Full citation: MacDonell, S.G., & Shepperd, M.J. (2003) Combining techniques to optimize effort predictions in software project management, Journal of Systems and Software 66(2), pp.91-98. http://dx.doi.org/10.1016/S0164-1212(02)00067-5

Combining techniques to optimize effort predictions in software project management

Stephen MacDonell

SERL, Auckland University of Technology
Private Bag 92006, Auckland 1142,
New Zealand
stephen.macdonell@aut.ac.nz

Martin J. Shepperd

Empirical Software Engineering Research Group,
School of Design, Engineering and Computing,
Bournemouth University, Bournemouth BH13LT, UK
mshepper@bmth.ac.uk

Abstract

This paper tackles two questions related to software effort prediction. First, is it valuable to combine prediction techniques? Second, if so, how? Many commentators have suggested the use of more than one technique in order to support effort prediction, but to date there has been little or no empirical investigation to support this recommendation. Our analysis of effort data from a medical records information system reveals that there is little, or even negative, covariance between the accuracy of our three chosen prediction techniques, namely, expert judgment, least squares regression and case-based reasoning. This indicates that when one technique predicts poorly, one or both of the others tends to perform significantly better. This is a particularly striking result given the relative homogeneity of our data set. Consequently, searching for the single ''best'' technique, at least in this case, leads to a suboptimal prediction strategy. The challenge then becomes one of identifying a means of determining a priori which prediction technique to use. Unfortunately, despite using a range of techniques including rule induction, we were unable to identify any simple mechanism for doing so. Nevertheless, we believe this remains an important research goal.

Keywords: Software effort prediction, Empirical analysis, Multiple techniques

1. INTRODUCTION

Among the more prominent and enduring challenges faced by software project managers is that of accurate and consistent prediction. Managers are called on to predict a variety of factors, including defect density, schedule slippage, effort requirements, project costs, and the like. In order to assist managers in this task extensive research has sought to build, evaluate and recommend prediction techniques, to the extent that a very large number of techniques now exist. The question that must be answered by a project manager, then, is ''Which technique, or techniques, should I use?''

There are a number of factors that can, and should, be considered in the selection of a prediction technique, and it is likely that trade-offs will need to be made in the process. Technique selection should be driven by both organizational need and organizational capability. In terms of need, the most common aim is to maximize accuracy in prediction; however, other issues may also need to be considered. For instance, perhaps a technique that produces slightly less accurate but generally more robust models would be preferred, particularly in cases where organisations do not have access to locally calibrated, well-behaved data sets. In terms of organizational capability, some modeling techniques are more complex than others, requiring significant expertise if they are to be used effectively. Whilst it is undoubtedly very positive that more sophisticated (and potentially more useful) techniques are being employed to build predictive models, this will only provide genuine benefit if the techniques are used appropriately.

For the moment, however, our main focus is on optimizing the accuracy of our predictions—in other words, we wish to produce estimates that are as close as possible to the actual values, irrespective of the other factors that may be important in the wider organizational setting. Given the availability of a range of techniques, recent research has attempted to determine which approach might be considered as the ''best'', this determination being based most commonly on one or more accuracy measures e.g. Briand et al. (2000) and Gray and MacDonell (1999). Since there are many factors that can vary from one study to another it is not surprising that the outcomes of these studies do not always correspond, and thus we find that the best technique varies from one study to another (Jeffery et al., 2001). In light of this inconsistency in outcomes, some authors have suggested that reliance on a single prediction is unnecessarily risky. To mitigate such a risk, researchers have recommended that for each prediction needed managers should use at least two approaches (Boehm, 1981; Kitchenham, 1996). One of the few studies to directly explore this area is (Kitchenham et al., 2002) based on prediction data for 145 projects from Computer Sciences Corporation. Here they had multiple estimates and used one of two strategies: either to take an average of the estimates or to select one that they decided they

would use (almost always an expert opinion estimate).

There are two ways in which multiple predictions might be harnessed. The first relates to the application of multiple techniques to different groups of observations in a data set. Predictive accuracy has been shown to be significantly affected by characteristics of the underlying data set (Pickard et al., 1999; Shepperd and Kadoda, 2001). Software engineering data is frequently heteroskedastic, and may contain a number of outlier observations. As a result we may see a prediction technique perform well on a subset of the data, but then perform very poorly on the remaining data points. This might be alleviated through the use of more than one technique across the data set. The second way in which multiple techniques could be utilized is in improving the accuracy of individual predictions. In order to predict a new value with greater confidence, more than one technique could be used to produce a range of estimated values. The project manager may then choose to adopt an upper and lower limit approach to the prediction or some (perhaps weighted) combination of the individual results could be used to produce a single adjusted value. Such an approach is based on the assumption that the larger the number of predictions made, the more likely that the predictions will converge to a reasonable value for the estimate.

In both cases reliance on a single prediction technique could therefore be a sub-optimal strategy, thus there may well be some potential in the application of multiple techniques. To date, however, there appears to have been no empirical work undertaken to assess the worth of such an approach. In the absence of any such evidence, most research concludes by recommending a single prediction technique as the most appropriate. An alternative approach might therefore be to use a combination of prediction techniques over a single data set. This may enable us to find an optimal set of approaches that together provide classifications and/or predictions that are more accurate, consistent and credible for a greater number of observations. Furthermore, it may also be possible to develop a set of heuristics that indicate, for a given environment or other characteristics, when an estimator should use a particular technique.

The remainder of the paper is organized as follows. The next section provides some background information on the three different prediction techniques that are used as the basis for our study. This is followed by a description of the case study itself. Then we describe our analysis and how we address the question of whether it is worth combining multiple techniques, followed by the question of how techniques could be combined. The paper concludes with discussion of the significance of our results and how the research could be progressed.

2. PREDICTION TECHNIQUES

Accurate and consistent prediction of resource requirements is a crucial component in the effective management of software projects. In spite of extensive research over the last 20 years the software community is still significantly challenged when it comes to effective resource prediction. On the whole, research efforts have focused on the development of techniques that are quantitatively based, in an effort to remove or reduce subjectivity in the prediction process. Examples of this work include the original parametric and regression-based models developed by Albrecht (1979), Putnam and Fitzsimmons (1979) and Boehm (1981). Of late, this work has been supplemented by the application of other data-driven techniques such as neural networks, in a further attempt to produce more accurate predictions of resource requirements (Bode, 1998; Gray and MacDonell, 1999; Wittig and Finnie, 1997).

The use of data-driven techniques does, however, have significant limitations. For instance it can lead to the development of models that may well be accurate for a given sample but that fail to generalize when conditions change. In an industry characterized by change, in technologies, processes, people and so on, this can prove to be a major constraint. Furthermore, whilst a model derived via a neural network may produce reasonably accurate predictions, it may be unacceptable to project managers because it is not sufficiently transparent to enable understanding of the predictive model in terms of how a particular prediction value is reached. As a result, in recent years we have observed the resurgence of modeling techniques that emphasize transparency and explanation. For example, expert judgment has become more acceptable as a genuine (albeit informal) modeling technique. Approaches that attempt to incorporate aspects of the philosophy underlying expert judgment have also gained prominence in recent studies—this includes techniques based on analogy and case-based reasoning, and techniques that explicitly build uncertainty into the prediction process (Mukhopadhyay et al., 1992; Shepperd and Schofield, 1997).

In this study we have chosen to use a representative selection of three modeling techniques to investigate whether techniques are indeed complementary—where one performs poorly, does another perform well? The three techniques chosen are expert judgment, least-squares linear regression (LSR) and case-based reasoning (CBR) via the ANGEL software tool (Shepperd and Schofield, 1997).

2.1 Expert judgment

As the name implies, expert judgment is the informal process whereby one or more informed individuals provide their own experience-based predictions. In spite of the existence of many other more formal or semiformal alternatives there remains strong evidence of the continued use of expert judgment in project management (Heemstra, 1992; Host and Wohlin, 1998; Hughes, 1996) and there is little suggestion that it will be entirely superseded by other approaches. However, there is also evidence of bias in such expert estimation (DeMarco, 1982; Gray et al., 1999). Accepting these two conditions leads us to suggest that we may be more successful in prediction and project management if, rather than attempting to replace expert judgment, we instead work with it by considering where expert judgment performs

less well and building alternative models in order to supplement personal expertise.

2.2. Least-squares linear regression

Linear regression attempts to find a straight-line relationship between one or more predictor parameters and a dependent variable, minimizing the square of the errors across the range of observations in the data set. Some researchers have advocated building simple local models, e.g. Kok et al. (1990), using this type of approach. The philosophy is essentially one of solving local prediction problems before attempting to construct universal models. The resulting prediction systems take the form:

$$\hat{Y} = \beta_0 + \beta_1 X_1, \ldots, \beta_n X_n \qquad (1)$$

where $\hat{Y}$ is the estimated value and $X_1,\ldots, X_n$ are independent variables, for example number of files or interfaces, that the estimator has found to significantly contribute to the prediction of effort. A disadvantage with LSR is its vulnerability to extreme outlier values although robust regression techniques, that are less sensitive to such problems, have been used successfully, e.g. Briand et al. (2000). Another potential problem is the impact of collinearity—the tendency of independent variables to be strongly correlated with one another— upon the stability of a regression type prediction system.

2.3. Case-based reasoning

CBR, otherwise known as estimation by analogy, attempts to predict by finding similar cases to the target project. Generally, similarity is measured as Euclidean distance in *p*-dimensional feature space, where each case is characterized by *p* features such as the number of interfaces or type of programming language. Having found similar projects with known effort values, these can then be utilised to predict effort for the target project. A number of researchers have used this type of approach with generally quite encouraging results, for example, Finnie et al. (1997), Prietula et al. (1996) and Shepperd et al. (1996). For a fuller discussion of different distance measures and adaptation strategies, see Kolodner (1993). CBR contrasts substantially with LSR in that it is more robust to problems of distribution and seeks to cluster observations rather than interpolate or extrapolate.

3. THE CASE STUDY

The empirical analysis undertaken in this study centres on a set of measures taken from modules in a single medical records database system, built and implemented over a period of five months[1]. There were 77 observations in the data set, each representing a module built to implement data entry/edit or reporting functionality. For each module we had 26 independent variables describing the data model (for example, the number of entities and the number and type of different entity relationships), the number and types of transaction to be processed and the number and types of different screens that were required. All this data was available from the module specification and so could legitimately be used as input to a prediction system. For each module we also had the project manager's estimate of effort and the actual effort in person hours.

The aim of the case study was to compare different prediction techniques and consider how they might usefully be combined. The three techniques were chosen on the grounds of contrasting approach and frequency of use either by practitioners or within the research community. In order to consider the predictive capability of each technique we divided the dataset in the ratio of 2:1 into a training set and validation set. Essentially this is the situation where one can imagine that 51 modules have been completed and 26 are outstanding. Since all the modules comprise a single system there was no meaningful ordering information so the division of the dataset was done randomly. From other work (Shepperd and Kadoda, 2001) we have found that one-off sampling can lead to misleading results so we repeated the sampling process in order to increase our confidence in the results. The training sets were labelled TS1 and TS2.

Since we were more concerned to explore combining techniques than identifying the best technique we decided to use a straightforward approach to both LSR and CBR. For the LSR we considered the cross correlations between the collected variables. From this it was evident that there was considerable inter-item correlation, with the underlying dimension corresponding to data model size.

We also observed the strongest relationship was between the number of attributes in the data model (ATTRIBS) and effort (ACTHRS) as revealed by Fig. 1. These patterns were quite stable between the two training sets. A stepwise procedure offered little additional explanatory power and may have led to potential problems due to collinearity, consequently we used two very simple, and similar, regression equations.

$$\text{Effort} = 1.147\ (\text{ATTRIBS}) \qquad (2)$$

$$\text{Effort} = 1.160\ (\text{ATTRIBS}) \qquad (3)$$

Note that Eq. (2) was derived from TS1 and Eq. (3) from TS2. Both intercepts were small, positive and not statistically significant and were therefore dropped from the equations.

When using CBR there are a number of decisions that must be made, e.g. the number of analogies to use, which distance measure to employ and whether to use a subset of all available features. Unfortunately, the answers to such questions are largely heuristically based since no general theory exists. As with LSR we adopted a fairly simplistic—and from a practitioner perspective realistic—approach. We decided to use three analogies as there is some evidence that increasing the number of analogies with the size of the case base is an effective strategy (Kadoda et al., 2000). We also decided to use the entire feature set even though there is evidence to suggest this is less than optimal. The reason for this decision is that there are no known efficient algorithms to solve this problem and our dataset comprised 26 features, so a brute force

[1] Readers wishing to obtain a copy of the data set should please contact either of the authors.

approach, as implemented in ANGEL, would take an inordinate amount of time.

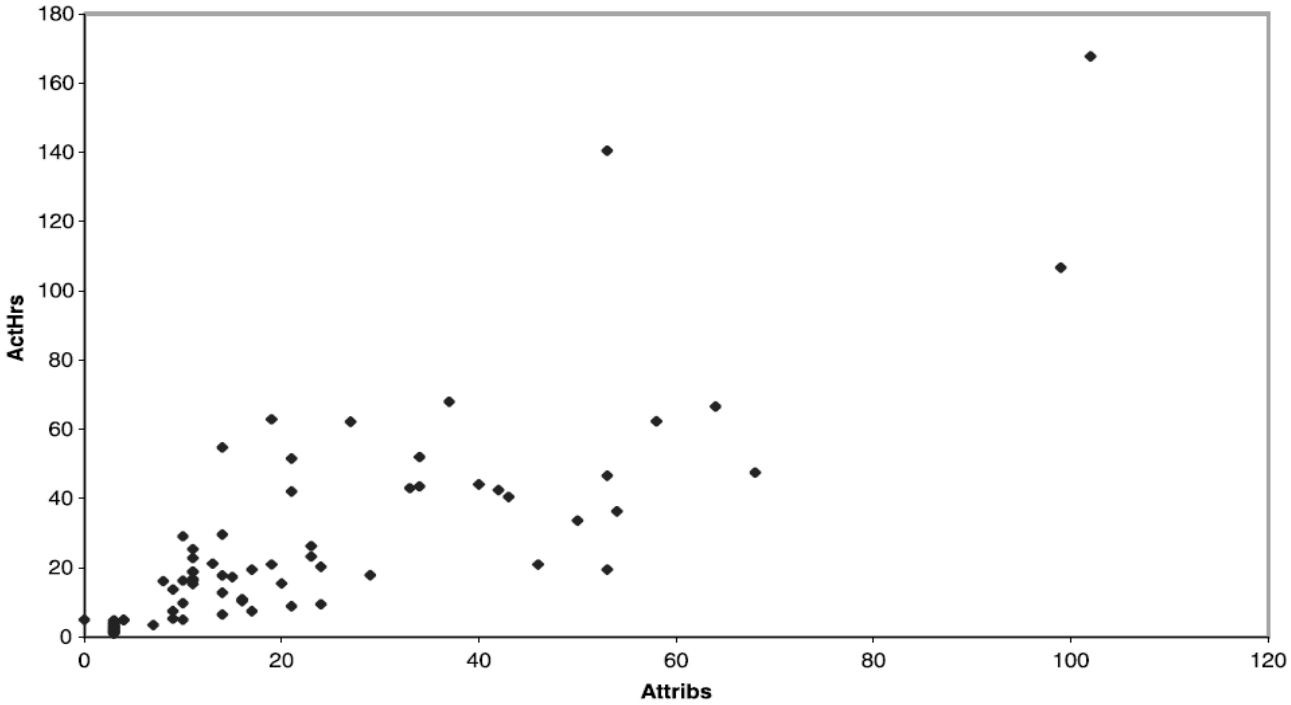

Fig. 1. Scatter plot of attributes against effort.

4. RESULTS

First we consider the errors that arise from each of the three prediction techniques (expert judgment, LSR and CBR). Recall that we repeated the sampling procedure so that we have two training sets (TS1 and TS2) with 51 cases randomly sampled from the data set leaving validation sets of 26 modules in each case.

Technique	Sum of absolute residuals	Median absolute residual	Range	MMRE (%)	Bias (%)
Panel A: TS1					
Expert judgment	295.90	5.65	103.00	74.9	−26
LSR	241.26	4.92	79.27	45.5	−9
CBR	253.66	3.89	89.10	49.2	−6
Panel B: TS2					
Expert judgment	410.10	6.35	94.90	87.5	−20
LSR	223.32	5.55	40.50	43.2	−11
CBR	359.57	8.12	54.37	58.8	−2

Table 1. Absolute error by technique for TS1 (panel A) and TS2 (panel B)

Prediction technique	TS1	TS2	Combined
Expert	6	6	12
LSR	9	12	21
CBR	11	8	19

Table 2. Frequencies of technique performing best

Table 1 (panels A and B) shows the relative accuracies of the three techniques using absolute[2] residuals in person hours—our preferred indicator since it is unbiased—and MMRE since it is easier to compare across different validation sets. In this case we observe no conflict[3] between the rankings for either training set, with LSR to be preferred to CBR which in turn is to be preferred to the expert. The righthandmost column denotes the tendency for bias. All three techniques tend to under-estimate, CBR exhibiting the fewest problems. Expert judgment performs particularly poorly in this regard. Thus a fairly straightforward analysis would point to the use of LSR as the preferred prediction technique.

Table 2 shows the frequencies of which technique is best, where best is defined as the technique with the minimum absolute residual for a particular module. This again suggests—in line with the sums of absolute residuals (Table 1 (panels A and B))—that LSR is the most effective technique and expert judgment the least accurate. Note, however, that pursuing a single-technique strategy (in this case using only LSR) would result in using a sub-optimal technique more than 50% of the time (31 out of 52 predictions). In fact careful examination of the prediction errors from the three techniques shows quite a marked tendency for the techniques to behave independently.

Fig. 2a–c provides an illustration of the different behaviors of each prediction technique. Each diagram

[2] We use absolute values since for the purposes of this analysis we are indifferent to under and over-estimates.

[3] Elsewhere we discuss some of the problems of different accuracy indicators and how to address potential conflicts when different indicators suggest different rankings for the same data set (Shepperd et al., 2000).

shows the distribution of errors, as absolute percentages for ease of viewing. Here the bold case indicates the same module for all three histograms and shows how the worst error from expert judgment has a corresponding less bad prediction from LSR and an even better prediction using CBR. Note also the tendency of each technique towards a bimodal distribution indicating some dichotomisation between predicting adequately and quite poorly.

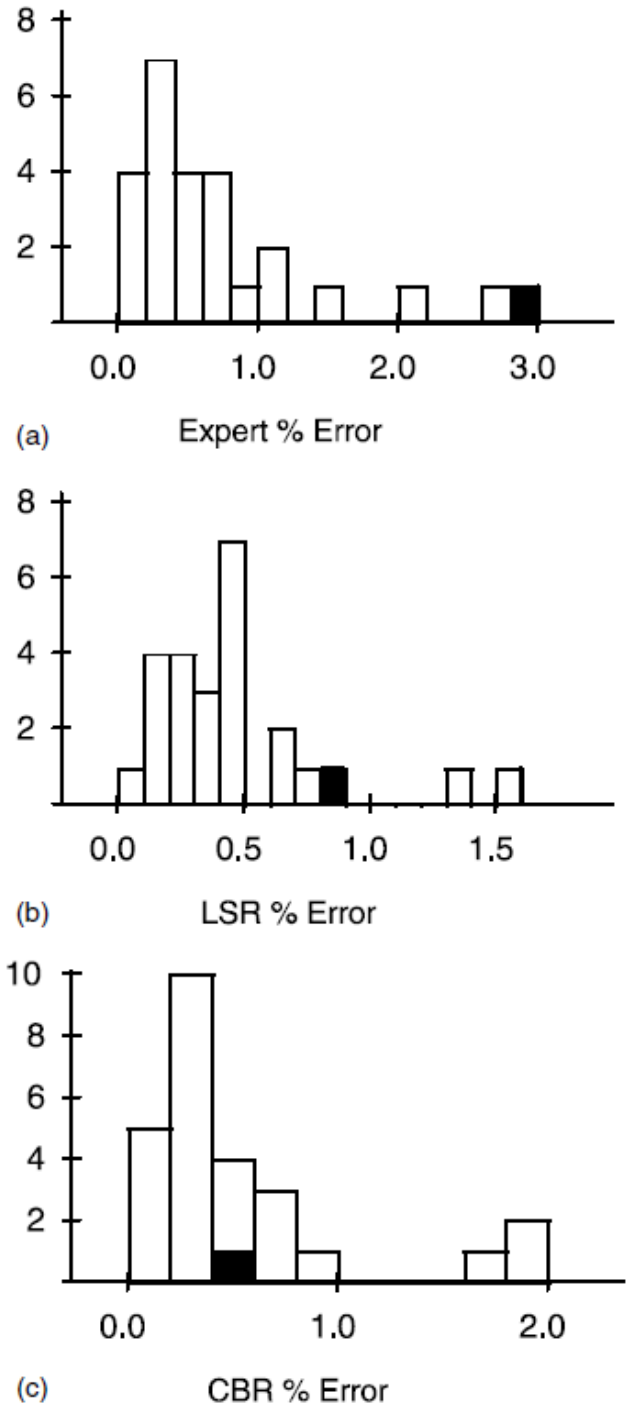

Fig. 2. Histogram of % prediction errors for (a) the expert, (b) LSR and (c) CBR.

Table 3 shows the covariance between the absolute errors where the covariance is a measure of association, however, unlike correlation covariance is not bounded by -1and +1. Pairs of values are shown for training sets TS1 and TS2; however, there is little difference in behavior between the two sets of predictions arising from the two different training sets. The generally low values denote little relationship between errors from different techniques for the *same* module. This is an interesting and important finding since it indicates that just because one technique predicts badly it does not necessarily follow that the other techniques will be equally poor. This suggests that there is potential for improving upon the strategy of merely selecting the best technique.

Put differently, if we could vary techniques and had a means of knowing a priori which would perform best, what scope is there for improving our prediction process? So our second research question becomes can we do better than just LSR? Unfortunately we were unable to find any simple statistical patterns to try and predict situations where LSR performs poorly and one or both other techniques do well.

We therefore decided to explore the use of rule induction to try to learn a decision tree as to which technique to use in which circumstances. Our approach was to train the rule induction (RI) algorithm C5.0 on each training set to see if we could generate trees that were able to predict which technique would have the smallest error for the validation sets. To prevent over adaptation—a potential problem given the relatively small size of the training sets—we allocated the data to one of five bins.

Prediction technique	Expert		LSR		CBR	
Expert	0.612	0.745				
LSR	0.100	−0.108	0.125	0.368		
CBR	0.028	0.009	0.012	0.108	0.264	0.423

Table 3. Covariance between absolute errors from prediction techniques

Technique	Sum	Median	Range
Panel A: TS1			
LSR	241.26	4.92	79.27
Average of techniques	214.45	3.25	90.54
Using RI	224.66	5.15	79.71
Theoretical optimum	169.63	1.74	79.71
Panel B: TS2			
LSR	223.32	5.55	40.50
Average of techniques	226.35	5.59	45.18
Using RI	247.58	5.46	51.24
Theoretical optimum	138.59	2.50	40.10

Table 4. Absolute error by technique for TS1 (panel A) and TS2 (panel B)

Table 4 (panels A and B) summarize the results for each training set, showing absolute errors from LSR (our baseline single technique), using a simple unweighted average of all three techniques, applying the rule tree to select a technique and the theoretical optimum if we could always correctly pick the best technique. For both training sets we see that the theoretical optimum—assuming perfect knowledge—is significantly better than the single best technique, LSR. Unfortunately, the picture is more confused between the other two approaches, and there is no strong support for using RI to determine which technique should be used instead of LSR or even the

average of LSR, CBR and expert judgment. We can only conclude that the factors that make CBR relatively more effective are to do with the absence of a good relationship between the data in the data set and effort leading to poor LSR performance. In other words, CBR might be better viewed as a default technique when LSR is unable to do well. Our difficulty is that searching for the absence of something is rather more problematic than its existence, which we believe partly explains our difficulties in finding objective rules for determining which prediction technique to utilize. So the problem is the data set—which is somewhat orientated towards characterizing the data model—and the lesson is that a richer set of attributes is likely to have been more helpful.

5. DISCUSSION

In this paper we have considered two questions. Firstly, is there any potential benefit in following the advice of a number of prediction experts indicating that we should use more than one technique? Secondly, if there is potential benefit, how can this be exploited at the time of prediction as opposed to after the event?

Our analysis suggests even for the relatively homogeneous data set under study, where all modules are subparts of the same project, the three different prediction techniques showed a marked independence. In other words, just because one technique fares badly for a given prediction this does not imply that the others will be equally ineffective. This view was confirmed by examining the covariance between the errors from each technique. We found very little or even negative covariance. We also found that although LSR was overall the most accurate technique, this was not a very strong result. Simply using the best technique, that is LSR, would result in using a sub-optimal technique on 31 of 52 occasions or the majority of the time. So therefore we can conclude that there is indeed potential benefit in using more than one technique within our data set.

Factors that will influence how generally this finding might apply include the diversity of the different prediction techniques. We intentionally selected contrasting approaches. It is less obvious, however, that using a number of closely related flavors of regression analysis will lead to such wide variation in the prediction accuracy of the various techniques. Without this relative variation there is of course little merit in using multiple techniques. Another factor is whether one technique dominates the others. In our data set this did not appear to be the case. Other studies, for example Mair et al. (2000) using the Desharnais (1989) data set, have likewise failed to find a dominant technique, so we conclude that this is not that an unusual circumstance.

Therefore, our advice to estimators is initially to establish whether a dominant prediction technique can be found. If so, there is little purpose in trying to combine more than one technique. If no dominant technique can be found, our recommendation is to employ as diverse a set of techniques as possible. Clearly LSR differs substantially from CBR since LSR uses interpolation and extrapolation, whilst CBR works more by clustering. Expert judgment differs in that it has access to a wider source of data than that necessarily captured in the data set. Another contrasting possibility, and not one that we have explored in this paper, would be to use fuzzy rules (Gray and MacDonell, 1997).

Turning to our second question, it is all very well suggesting that multiple techniques can give more accurate results in theory but one is still left with the question, how. Unfortunately, our statistical analysis was unable to reveal any very clear patterns as to which conditions favored which technique, e.g. for large functions use expert judgment. Our next approach was to use a rule induction algorithm to learn rules from the training sets. The rules took the form of decision trees where the leaf nodes were the different prediction techniques. We then evaluated the rules on the two validation sets. Again our results were disappointing and although it could be argued that it is hard to show significance when there are only 26 cases in the validation set, there was no real evidence that this approach offered any real benefit. Also, the trees differed considerably between the two training sets, which did not increase our sense of confidence in their usefulness. Part of the problem would seem to be that there are many external factors that influence the effort to implement a function, so our search for an automated procedure to determine which technique to use is somewhat inhibited. Ideally the next step would have been to have sought input from the project manager and developers, particularly regarding those cases where either CBR or expert judgment had performed significantly better than the best technique, namely LSR. Unfortunately this was not possible since the data had been collected in 1996 and we no longer had access to the relevant staff. However, we note that this is consistent with the CSC data (Kitchenham et al., 2002) where they found that there was little difference in accuracy between estimates based on expert opinion and estimates based on averages (indeed, if anything the simple expert opinion estimates were marginally, but not significantly better). Another interesting implication of the CSC work is that in many cases the estimators were reasonably sure of which was the best estimate, that is, they correctly chose the most accurate one. Consequently, even if an automated RI technique does not produce useful results, expert opinion might.

So, in conclusion, we have shown that the advice to use more than one effort prediction technique has some basis, and for our data set there were substantial *potential* benefits from using the three techniques of expert judgment, LSR and CBR. We were unable, however, to find a reliable means of knowing a priori which technique to use, so this must remain an open research question. Nevertheless, it is our view that this is an important question since it would be a means of unlocking considerable improvements in terms of prediction accuracy.

ACKNOWLEDGEMENTS

The work described here was carried out while S.G. MacDonell was working at the University of Otago. M. Shepperd was supported as the J.A. Valentine Visiting

Professor at the University of Otago whilst undertaking this research work. The authors would also like to thank Barbara Kitchenham for her valuable comments on an earlier draft of the paper.

REFERENCES

Albrecht, A.J., 1979. Measuring application development productivity. In: SHARE-GUIDE Symposium. IBM, Monterey, CA.

Bode, J., 1998. Neural networks for cost estimation. Cost Engineering 40 (1), 25–30.

Boehm, B.W., 1981. Software Engineering Economics. Prentice-Hall, Englewood Cliffs, NJ.

Briand, L., Langley, T., Wieczorek, I., 2000. Using the European space agency data set: a replicated assessment and comparison of common software cost modeling techniques. In: 22nd IEEE International Conference on Software Engineering. Computer Society Press, Limerick, Ireland.

DeMarco, T., 1982. Controlling Software Projects. Management, Measurement and Estimation. Yourdon Press, New York.

Desharnais, J.M., 1989. Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction, University of Montreal.

Finnie, G.R., Wittig, G.E., Desharnais, J.-M., 1997. A comparison of software effort estimation techniques using function points with neural networks, case based reasoning and regression models. Journal of Systems and Software 39, 281–289.

Gray, A.R., MacDonell, S.G., 1999. Software metrics data analysis— exploring the relative performance of some commonly used modeling techniques. Empirical Software Engineering 4, 297–316.

Gray, A.R., MacDonell, S.G., 1997. Applications of fuzzy logic to software metric models for development effort estimation. In: Annual Meeting of the North American Fuzzy Information Processing Society, NAFIPS, Syracuse, NY.

Gray, A.R., MacDonell, S.G., Shepperd, M.J., 1999. Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgement. In: IEEE 6th International Metrics Symposium 1999. IEEE Computer Society, Boca Raton, FL.

Heemstra, F.J., 1992. Software cost estimation. Information & Software Technology 34 (10), 627–639.

Host, M., Wohlin, C., 1998. An experimental study of individual subjective effort estimations and combinations of the estimates. In: 20th IEEE International Conference on Software Engineering. Computer Society Press, Kyoto, Japan.

Hughes, R.T., 1996. Expert judgement as an estimating method. Information & Software Technology 38 (2), 67–75.

Jeffery, R., Ruhe, M., Wieczorek, I., 2001. Using public domain metrics to estimate software development effort. In: 7th IEEE International Metrics Symposium. Computer Society Press, London.

Kadoda, G., Cartwright, M., Chen, L., Shepperd, M.J., 2000. Experiences using case-based reasoning to predict software project effort. In: 4th International Conference on Empirical Assessment & Evaluation in Software Engineering, Keele University, Staffordshire, UK.

Kitchenham, B.A., 1996. Measurement for Software Process Improvement. Blackwell, Oxford.

Kitchenham, B.A. et al., 2002. A case study of maintenance estimation accuracy. Journal of Systems and Software 64 (1), 57–77.

Kok, P., Kitchenham, B.A., Kirakowski, J., 1990. The MERMAID approach to software cost estimation. Esprit Technical Week.

Kolodner, J.L., 1993. Case-Based Reasoning. Morgan-Kaufmann, Los Altas, CA.

Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M.J., Webster, S., 2000. An investigation of machine learning based prediction systems. Journal of Systems Software 53 (1), 23 – 29.

Mukhopadhyay, T., Vicinanza, S.S., Prietula, M.J., 1992. Examining the feasibility of a case-based reasoning model for software effort estimation. MIS Quarterly 16, 155–171.

Pickard, L., Kitchenham, B., Linkman, S., 1999. An investigation of analysis techniques for software datasets. In: 6th IEEE International Software Metrics Symposium. IEEE Computer Society, Boca Raton, FL.

Prietula, M.J., Vicinanza, S.S., Mukhopadhyay, T., 1996. Software effort estimation with a case-based reasoner. Journal of Experimental & Theoretical Artificial Intelligence 8, 341–363.

Putnam, L.H., Fitzsimmons, A., 1979. Estimating software costs. Datamation, 189–198.

Shepperd, M.J., Kadoda, G., 2001. Using simulation to evaluate prediction techniques. In: 7th IEEE International Metrics Symposium. IEEE Computer Society, London.

Shepperd, M.J., Schofield, C., 1997. Estimating software project effort using analogies. IEEE Transactions on Software Engineering 23 (11), 736–743.

Shepperd, M.J., Cartwright, M.H., Kadoda, G.F., 2000. On building prediction systems for software engineers. Empirical Software Engineering 5 (4), 175–182.

Shepperd, M.J., Schofield, C., Kitchenham, B.A., 1996. Effort estimation using analogy. In: 18th International Conference on Software Engineering, IEEE Computer Society Press, Berlin.

Wittig, G., Finnie, G., 1997. Estimating software development effort with connectionist models. Information & Software Technology 39 (7), 469–476.

ABOUT AUTHORS

Stephen G. MacDonell is Professor of Software Engineering and Head of the School of Information Technology at Auckland University of Technology in New Zealand. He holds BCom(Hons) and MCom degrees in Information Science from the University of Otago and a PhD in Software Engineering from the University of Cambridge. His main research activities are in the areas of software measurement, project planning, estimation and management, software engineering data analysis, and software forensics. He is a member of the ACM and NZCS.

Martin J. Shepperd received a PhD in computer science from the Open University, UK in 1991. He is professor of software engineering at Bournemouth University, UK. He has published more than 70 refereed papers and three books in the area of empirical software engineering. He is the Editor of the Journal Information & Software Technology and Associate Editor of IEEE Transactions on Software Engineering.