
c Single maxima
d Monotonically increasing
e Monotonically decreasing

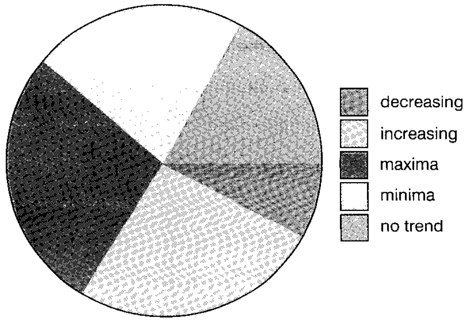

Fig. 3 *Distribution of 'shape-types' for mean (|r|) vs k*

If there is such a range of results from a randomly selected training set, how can any underlying trend be detected? Clearly, using a single training set will not allow any analysis of whether the value is typical of the underlying population of possible training sets that the measure is intended to represent. A common approach to dealing with the problem of variability in measurements is to take large numbers of repeated measurements (or in our specific case, to use large numbers of sampled training sets). This allows the calculation of both central tendencies and confidence limits on the properties being observed.

Fig. 4 shows the relationship of the mean and the 95% confidence limits for the mean absolute residuals as the number of data points (training sets) is increased. The data shown is cumulative, i.e. the datasets used for ten samples are those used for the five-sample point plus an additional five samples; similarly, the ten samples are a subset of the fifteen samples and so on. This data is intended to be an example of what would happen to the confidence limits as more datasets are added, rather than the typical value we might get from using disjoint sample sets for each point. The data indicates how confidently we can estimate where the true centre [Note 4] lies, given the number of training sets used to assess the accuracy of a prediction system. This gives an indication of the number of sampled data sets that should be used to gain a particular level of confidence. Sampling theory also shows that the confidence limits cannot be expected to be accurate for non-normal populations where the number of samples is less than 30 [19]. This can also be seen in the data where the lower bounds on the confidence limits at some points for less than 30 samples are actually higher than the value the mean converges to.

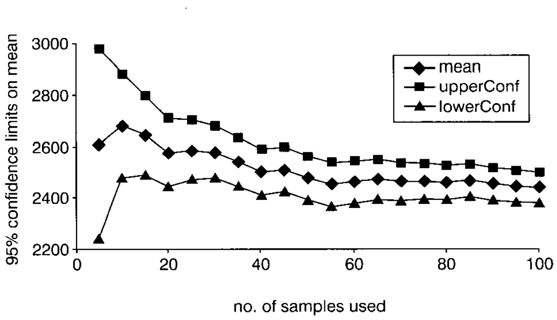

Fig. 4 *Variation of 95% confidence limits with number of samples (n = 30, k = 5)*

Note 4: By 'true centre' we wish to denote the mean of the absolute residuals produced from the population of all possible training sets.

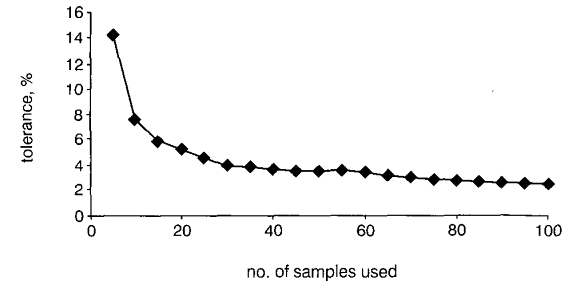

Fig. 5 *Tolerance on the mean absolute residuals vs number of samples used*

It is worth noting that if we compare the left-most range of uncertainty (five samples), it is wider than the entire range of variation shown on most of the examples in Fig. 2. Even if we had taken the average of five samples, we could only say that the data points lay somewhere in the entire height of the graph! This means any shape could be drawn on the example graphs and still lie within the bounds of uncertainty. Such a graph could tell us nothing about the functional form of any relationship in the data.

Fig. 5 expresses the confidence limits as a fractional tolerance given as a percentage. For the given data using five samples we can be 95% confident that the actual value is within ± 15% of the sample mean. With twenty samples this reduces to ± 5%. It is probably only worth going beyond 40 sample if the effect size being observed is very small, since there is usually a substantial effort associated with each validation.

5 Using large number of sampled training sets

An initial study into the relationship between *n* and *k* was first performed with a set of twenty randomly sampled training sets. Although this was sufficient to suggest the structure of the underlying effect, it proved insufficient to gain successful tests of significance. The full study described in this Section uses a set of 100 randomly sampled training sets to build prediction systems for each combination of *k* (1–10) and *n* (10, 20, 30, 40, 50 and 60). This resulted in the building and assessment of 6000 prediction systems (100 × 10 × 6). For the purposes of this study, assessment of the prediction systems is done using the mean of the absolute residuals produced by the prediction systems when applied to the validation sets. The cases used in the $n = 10$ training set were a subset of the $n = 20$ set, $n = 20$ is a subset of $n = 30$, and so on. The same training sets are used for the various values of *k*.

We can summarise the relationships between *k*, *n* and mean($|r|$) either by treating constant *n* (Fig. 6) or constant *k* (Fig. 7) as a series.

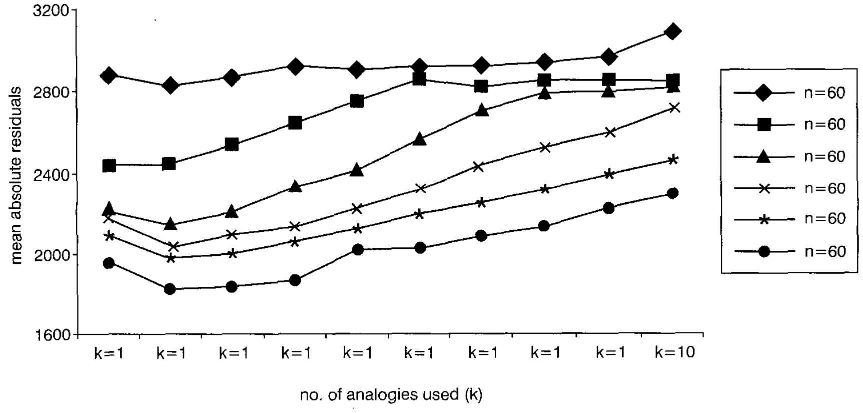

Fig. 6 *Variation of accuracy with k for different n-values*

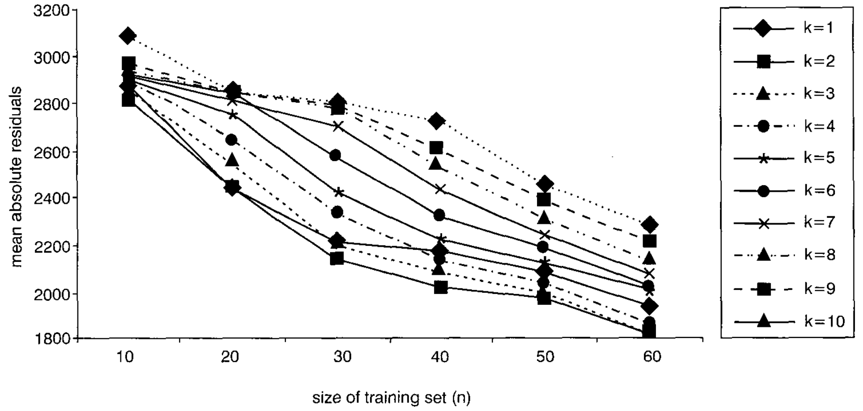

Fig. 7 *Variation of accuracy with n for different k-values*

Table 2: Variation of *k* with *n*

Value of *n*	Optimum *k*
10	$1 \leq k \leq 2$
20	$I \leq k \leq 2$
30	$k=2$
40	$k=2$
50	$k=2$
60	$2 \leq k \leq 3$

Fig. 6 clearly shows the variation of prediction system 'accuracy' with n. Basically, this says that larger training sets lead to better prediction systems (no surprise there). It can also be seen that for all series except $n=20$, there is a minimum value at $k=2$. This trend can be better viewed in Fig. 7. For the majority of the length of the graph the $k=2$ line is the lowest. Visually, $k=2$ is the optimum value that we were seeking, but is it significantly better than the other k values?

Since we have results from a population of prediction systems for each value of n, we can test to see if the k-value with the lowest median is significantly lower than other k-values. We can do this by making a one-tailed Wilcoxon signed rank test between the lowest k-value set and each of the other sets (with $\alpha = 0.05$). For example, for $n=10$, $k=2$ has the lowest median, but it did not prove to be significantly better than $k=1$. From this we can conclude that for $n=10$, the optimum value lies between $k=1$ and $k=2$ (inclusive). If a similar analysis is performed for the other n values we get the results shown in Table 2.

From this data we conclude that two analogies is the optimum value. There is also no significant change in the optimum value of k with variation of n. We would suggest that the uncertainty in the optimum k value for $n=10$ and $n=20$ is due to instability in the prediction systems because of the small size of the training sets. Uncertainty in the optimum k value for $n=60$ may be due to convergence of the residual results as the size of the training set increases (see Finnish dataset results in Fig. 8).

6 Corroborative work

A criticism that could be levelled at the above example is that it only uses a single data set and a single prediction method. In a paper espousing the use of multiple observations this would be particularly worrying. We therefore offer some limited, additional corroborative evidence.

The initial analysis of another (and much larger) dataset from Finland shows very similar results to the Desharnais dataset. Individual sampled training sets can show a range of functional forms. Despite permitting larger training and validation sets, the Finnish dataset still shows large variations in individual results. This leads to a large uncertainty range if less than twenty sampled training sets are used. Fig. 9 shows the variation of uncertainty range for the Finnish dataset against the number of sampled training sets used.

The analysis of n against k was also repeated for the Finnish dataset. It can be seen from Fig. 8 that $k=2$ is optimum across the entire range of n values so far analysed (10, 20, 50, 100, 200 and 350).

The analysis of the Finnish dataset provides a much 'cleaner' set of results. This may be due to the much larger number of cases, or perhaps it suggests that the dataset is more homogenous. Whatever the reason, with this dataset there is also the clear suggestion that as the size of the training set increases the relative importance of selecting the optimum k value reduces. This analysis helps to show that the problem of variability in accuracy results due to training set sampling is not simply an artefact of a single dataset. Can we show that this is not a feature of CBR technology, but is also a problem with other types of prediction systems? The procedure for showing the variation of confidence limits with number of samples was repeated for ordinary least squares regression on the Desharnais dataset (see Fig. 10). These results were obtained by trying to predict actual effort using raw function point counts with a training set size of 30. The variation of confidence limits with number of training sets used is similar to that for CBR. This problem, it would seem, is common to other prediction methods, not just CBR.

Fig. 10 shows the variation of 95% confidence limits for MMRE and Pred(25) as well as mean($|r|$). This was done primarily to show that other accuracy indicators are also affected by this problem (which is clearly the case). Mean

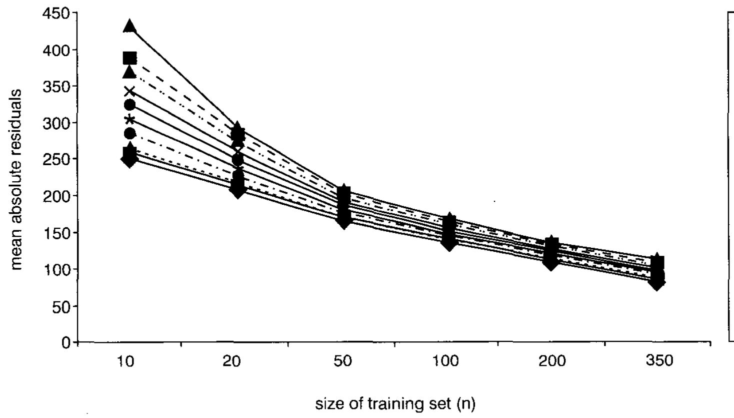

Fig. 8 *Line-plots of mean ($|r|$) vs n for different values of k (Finnish dataset)*

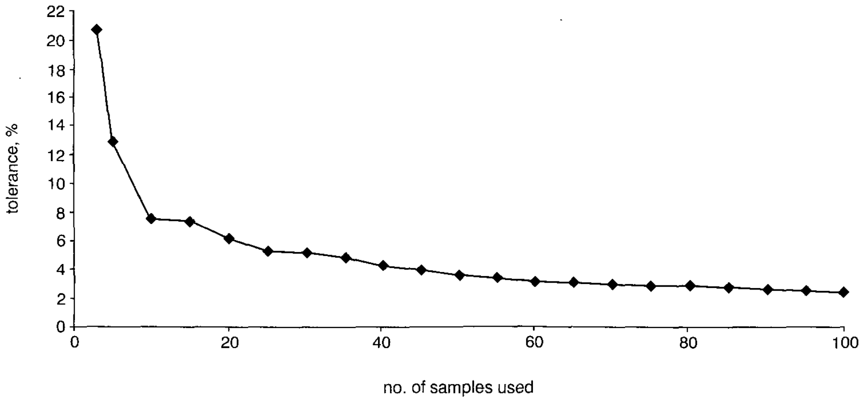

Fig. 9 *Variation of fractional confidence with n (Finnish dataset n = 200, k = 5)*

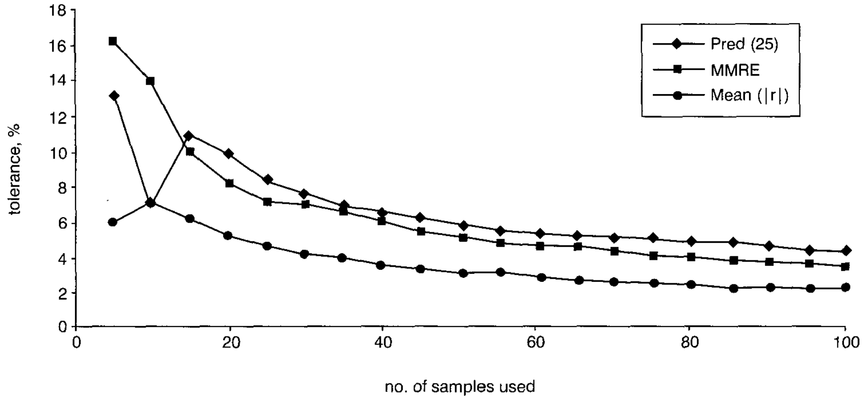

Fig. 10 *Fractional uncertainty for OLS regression with n = 30*

absolute residuals were chosen as the accuracy statistic to use for the worked example because the authors considered residual based measures to be more stable and 'well-behaved'. The results shown in Fig. 10 appear to vindicate this decision as it shows lower uncertainty than the other two accuracy indicators. Mean($|r|$) converges to 2.2% after 100 samples, while MMRE and Pred25 converge to 3.5% and 4.3%, respectively. This means that a smaller number of samples would be necessary to achieve the same level of confidence when using mean($|r|$).

7 Discussion

The results of our analysis lead to two conclusions, one minor and one major.

The minor conclusion concerns the use of CBR for prediction. One of the design decisions that must be made is the choice of k, that is the number of analogies to be used. From the Desharnais data set, we see that $k = 2$ appears to be the optimum choice. We have high confidence in this result, not only because we formally tested the differences in accuracy for different values of k, but also because we repeated the sampling process 100 times and therefore have a high degree of confidence in the accuracy levels for each treatment. This is useful progress over earlier work, where we were less able to discern patterns with any confidence [17]. Interestingly, the much larger Finnish data set points to a similar pattern with $k = 2$ also being the preferred value. This optimum value for k also appears to be independent of the size of the training set used. This runs contrary to the intuitive argument given in the rationale for the investigation—that larger training sets would favour higher values of k. If this result is found to be generally applicable it will simplify the set-up of CBR-based prediction systems by reducing the number of variables that need to be tuned.

The major conclusion of this paper is, however, that it is dangerous to make inferences concerning the accuracy of prediction systems based on a small number of sampled training sets. This position was argued for a number of reasons. Firstly, when single samples are used, there is no way of assessing the confidence limits on any observations, i.e. it is impossible to assess whether the results are typical for the population they purport to represent (where the population is the set of all possible training sets that could have been derived from the underlying data set). Secondly, with small numbers of samples the confidence limits for prediction systems appear so large that they would be useless for showing one prediction system to be better than another, unless the difference in performance were very marked (which is often not the case). This is particularly true when tuning prediction systems, since we are typically comparing slight variations of one system with only small differences in performance. Finally, it has been demonstrated that apparent patterns in an investigation's results may be due to the particular training set rather than the phenomenon under study.

The examples using both the Desharnais and Finnish data sets show a significant degree of variation in the performance of prediction systems due to the random selection of training sets. It shows variations in central tendency and levels of uncertainty in prediction system accuracy for differing sizes of training sets. In this paper, the mean has been used as this measure of central tendency rather than the median, even though the distributions of the accuracy indictors such as MMRE and Mean($|r|$) are not symmetric. This is done because managers typically hold a portfolio of projects, and although the median may give a more representative indication of an individual result, the mean gives an unbiased view of likely cost across such a portfolio.

The case study also shows how large numbers of training sets can be used to produce results with clear confidence limits. How far these findings generalise is uncertain, although the two data sets we studied are quite distinct and the Finnish data set is relatively large with over 400 projects. This would seem to be an important topic for further investigation, since if we cannot reliably compare the accuracy of different prediction techniques then progress will be almost impossible.

From a pragmatic viewpoint, while encouraging the use of multiple training sets in validation studies, the authors acknowledge the level of additional work involved. The analysis of the worked example was only made possible through the use of automated tool support. Where such automation is not available other strategies might have to be adopted to reduce the number of samples needed. The large variation in results from different training sets may be due to training sets being chosen that are unrepresentative samples of the underlying population. Using stratified sampling to help ensure that each sample is more representative of the population might therefore reduce the variation. However, the data used in building prediction systems is often highly multidimensional, and there may be difficulties in stratifying such multidimensional data. This issue is simply noted here for future work.

Where possible we should strive to give confidence limits for accuracy results from empirical validations of prediction systems when a hold-out strategy is deployed. This is not relevant for model fitting or jack-knifing since the entire data set is utilised. However, these techniques suffer from the disadvantage that they tell us much less about the likely predictive performance of a given technique when used in a real-world context.

8 Acknowledgments

We wish to thank J.-M. Desharnais for making his data available and also STTF for providing the Finnish data set.

9 References

1 KOK, P., KITCHENHAM, B.A., and KIRAKOWSKI, J.: 'The MERMAID approach to software cost estimation'. presented at Esprit Technical Week, 1990

2 MACDONELL, S.G., SHEPPERD, M.J., and SALLIS, P.J.: 'Metrics for database systems: an empirical study', presented at 4th IEEE International Metrics Symposium, 1997, Alberqueque

3 WITTIG, G., and FINNIE, G.: 'Estimating software development effort with connectionists models', *Inf. Softw. Technol.*, 1997, **39**, pp. 469–476

4 SRINIVASAN, K., and FISHER, D.: 'Machine learning approaches to estimating development effort', *IEEE Trans. Softw. Eng.*, 1995, **21**, pp. 126–137

5 MAIR, C., KADODA, G., LEFLEY, M., PHALP, K., SCHOFIELD, C., SHEPPERD, M., and WEBSTER, S.: 'An investigation of machine learning based prediction systems', *J. Syst. Softw.*, 2000, **53**, pp. 23–29

6 EBERT, C.: 'Experiences with criticality predictions in software development', *ACM SIGSoft SEN*, 1997, **22**, pp. 278–293

7 DESHARNAIS, J.M.: 'Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction,' Masters Thesis, University of Montreal, 1989

8 SHEPPERD, M.J., and SCHOFIELD, C.: 'Estimating software project effort using analogies', *IEEE Trans. Softw. Eng.*, 1997, **23**, pp. 736–743

9 WALSTON, C.E., and FELIX, C.P.: 'A method of programming measurement and estimation', *IBM Syst. J.*, 1977, **16**, pp. 54–73

10 KITCHENHAM, B.A., and TAYLOR, N.R.: 'Software cost models', *ICL Tech. J.*, 1984, **4**, pp. 73–102

11 KEMERER, C.F.: 'An empirical validation of software cost estimation models', *Commun. ACM*, 1987, **30**, pp. 416–429

12 BOEHM, B.W.: 'Software engineering economics', *IEEE Trans. Softw. Eng.*, 1984, **10**, pp. 4–21

13 ALBRECHT, A.J., and GAFFNEY, J.R.: 'Software function, source lines of code, and development effort prediction: a software science validation', *IEEE Trans. Soft. Eng.*, 1983, **9**, pp. 639–648

14 MIYAZAKI, Y., and MORI, K.: 'COCOMO evaluation and tailoring'. presented at 8th IEEE International Software Engineering Conference, 28–30 August, 1985, London

15 EFRON, B., and GONG, G.: 'A leisurely look at the bootstrap, the jackknife and cross-validation', *Am. Stat.*, 1983, **37**, pp. 36–48

16 SHEPPERD, M.J., and KADODA, G.: 'Using simulation to evaluate prediction techniques', *IEEE Trans. Softw. Eng.*, 2001, **27**, pp. 987–998

17 KADODA, G., CARTWRIGHT, M., CHEN, L., and SHEPPERD, M.: 'Experiences using case-based reasoning to predict software project effort', presented at 4th International Conference on Empirical Assessment & Evaluation in Software Engineering, 17–19 April, 2000, Staffordshire, UK, Keele University

18 KITCHENHAM, B.A., MACDONELL, S.G., PICKARD, L., and SHEPPERD, M.J.: 'What accuracy statistics really measure', *IEE Proc. Softw. Eng.*, 2001, **148**, pp. 81–85

19 MAXWELL, K.: 'Applied statistics for software managers' (Prentice Hall, Upper Saddle River, NJ, 2002)