Making Inferences with Small Numbers of Training Sets

Colin Kirsopp, Martin Shepperd

Empirical Software Engineering Research Group
School of Design, Engineering & Computing
Bournemouth University
Royal LondonHouse
Bournemouth, BH1 3LT, UK

{ckirsopp, mshepper}@bmth.ac.uk

Abstract

This paper discusses a potential methodological problem with empirical studies assessing project effort prediction systems. 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. Typically this is done on very small numbers of sampled training sets. We show that such studies can lead to almost random results (particularly where relatively small effects are being studied). To illustrate this problem, we analyse two data sets, using a configuration problem for case-based prediction and generate results from 100 training sets. This enables us to produce results with quantified confidence limits. From this we conclude 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 poses something of a question over a number of empirical validations of prediction techniques and so we suggest that further research is needed as a matter of urgency.

Keywords: software effort prediction, empirical validation, hold-out strategy, case-based reasoning.

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 takes the form of comparing different prediction techniques on data that has been collected from real completed software projects. The goal is then to try and establish which technique, or techniques, are the most accurate. Over the last 10 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 [1, 2], artificial neural nets [3], case-based reasoning [4], rule induction [5] and fuzzy rule induction [6].

A strength of this 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 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 can be unacceptable and prevent meaningful comparisons between competing prediction techniques, particularly when the size of the effect is small.

The remainder of the paper is organised as follows. The next section provides a summary of research activity into project effort prediction systems with a particular emphasis upon empirical validation. Next we consider the problems associated with the common practice of splitting a data set into training and validation sets and illustrate this with a publicly available data set provided by Desharnais [7] and the ANGEL prediction system [8]. The results are confirmed by considering a much larger benchmarking data set known as the Finnish data set. We then discuss some of the options available for researchers wishing to make empirical comparisons of accuracy between different prediction systems.

2. A Short Review of Software Effort Estimation

As suggested by the introduction, there has been substantial interest — and therefore research — in the problem of predicting software costs, principally effort, at an early stage in a project. Early work included attempts to fit simple non-linear models to data collected, such as the work 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. Subsequently 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 adaptation or calibration. See, for example, the study conducted by Miyazaki and Mori [14] who demonstrated the positive 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 a very major repercussion upon how we evaluate it. Namely, we need to be careful not to use the same data for building and for evaluating.

Consequently most empirical research into prediction systems uses some kind of hold-out strategy. These 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 [15]. 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. 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 4GL 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 to what extent did our findings depend upon (i) the validation procedure including the proportion of cases in the training and validation sets and (ii) 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]

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

Table 1: Summary Statistics of Prediction Techniques (reproduced from [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.

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 α=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, however, 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. Empirical work

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. 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. This section provides an empirical exploration of the scale of the problem and of the utility of deploying multiple sampled training sets in validation studies. The main case study deals with different configurations 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 remainder of this section provides some additional corroborative work by considering a second much larger data set (unfortunately not in the public domain).

3.1Case Study on Configuring ANGEL

The purpose of this case study, is to investigate the distribution of accuracy levels associated with a particular training set which should be thought of as a sample taken from the population of all possible training sets. If the distribution of accuracy values is very large then it is likely that a single sample may wrongly estimate the centre and lead to problems of incorrect inference. Effectively, we may erroneously prefer Prediction System A to Prediction System B. Conversely the more samples taken, the better we can estimate the centre (and also the shape of the distribution). The results from the case study will be specific to the Desharnais data set, though we will then move on to consider the wider question of other data sets.

The research problem used is an investigation into the behaviour of case-based prediction systems and builds on earlier work [17]. Such 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, we must also know the feature that is to be predicted. Features can either be continuous (e.g. experience of the project manager), discrete (e.g. the 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 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, to generate a prediction for the target case. For all of the results described below the prediction is obtained by taking the mean of the target feature values from the *k* most similar cases. Where CBR is used without any adaptation, of the cases retrieved, this is referred to as a *k*-Nearest Neighbour (*k*-NN) technique.

In this case study we are interested in systematically exploring the relationship between the size of the training set (*n*) and the optimum of the number of cases (*k*) on which to base the prediction. In order to explore this relationship we use 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, 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 sub-set of features that will actually be used to build the prediction system. This procedure resulted in the removal of 5 features. This leaves us with a data set of 77 cases each with 5 features.

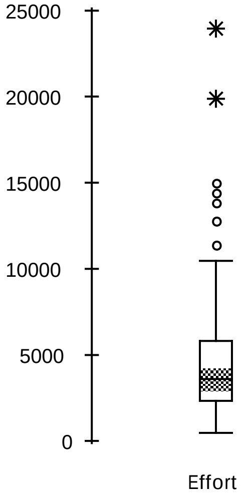

Figure 1: Boxplot of Project Effort for the Desharnais Data Set

One characteristic of the Desharnais data set is the presence of a small number of extreme outliers, denoted by stars in Figure 1. These are defined as exceeding (since the distribution is positively skewed) the upper hinge + 3.0(upper hinge - lower hinge). This is common characteristic of software engineering data sets and clearly leads to vulnerability in the sampling process to the creation of unrepresentative training sets.

To empirically assess the optimum value of *k* to use for CBR-based effort prediction, the accuracy of prediction systems build 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 error[‡]. 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. Figure 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.

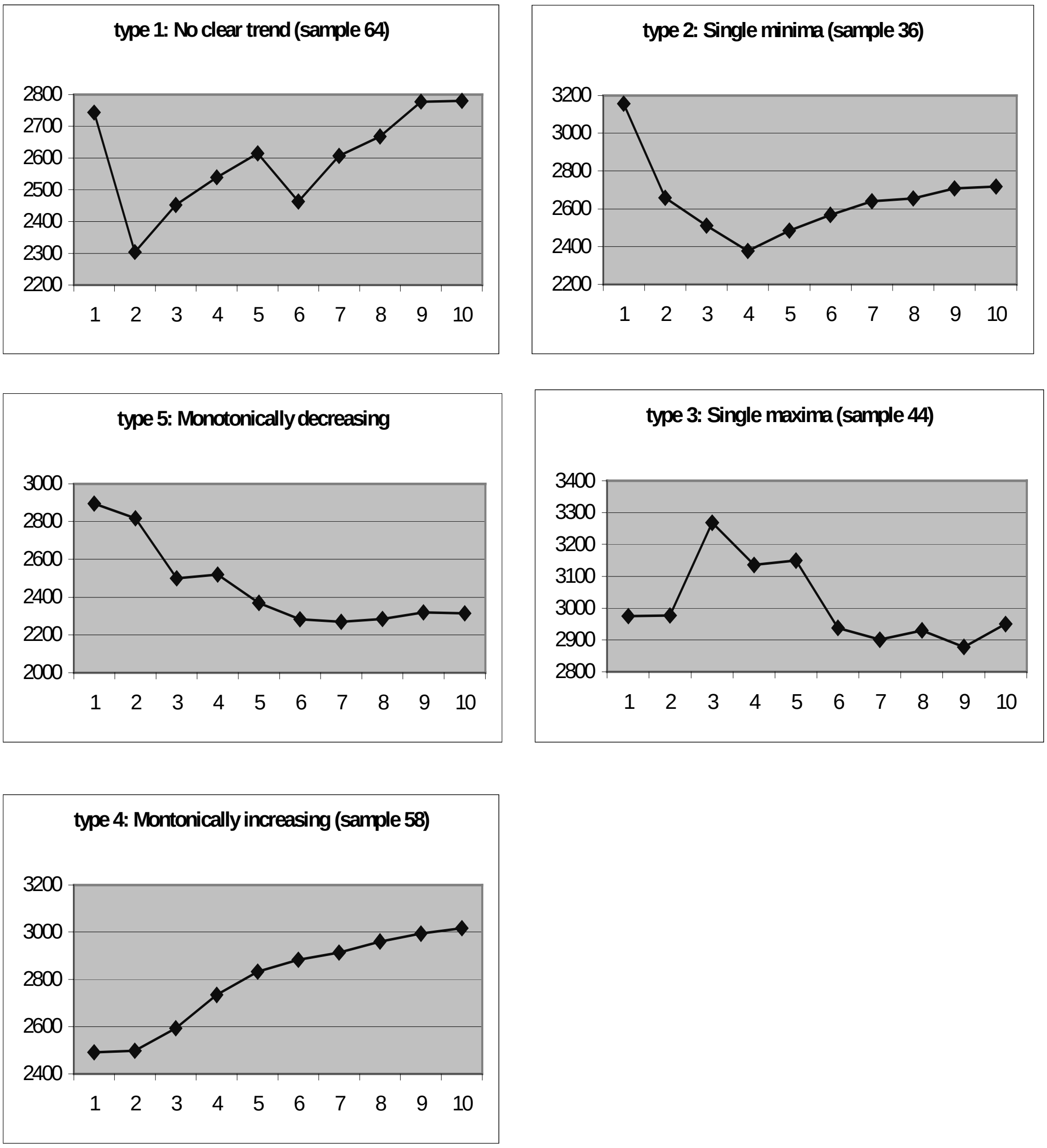

Figure 2: Example 'shape types' for mean(|r|) vs *k* of different samples

All 100 plots were then classified into the various categories shown in Figure 2. The categories are somewhat arbitrary, however, it is only intended to give an idea of the possible variation. The results from this classification are given in the pie chart of

[‡] We choose mean absolute residual as our accuracy indicator as we're indifferent 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 demerits of various accuracy indicators see Kitchenham *et al.* [18].

Figure 3. Only around one sixth of the samples showed no discernible trend. This means that there is a 5 in 6 chance of randomly sampling a training set that will show any one of a number of distinct functional forms.

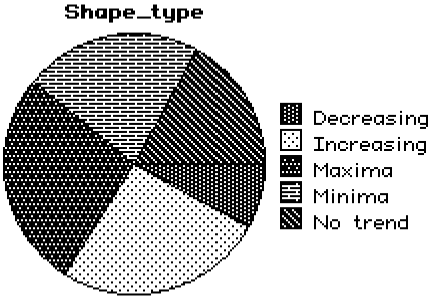

Figure 3: Distribution of 'shape-types' for mean(|r|) vs *k*

If there is such a range of results from a randomly selected training set how can any underlying trend be detected? Clearly, a single observation 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 this problem is to take large numbers of repeated measurements. This allows the calculation of both central tendencies and confidence limits on the properties being observed.

Figure 4 shows the relationship of the mean and the 95% confidence limits on the mean absolute residuals as the number of data points (training sets) is increased. In other words this indicates how confidently we can estimate where the true centre lies, given the number of training sets used to assess the accuracy of a prediction system (s). The data shown is for *n*=30, *k*=5 for illustrative purposes. This gives an indication of the number of sampled data sets that should be used to gain a particular level of confidence.

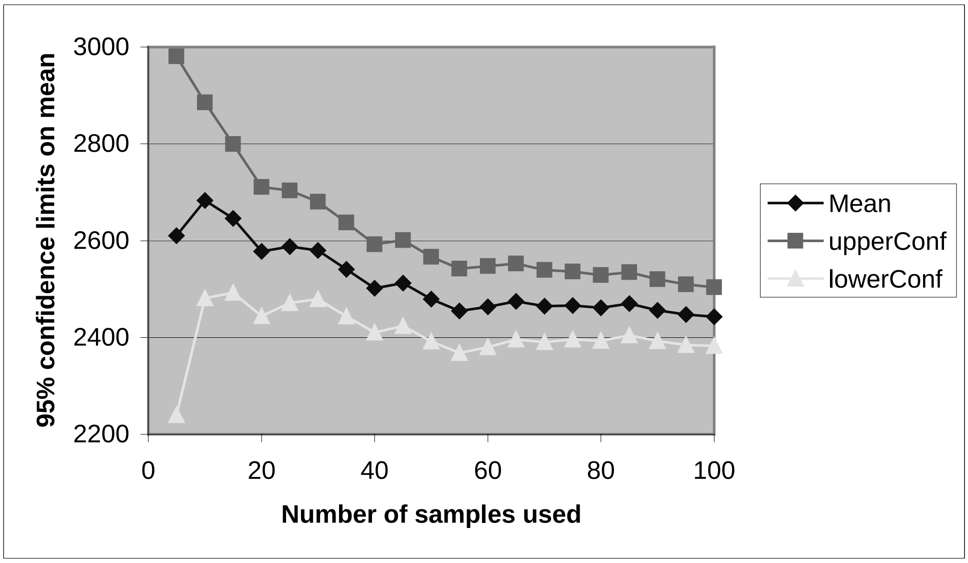

Figure 4: Variation of 95% confidence limits with number of samples

If we already have an estimate of the effect size we are looking for this data could be used to set a suitable number of samples to use. It is worth noting that if we compare the left-most range of uncertainty (5 samples) it is wider than the entire range of variation shown on most of the examples in Figure 2. In short, even if we had taken the average of 5 samples, we could only say that the data points lay somewhere in the entire height of the graph!

Figure 5 expresses the confidence limits as a fractional tolerance given as a percentage. For the given data using 5 samples we can be 95% confident that the actual value is within ±15% of the sample mean. With 20 samples this reduces to ±5%. It is probably only worth going beyond 40 samples if the effect size being observed is very small since usually there is a substantial effort associated with each validation.

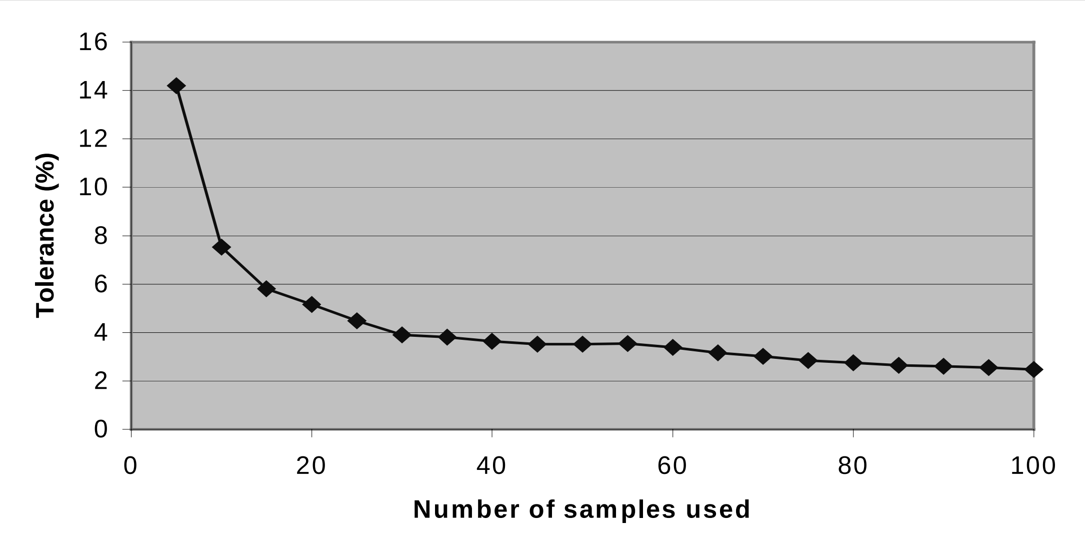

Figure 5: Tolerance on the mean absolute residuals vs number of samples used

An initial study into the relationship between *n* and *k* was first performed with a set of 20 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 paper uses a set of 100 randomly sampled training sets to build prediction systems for each combination of *k* (1-10) and *n* (10,20,30,40,50,60). This resulted in the building and assessment of 6000 prediction systems (100 x 10 x 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* as a series (Figure 6) or constant *k* as the series (Figure 7).

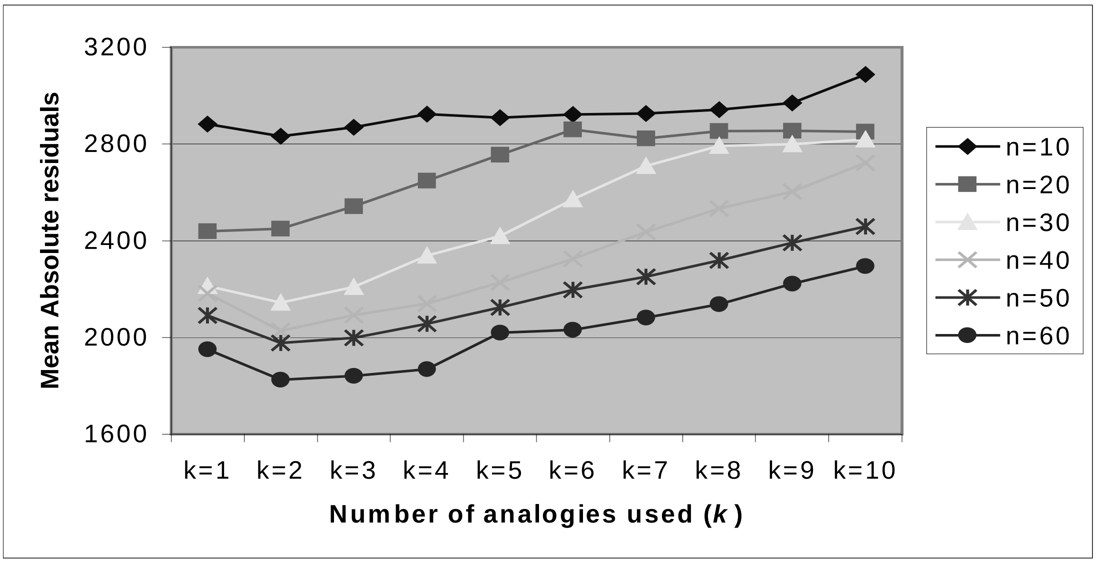

Figure 6: Variation of accuracy with *k* for different *n*-values

Figure 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 Figure 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?

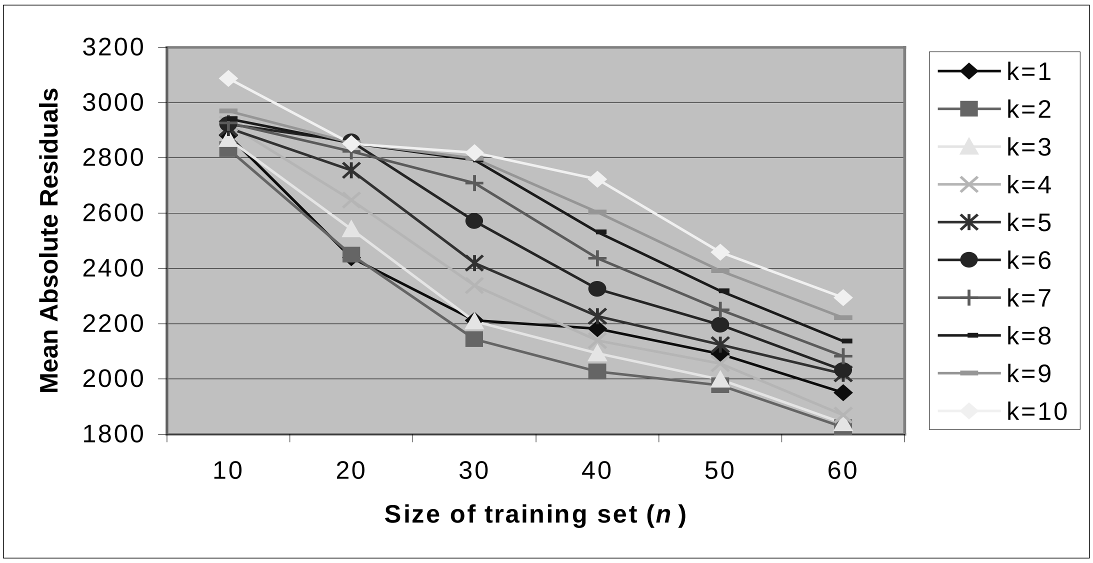

Figure 7: Variation of accuracy with n for different k-values

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-sided Wilcoxon Signed Rank test between the lowest *k*-value set and each of the other sets (an alpha level of 0.05 was used). 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 than 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.

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

Table 2 Variation of *k* with *n*

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 figure 9).

3.2 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 almost identical results to the Desharnais dataset. Individual sampled training sets can show a range of functional forms. Despite permitting larger training sets the confidence limits on prediction system results are still surprisingly high if less than 20 sampled training sets are used. This is shown in Figure 8 below.

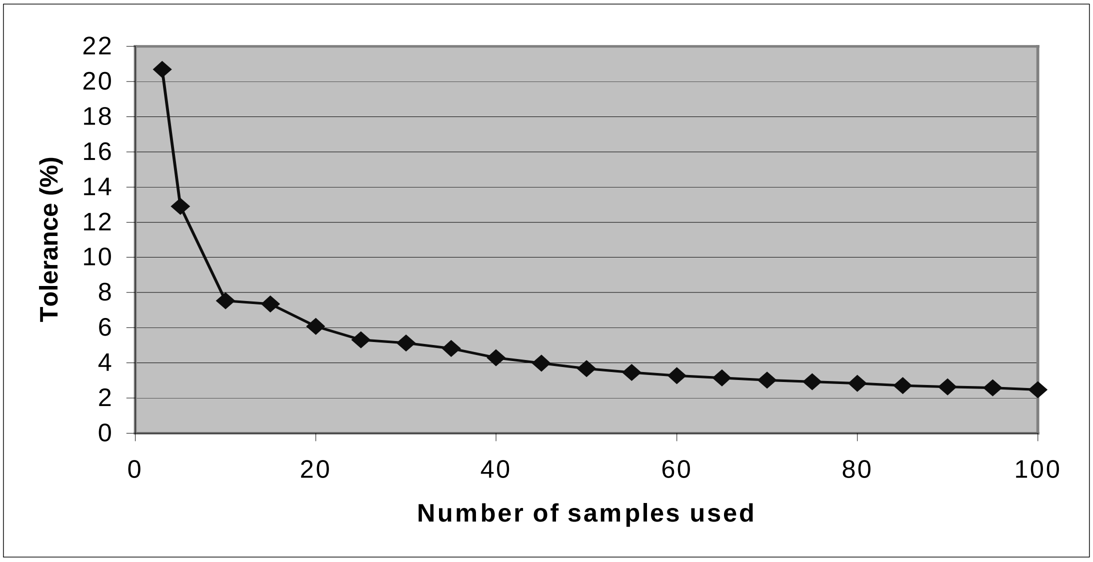

Figure 8: Variation of fractional confidence with n (Finnish dataset n=200, k=5)

The analysis of *n* versus *k* was also repeated for the Finnish dataset. It can be seen from figure 9 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.

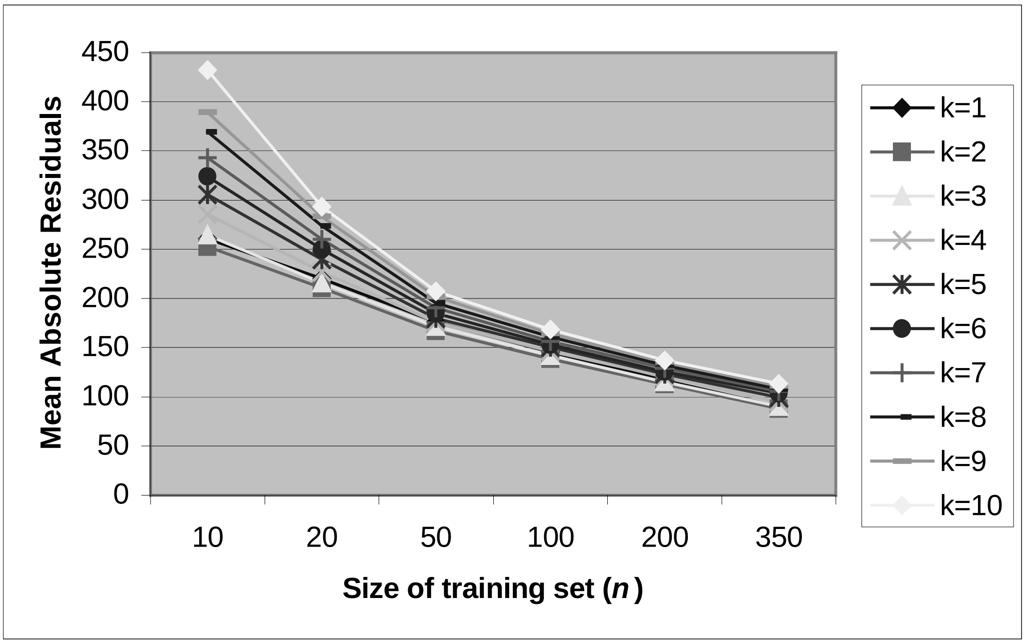

Figure 9: Line-plots of mean(|r|) vs n for different values of k (Finnish dataset)

The analysis of the Finnish dataset 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 Figure 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.

Figure 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 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 Figure appear to vindicate this decision as it converges faster (i.e. needs fewer samples) than the other two accuracy indicators.

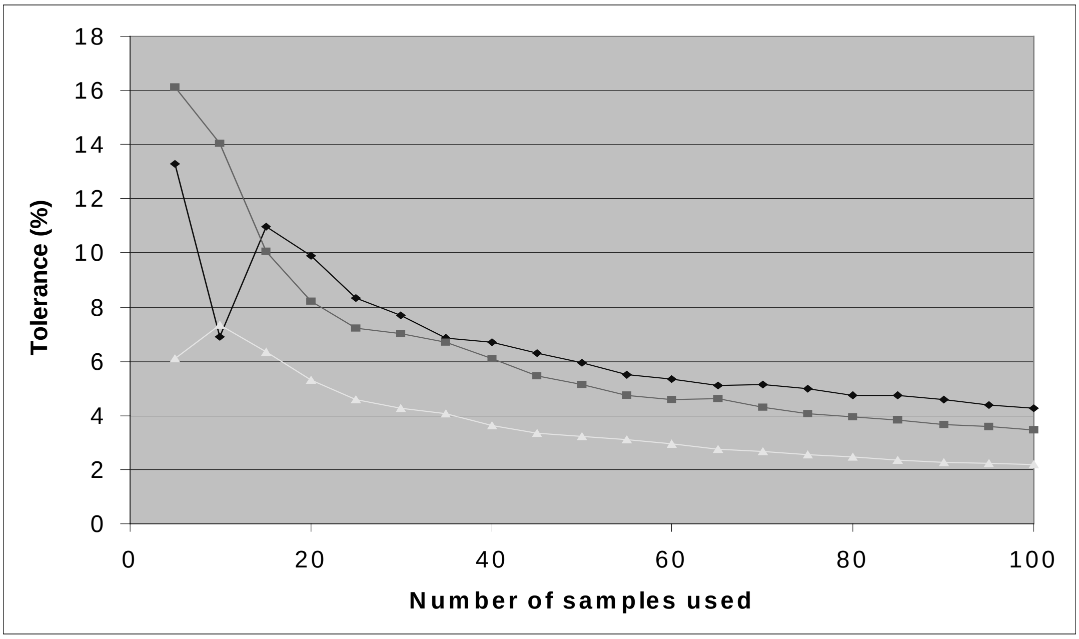

Figure 10: Fractional uncertainty for OLS regression with n=30

4. Discussion

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

The minor result 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.

The major claim 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 investigating small effects. This, unfortunately is typically what we try to do, especially when, say, tuning a prediction system. 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 the 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. It 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 in software effort prediction will be almost impossible.

From a pragmatic viewpoint, whilst 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.

Acknowledgements

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

References

[1] P. Kok, B. A. Kitchenham, and J. Kirakowski, "The MERMAID approach to software cost estimation," presented at Esprit Technical Week, 1990.

[2] S. G. MacDonell, M. J. Shepperd, and P. J. Sallis, "Metrics for Database Systems: An Empirical Study," presented at 4th IEEE Intl. Metrics Symp., Alberqueque, 1997.

[3] G. Wittig and G. Finnie, "Estimating software development effort with connectionists models," *Information & Software Technology,* vol. 39, pp. 469-476, 1997.

[4] K. Srinivasan and D. Fisher, "Machine learning approaches to estimating development effort," *IEEE Transactions on Software Engineering,* vol. 21, pp. 126-137, 1995.

[5] C. Mair, G. Kadoda, M. Lefley, K. Phalp, C. Schofield, M. Shepperd, and S. Webster, "An investigation of machine learning based prediction systems," *J. of Systems Software*, vol. 53, pp. pp23-29, 2000.

[6] C. Ebert, "Experiences with criticality predictions in software development," *ACM SIGSoft SEN*, vol. 22, pp. 278-293, 1997.

[7] J. M. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction," Masters Thesis, University of Montreal, 1989.

[8] M. J. Shepperd and C. Schofield, "Estimating software project effort using analogies," *IEEE Transactions on Software Engineering*, vol. 23, pp. 736-743, 1997.

[9] C. E. Walston and C. P. Felix, "A method of programming measurement and estimation," *IBM Systems Journal*, vol. 16, pp. 54-73, 1977.

[10] B. A. Kitchenham and N. R. Taylor, "Software cost models," *ICL Technical Journal*, vol. 4, pp. 73-102, 1984.

[11] C. F. Kemerer, "An empirical validation of software cost estimation models," *Communications of the ACM*, vol. 30, pp. 416-429, 1987.

[12] B. W. Boehm, "Software engineering economics," *IEEE Transactions on Software Engineering*, vol. 10, pp. 4-21, 1984.

[13] A. J. Albrecht and J. R. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation," *IEEE Transactions on Software Engineering*, vol. 9, pp. 639-648, 1983.

[14] Y. Miyazaki and K. Mori, "COCOMO Evaluation and Tailoring," presented at 8th IEEE Intl. Softw. Eng. Conf., London, 1985.

[15] B. Efron and G. Gong, "A leisurely look at the bootstrap, the jackknife and cross-validation," *The American Statistician*, vol. 37, pp. 36-48, 1983.

[16] M. J. Shepperd and G. Kadoda, "Using Simulation to Evaluate Prediction Techniques," *IEEE Trans. on Softw. Eng.*, vol. 27, pp. 987-998, 2001.

[17] G. Kadoda, M. Cartwright, L. Chen, and M. Shepperd, "Experiences using Case-Based Reasoning to predict software project effort," presented at 4th Intl. Conf. on Empirical Assessment & Evaluation in Software Engineering, Keele University, Staffordshire, UK, 2000.

[18] B. A. Kitchenham, S. G. MacDonell, L. Pickard, and M. J. Shepperd, "What accuracy statistics really measure," *IEE Proceedings - Software Engineering*, vol. 148, pp. 81-85, 2001.