An Empirical Analysis of Linear Adaptation Techniques for Case-Based Prediction

Colin Kirsopp[1], Emilia Mendes[2], Rahul Premraj[1], and Martin Shepperd[1]

[1] Bournemouth University, U.K.
[2] University of Auckland, New Zealand

Abstract. In this paper we explore the role of case adaptation for feature vector prediction problems. We focus on software project effort. We study three data sets that range from small (less than 20 cases) through medium (approximately 80 cases) to large (approximately 400 cases). These are typical sizes for this problem domain. We compare two variants of a linear size adjustment technique and (as a baseline) a simple k-NN approach. Our results show that the linear scaling techniques studied result in statistically significant improvements to predictions. However, the size of these improvements is relatively small, typically about 10%. The results include a number of extreme outliers which might be problematic if the techniques are to be used in practice. This suggests further work is required to cope better with the outlier problem.

1 Introduction

Over the past 15 years case-based reasoning (CBR) has been successfully applied to a wide range of problem domains. Our particular interest is in predicting effort (and related factors such as duration) for software projects. Of course to be useful, such predictions are required at an early stage. This is important because software projects are difficult to justify or manage if it isn't possible to estimate how long they will last and how much effort they will consume. For this reason cost modelling has been an active research topic for more than 30 years. Despite this activity, no one technique has been found to be consistently effective. It has proved to be challenging for a number of reasons. Typically, data sets are small, as projects occur relatively infrequently, perhaps just a few per year. Data is heterogeneous so merging data from different environments is seldom fruitful. Data collection environments are characterised by change, noise and uncertainty. Moreover, software engineering is a predominantly creative activity, consequently we do not have a strong underlying theory.

Various research groups, including ours, started to explore the application of CBR methods to predicting project effort, motivated in part by the obvious similarities between project managers seeking to estimate based on recall of past similar projects, and the formal use of analogies in CBR [17, 19, 5]. Encouraging results were reported, for example, in an analysis of 9 different data sets and using stepwise regression (SWR) as a benchmark, CBR was found to consistently outperform SWR [20]. Several more recent studies, however, failed to replicate

these results [3]. Closer investigation revealed that this later work used relatively large case bases with more than 40 features. Unfortunately, this prevented them from using an effective feature subset selection approach, instead applying a simple filter method based on a t test. This then initiated research on the use of meta-heuristic search techniques and, subsequently, we have successfully used greedy search methods, such as forward selection search, to yield good results from large case-bases [12, 13].

Our previous work differs from many other CBR approaches in that we have not made significant use of adaptation, that is modifying the solution(s) of retrieved cases in some systematic way. Effectively we have used a k-Nearest Neighbour (k-NN) method using inverse distance weighting. In this paper we address the question does case adaptation improve the quality of our predictions? To explore this question we use three different project data sets that are representative of the different data sets we encounter in software effort prediction.

The remainder of this paper is organised as follows. The next section reviews different case adaptation strategies and then describes a structural adaptation algorithm that has been successfully applied to web projects. The following section provides background on the three case bases used for our analysis. We then describe our method of data collection and analysis. This is followed by the study's results. We conclude with a discussion of the results and make suggestions for follow up work.

2 Related Work on Case Adaptation

One aspect of CBR that is attracting much interest is adaptation. This involves modification of the proposed solution in order to better fit the target case. As well as enabling CBR systems to accommodate novel situations, adaptation may also be useful in counteracting the impact of occasionally retrieving poor cases [18].

The value of adaptation has been investigated by many researchers with varying results. The need for adaptation seems to be largely application dependent. For example, as suggested by Hanney et al. [7], classification tasks might be accomplished with little or no adaptation, while design and prediction applications call for varying degrees of adaptation strategies to achieve acceptable outcomes. Another challenge relates to the difficulties of eliciting the adaptation knowledge [4] although a range of new techniques such as the incremental approach are being investigated [10].

Wilke and Bergmann [25] classify adaptation into three main types:

- null adaptation
- transformational adaptation
- generative adaptation

Null adaptation, the simplest, involves directly applying the solution from the retrieved case(s) to the target case. This is the approach adopted by a simple

Nearest Neighbour technique and in a slightly more sophisticated form such as inverse distance weighted mean for kNN when $k > 1$.

With transformational adaptation, the old solution derived from the retrieved case is modified. There are two general approaches to achieving this. First, there is what is often termed *structural* transformation based on some function of the target and retrieved case feature vectors. Examples include Finnie et al. [5] and Hanney and Keane [8]. The other approach — often used when dealing with more complex problem representations — is *rule-based* transformation. Here, rules are either elicited from a domain expert or learnt using an induction algorithm. The use of fuzzy rule induction has also been proposed, see Shiu et al. [21].

Generative adaptation entails deriving the solution to the problem from-scratch. In principle, the derivation is handled by the case-based system, largely independent of the case base. Voss [23] describes a number of examples of this approach and more recently Munoz-Avila et al. described a hybrid generative adaptation method that involves user interaction [16].

In the field of software project prediction, cEstor is an early example of an adaptive case-based reasoning system developed by Prietula et al. [17]. The case adaptation knowledge was actually acquired in the raw form from an expert doing the task. This knowledge was translated into procedural rules in the form of `if <conditions> then <actions>.` The result was good predictions but a lack of generality even to other data sets in the same problem domain.

Finnie et al. [5, 6] used structural adaptation for predicting effort using CBR. Their adaptation model was primarily based on the relative size of the source and the target case and involved adaptation by means of a simple linear regression model. Effort was estimated by using a multiplier computed on the basis of the contribution of the selected features to productivity. Overall they found MMREs[3] for a simple regression model of 62.3%, neural net 35.2% and CBR 36.2% (smaller MMREs are preferred). Structural adaptation has also been applied by Walkerden and Jeffery [24] and Mendes et al. [15]. Both studies employ adaptation rules based on the linear size adjustment to the estimated effort. The linear size adjustment attempts to take into account the difference in size between the target and finished projects. For Walkerden and Jeffery, once the most similar finished project in the case base has been retrieved, its effort value is adjusted to estimate effort for the target project. A linear extrapolation is performed along the dimension of a single 'size' feature that is chosen as being strongly correlated with effort. The linear size adjustment is represented as follows:

$$e'_i = e_i \frac{s_t}{s} \tag{1}$$

where e is the actual effort of a retrieved project, s is the value of a size related feature for that project and e' is the adjusted effort value to be used in calculating the predicted effort for the target case. Note that s_t is the value for

[3] Mean magnitude of relative error (MMRE) is a widely used indicator of prediction accuracy and is defined as $\frac{100}{n} \sum_{i=1}^{i=n} \frac{|x_i - \hat{x_i}|}{x_i}$ where n is the number of predictions $\hat{x}$ of x.

the size feature that typically might be a measure of functionality described in the system specification using function points [22]. Mendes et al. [15] apply two types of adaptation rules, both based on linear size adjustment. The first type is called "adaptation without weights", and calculated by generalising the linear size adjustment to an arbitrary number of size related features, and then the estimated efforts generated averaged to obtain an effort estimate (Equation 2). When using this adaptation, all size measures contribute equally towards total estimated effort, indicated by the use of a simple average.

$$\hat{e}_t = \frac{1}{k} \sum_{i=1}^{i=k} \left(\frac{1}{q} \sum_{j=1}^{j=q} e_i \frac{s_{jt}}{s_{ji}} \right) \tag{2}$$

where $\hat{e}_t$ is the predicted effort for the target case, we are basing the prediction on k retrieved cases and there are q size related features. We denote each such feature as $s_{1i} \ldots s_{qi}$ for the ith retrieved case.

The second type of adaptation rule they used is called "adaptation with weights". In this type of adaptation, different weights are applied to size metrics to indicate the strength of relationship between a size metric and effort (see Equation 3).

$$\hat{e}_t = \frac{1}{k} \sum_{i=1}^{i=k} \left(\frac{1}{\sum_{j=1}^{j=q} w_j} \left(\frac{1}{q} \sum_{j=1}^{j=q} e_i \frac{s_{jt}}{s_{ji}} \right) \right) \tag{3}$$

where w_j is the weight or relative significance w attributed to the jth size measure s.

As stated in the introduction, to date, our approach has been a null adaptation, and to focus on feature and case subset selection in order to reduce the likelihood of retrieving poor analogies. Given the positive results of other researchers in this problem domain we now examine the impact of using a structural adaptation method i.e. a linear size adjustment similar to that employed by Mendes et al. [15]. This approach is selected because our solution representation is trivial (a single continuous feature) and the case representation is merely a feature vector so the more complex adaptation strategies appear unwarranted at this stage. Moreover, we wish to have a method that generalises to many data sets, unlike say, the method of Prietula et al. [17] where the adaptation rules are couched in terms of the specific features of a particular data set. In addition, we wish to avoid the problems of knowledge elicitation given that we do not possess any deep theory of software project management!

3 Our Data Sets

In this section we provide some background on the three data sets used in our study. These are chosen to represent varying sizes of data set that are commonly encountered in the project prediction domain. The data sets are:

– BT: a small data set ($n = 18$) derived from one division of a large telecommunications company. This is representative of many organisations that embark upon an internal data collection programme to support their effort prediction activities. The data is relatively homogeneous.
– Desharnais: a medium sized data set ($n = 77$) collected by a Canadian software house from projects distributed amongst 11 different organisations.
– Finnish: a large data set ($n = 405$) collected by the benchmarking organisation STTF Ltd. This data is collected over a number of years for a diverse range of software developers thus this is the most heterogeneous of the three data sets. Therefore the data set used in this paper is at the large end of this spectrum. The features are a mixture of continuous, discrete and categorical. However, there are a number of missing data values and also some features that would not be known at prediction time and so are not included in our analysis. Removing features with missing values or after-the-event data, leaves a subset of 42 features that are actually used in the case study. The data set also exhibits significant multi-collinearity, in other words there are strong relationships between features as well as with the feature to be predicted, namely effort.

Data set	no. of cases (n)	no. of features (p)	no. of continuous features
BT	18	3	3
Desharnais	77	9	8
Finnish	405	42	37

Table 1. Example Data Set Classification Scheme

Table 1 provides some summary information for each of the three data sets. It must be emphasised that the data sets are quite varied not only in terms of size but also in terms of the specific features that have been collected. Building prediction systems from such small case bases as exemplified by the BT data set is a common challenge in this problem domain. The three data sets contrast considerably, from the extreme simplicity of the BT data set to the large number of features and cases contained in the Finnish data set. In large data sets, such as the Finnish, there is clearly more scope for feature subset selection and potentially for adaptation to overcome problems of poor analogies. For this reason a sub-research question is: what, if any, relationship exists between data set complexity and the value of case adaptation?

4 Method

The techniques to be compared in this study are variants of a linear size adjustment scheme. The techniques used are based on those of Mendes [15], but with some amendments. This study does not assume that all available features

are suitable for scaling. Firstly, the data sets used in this paper contain some categorical features and these features are clearly not suitable for linear adjustment. Although differences in categorical features could be used for adaptation we will leave this for further work and concentrate on only the continuous features. Secondly, not all of the continuous features may be suitable for adaptation. We use robust correlation (e.g., Spearman's rank correlation) as a means of selecting which features will be scaled. The rationale here is that linear scaling would only work if there were a monotonic relationship (at least locally to the target case) between a particular feature and effort. The correlation is intended to be an indication of whether such a relationship exists and so whether linear adaptation is likely to be useful for that feature.

Two variants of linear adaptation are are investigated in this paper. In variant one, only the most highly correlated feature (either positively or negatively) is used for linear size adjustment (single feature adjustment). The second variant applies size adjustment to any feature that is significantly correlated with the dependent variable (multiple feature adjustment).

This study is restricted to the unweighted feature formulation presented in Equation 2. However, there are some issues with this formulation that need to be addressed. Firstly, it is assumed that if $s_{ij} > s_{tj}$ then the effort should be adjusted upward, i.e., that all features are positively correlated to effort. Although this may have been a reasonable assumption in the study from Mendes [15] where all features were size measures, this may not be the case for the data sets in this study where the features represent various attributes of the systems under development or the development environment, e.g. the level of reuse or experience. An alternative formulation for negatively correlated features is presented in Equation 4. Here s_{ij} and s_{tj} have been swapped so that effort will be adjust negatively if s_{ij} increases.

$$\hat{e}_t = \frac{1}{q} \sum_{j=1}^{j=n} e_i \frac{s_{ji}}{s_{jt}} \tag{4}$$

Another issue is that, in Equation 2, $\hat{e}_t$ will become infinite if $s_{ij} = 0$ (as it would if $s_{tj} = 0$ in Equation 4). To avoid this problem, features that would introduce a zero into the denominator (for a particular case) are excluded in the calculation (for that case).

The work of the study will be to build prediction systems based on the variants of linear size adjustment previously described and compare their accuracy. Standard prediction system validation requires a training set and a hold-out set, however, previous studies [9, 11] have shown that results vary widely depending on the random allocation of cases in these sets due to the heterogeneity of the data set. Large numbers of such sets may be necessary to provide acceptable confidence limits on the result and to allow statistical testing of apparent differences. The results gained from these adaptation techniques are also compared to case-based prediction without adaptation (simple k-NN prediction) to provide a benchmark.

Previous studies have shown that feature subset selection (FSS) prior to building case-based prediction systems can greatly improve their accuracy [1, 2, 12, 13]. This study investigates whether these strategies can improve a prediction system already tuned using FSS. Although, ideally, FSS should be repeated for each training set, it is computationally prohibitive given the large numbers of prediction systems to be built. Instead we perform one FSS for each treatment of each data set, based on a jack-knife [4] of the entire data set. This means that the same feature subset is used for each training set within a treatment and for some of these training sets it will be sub-optimal. However, since a previous study [13] has shown the optimal feature subset varies little with variations in the randomly sampled cases present in the training set, this should have little impact on the results.

Prediction accuracy is measured using the MMRE and the Sum of absolute residuals ($Sum(|r|)$). MMRE is chosen as it is a standard measure of prediction accuracy in software effort prediction and also because it allows comparison between data sets. The sum of absolute residuals is also used as it is less vulnerable to bias than the asymmetric MMRE [14]. Since the absolute residuals were skewed, we used a non-parametric test to check the significance of our results. The data was naturally paired (the same 100 training and validation sets were used within each data set), so we used the Wilcoxon Signed Rank test to compare the difference in location between two populations. The procedure followed for the data collection is given below:

1. Remove any after-the-event [5] features from the data (other than the dependent variable).
2. No adaptation (simple k-NN)
   (a) Do feature subset selection
   Subset search settings:
   - Use mean of the 2 nearest neighbours as the prediction.
   - Jackknife the data set and use the $Sum(|r|)$ as the measure of prediction system accuracy.
   - Search for the feature set that gives the lowest $Sum(|r|)$. Use the most effective search technique for the size of the data set and number of features (by exhaustive search other than for the Finnish data set which uses a forward selection search).

   (b) Run 100 validation trials with randomly sampled training sets with a 2:1 split between training and hold-out sets.
   (c) Collect MMRE and $Sum(|r|)$ values for each of the 100 sets.
3. Single feature adaptation

[4] Jack-knifing is a form of hold-one-out validation where each case is removed from the training set in turn and the remaining cases used to make a prediction for the holdout case

[5] These are features present in the published data sets that would not be known at prediction time.

(a) Calculate the correlations between the dependent variable and the other (continuous) features. As some of the features are measured on an interval scale and most of the continuous variables are not normally distributed the non-parametric Spearman rank correlation is used.
(b) Do feature subset selection. Subset search settings will be as for 'No adaption' except that linear distance adjustment is used on the feature with the highest correlation (the adjustment is done in the direction of that correlation - positive or negative).
(c) Run 100 validation trials based on the same 100 training sets as for 'no adaptation'.
(d) Collect MMRE and $Sum(|r|)$ values.

4. Multiple feature adaptation
(a) Calculate the significant correlations between the dependent variable and the other features at a significance level of α =0.05). Bonferroni adjustment will be used to adjust α based on the number of features to be tested.
(b) Do feature subset selection. Subset search settings are as for 'single feature adaptation' except that all features significantly correlated with effort are used for adjustment.
(c) Run 100 validation trials based on the same 100 training sets as for 'no adaptation'.
(d) Collect MMRE and $Sum(|r|)$ values.

5 Results

5.1 BT Data Set

No Adaptation: Given that the BT data set (following the removal of after-the-event features) contained only 3 independent variables an exhaustive feature selection was straightforward and resulted in only one variable being selected for use. The $Sum(|r|)$ results from running the 100 validation trials using just this feature and no adaptation are given in first boxplot of Figure 1. The median $Sum(|r|)$ is 872 and the median MMRE is 46.6%.

Single Feature Adaptation: In single feature adaptation the feature with the highest Spearman rank correlation is selected. The feature subset selection using single feature adaptation selected two variables. The results for the validation of the single adaptation technique are shown in the second boxplot of Figure 1. The median $Sum(|r|)$ is 877 and the median MMRE is 63.1%.

Multiple feature adaptation: Both (continuous) features must were significantly correlated and so both were adapted.

The feature subset search using multiple feature adaptation chose the same features as for single feature adaptation. As only one of these FSS selcted features was adapted, the results are identical. These are shown in the third boxplot of Figure 1.

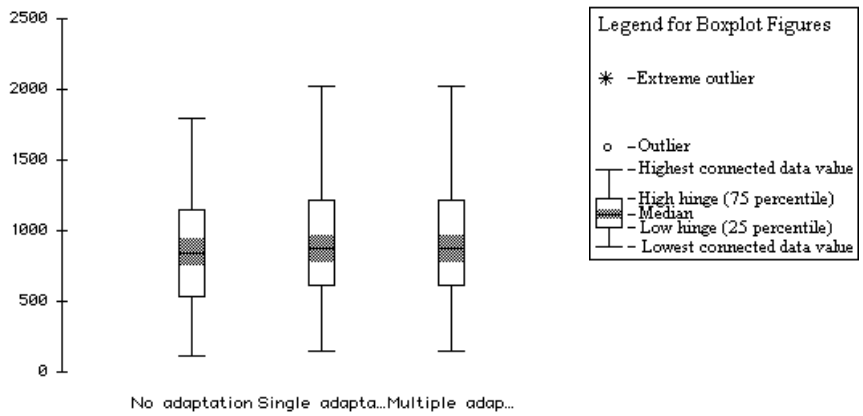

Fig. 1. Boxplot of results for BT data set

Summary of BT results: Using a two-tailed Wilcoxon signed rank test on the $Sum(|r|)$ values ($\alpha = 0.05$), there are no significant differences between the results from the different treatments indicating we have no grounds for believing that adjustment has either a positive or negative impact upon the accuracy of predictions for this data set.

5.2 Desharnais Data Set

No adaptation: Following the removal of after-the-event features the Desharnais data set contains 9 independent variables, so an exhaustive feature selection was also possible with this data set. The feature subset search resulted in three variables being selected for adaptation. The results from running the 100 validation trials using just this feature set and no adaptation are given in the first boxplot of Figure 2. The median $Sum(|r|)$ is 49377 and the median MMRE is 51.6%.

Single feature adaptation: With single feature adaptation the feature subset selected contained three variables (2 continuous and 1 categorical). The results for the validation of the single adaptation technique are shown in the second boxplot of Figure 2. The median $Sum(|r|)$ is 44754 and the median MMRE is 41.2%.

Multiple feature adaptation: The correlations for four of the features are significant and these are therefore selected for adjustment.

Feature subset selection chose the same three features as for single feature adaptation, and the only feature which is scaled is also the same, the results are identical. These are shown in the third boxplot of Figure 2.

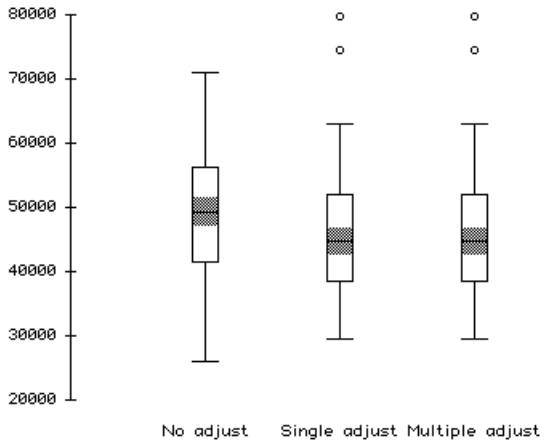

Fig. 2. Boxplot of results for Desharnais data set

Summary of Desharnais results: Using a two-tailed Wilcoxon signed rank test ($\alpha = 0.05$), there are significant differences between the results from the different treatments. Both adaptation techniques give better results than the simple k-NN treatment ($p \leq 0.0001$).

5.3 Finnish Data Set

No adaptation: After the removal of features and cases with missing data, the Finnish data set contains 42 independent variables, so an exhaustive feature selection was not possible with this data set. The alternative forward selection search strategy was used instead. The feature subset search resulted in four variables being selected for use. The results from running the 100 validation trials using just this feature set and no adaptation are given in first boxplot of Figure 3. The median $Sum(|r|)$ is 358022 and the median MMRE is 108.8%.

Single feature adaptation: The results for the validation of the single adaptation technique are shown in the second boxplot of Figure 3. The median $Sum(|r|)$ is 320645 and the median MMRE is 71.7%.

Multiple feature adaptation: Since the adaptation must be done in the direction of the correlation, significant positive and negative correlations are handled separately. Of the 37 continuous variables, 16 features were selected for positive adaptation and 10 features for negative adaptation.

The feature subset selected consisted of six features and although this feature set is different from that used for 'single adaptation' (and therefore has different results), there is actually only one scaled variable used.

The results for the validation of the multiple adaptation technique are shown in the third boxplot of Figure 3. The median $Sum(|r|)$ is 310190 and the median MMRE is 71.2%.

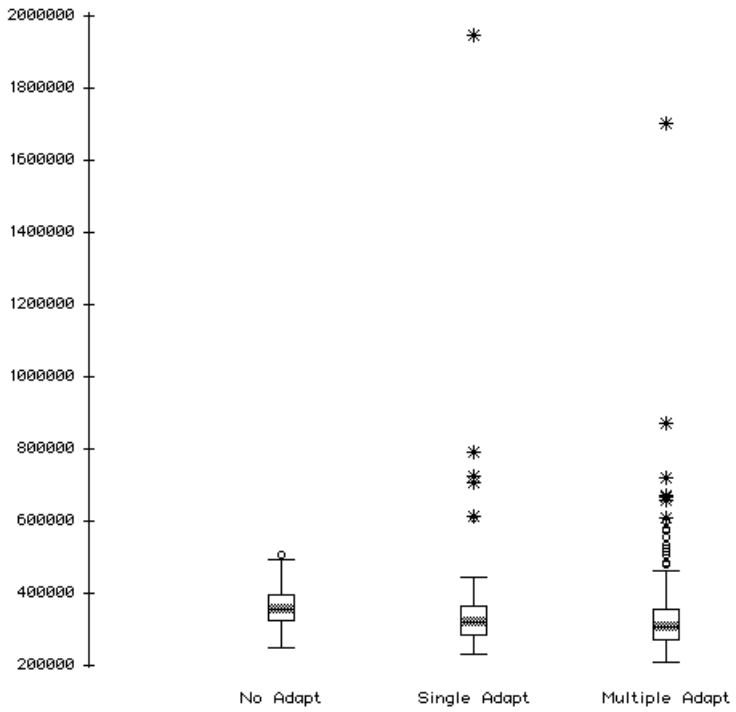

Fig. 3. Boxplot of results for Finnish dataset

Summary of Finnish results: Using a two-tailed Wilcoxon signed rank test ($\alpha = 0.05$), there are significant differences between the results from the different treatments. Both single adaptation and multiple adaptation give significantly better results than the simple k-NN treatment ($p \leq 0.0001$ and $p = 0.0069$ respectively). The difference between single adaptation and multiple adaptation is not significant ($p = 0.0555$).

5.4 Analysis of extreme outliers

One notable feature of the results where adaptation was applied to the Finnish data set is the large number of extreme outliers produced. Each outlier represents one sampled training set where the $Sum(|r|)$ of the predictions made

was particularly high. Further investigation showed that rather than particular training sets producing generally poor results, the poorer $Sum(|r|)$ values were caused by a few extreme predictions. These extreme predictions were in turn caused by extreme adjustment multipliers on these predictions (sometimes as much as several hundred).

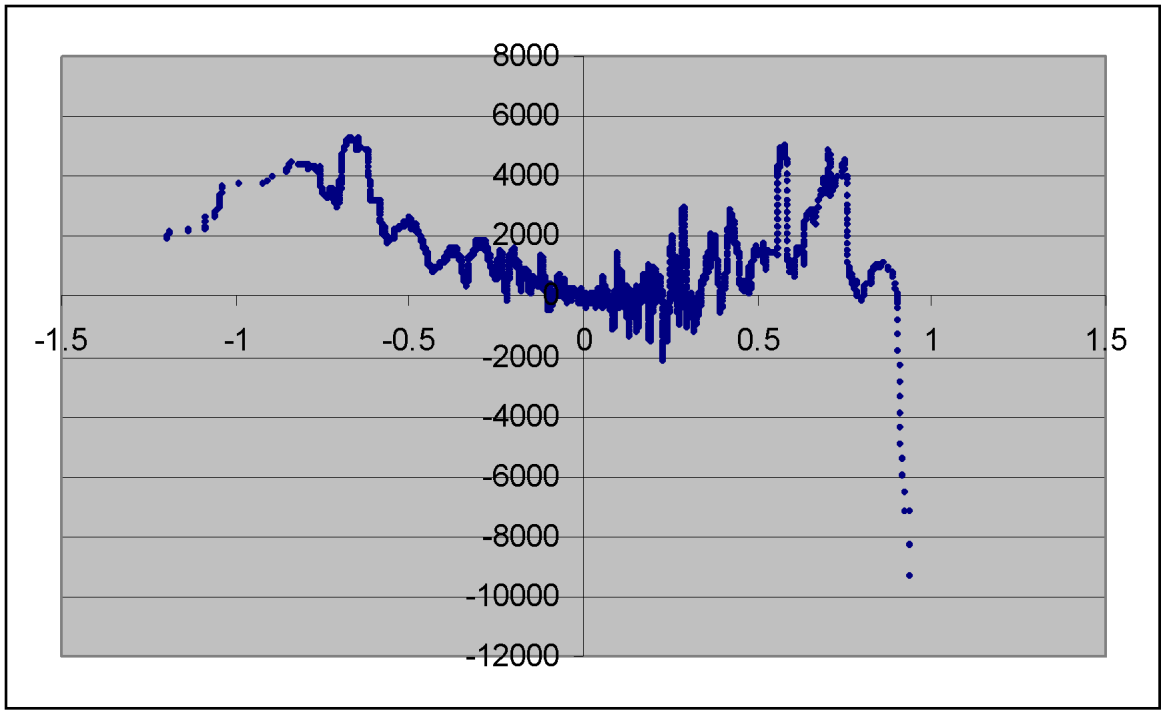

Fig. 4. Scatter plot of scaling against improvement in prediction

What we're particularly interested in is whether linear size adjustment improves prediction as compared to the basic k-NN approach. If there is a relationship between the size of the scaling used when doing adjustment and the likely improvement in prediction, then it may be possible to apply transformation or capping to the scaling values to reduce the number and size of outliers. In order to investigate this further, the relationship between the size of the scaling and improvement in prediction was examined. Figure 4 shows a smoothed plot of scaling against improvement for single feature adaptation of the Finnish data set. The large amount of random variation in the $Sum(|r|)$ values required smoothing to extract the underlying trend (here a rolling mean of 100 predictions was used). Scaling is shown on a Log_{10} scale to compress the higher scaling values

whilst still giving detail for small fractional values present. The points with the most extreme negative improvement values have been omitted to improve the clarity of the diagram.

Figure 4 clearly shows a pattern of improvement against scaling. For scaling values close to zero, improvement is close to zero. This would be expected since at scaling = 0 linear size adjustment is equivalent to k-NN and so the results should not differ. As the scaling moves away from zero (in both positive and negative directions) the adjustment improves the accuracy of the predictions. As the scaling continues to move away from zero the level of improvement starts to reduce. For very large adjustment values linear size adjustment produces less accurate predictions. Extrapolating the trend in very small (fractional) scaling suggests that the same would also happen in the negative direction. From figure 4 we can estimate that (for this data set) scaling values above 8 and below 0.05 will (on average) give worse predictions. These values could be used as capping limits on the scaling values.

6 Discussion and Conclusions

Table 2 gives a summary of the results. Two of the three data sets showed statistically significantly improved results when using linear size adjustment whilst the smallest data set (BT) showed no significant difference in either direction. So it seems that linear scaling adaptation generally improves the accuracy of the predictions. It must be noted, however, that although the improvements were relatively modest (improvements in median $Sum(|r|)$ were 9.4% for Desharnais and 13.4% for the Finnish data set) these are cumulative improvements over and above those possible through feature subset selection. Moreover, given the value of many software projects, 10% of total costs may well represent a considerable amount of money. Also, this adaptation approach is entirely automated which yields the advantage of avoiding explicit knowledge elicitation. It can also be applied to different data sets containing different features.

	BT		Desharnais		Finnish	
	$Sum(\|r\|)$	MMRE	$Sum(\|r\|)$	MMRE	$Sum(\|r\|)$	MMRE
No adaptation (k-NN)	872	46.6%	49377	51.6%	358022	108.8%
Single adaptation	877	63.1%	44754	41.2%	320645	71.7%
Multiple adaptation	877	63.1%	44754	41.2%	310190	71.2%

Table 2. Summary of results

Applying adaptation to the Finnish data set produced a significant number of extreme outliers. Linear size adjustment assumes that similar cases will have an approximately linear relationship between the adapted features and the target feature. For the Finnish data set, the adapted feature was a size measure

(Function Points) and the target feature was effort. The root cause of the outliers was the large variation in productivity (size/effort) for the projects. When apparently similar cases have widely varying productivity, gross errors in estimation can result. Although the intention was to see the effect of both single and multiple features being adapted, the suggested method selected a single primary size measure for scaling in each case. This was due to the combination of using feature subset selection to improve accuracy (which also reduces the number of available features) and also the introduced requirement for a feature to be significantly correlated before it could be considered for adjustment.

The combination of the modest effect size of the improvements and the presence of extreme outliers means that, although this form of adaptation did show significant improvements, the authors would advise some caution in using this method in practice. Further work is necessary to investigate whether transforming or clamping the magnitude of the scaling would solve the outlier problem.

Acknowledgments

The authors are indebted to STTF Ltd for making the "Finnish" data set available and to Jean-Marc Desharnais and BT for their respective data sets.

References

1. Aha, D. W. and R. L. Bankert, 'Feature selection for case-based classification of cloud types: an empirical comparison'. *AAAI-94 Workshop on Case-based Reasoning*, 1994.
2. Aha, D.W. and R.L. Bankert, 'A comparative evaluation of sequential feature selection algorithms', in *Artificial Intelligence and Statistics V.*, Fisher, D. and Lenz, J.-H., Editors, Springer-Verlag: New York, 1996.
3. Briand, L., T. Langley and I. Wieczorek, 'Using the European Space Agency data set: a replicated assessment and comparison of common software cost modeling techniques'. *22nd IEEE Intl. Conf. on Softw. Eng.*, Limerick, Ireland, Computer Society Press, 2000.
4. Craw, S., J. Jarmulak and R. Rowe, 'Learning and applying case-based adaptation knowledge'. Case-Based Reasoning Research and Development, Proceedings, Lecture Notes in Artificial Intelligence, No. 2080, pp131-145,Springer-Verlag, 2001.
5. Finnie, G.R., G.E. Wittig and J.-M. Desharnais, 'Estimating software development effort with case-based reasoning'. *2nd Intl. Conf. on Case-Based Reasoning*, 1997.
6. Finnie, G.R., G.E. Wittig and J.-M. Desharnais, A comparison of software effort estimation techniques using function points with neural networks, case based reasoning and regression models. *J. of Systems & Software*, 39, pp281-289, 1997.
7. Hanney, K. et al. 'When Do You Need Adaptation?: A Review of Current Practice'. *AAAI-95 Fall Symposium on Adaptation in Knowledge Reuse*, Cambridge, MA, USA, 1995.
8. Hanney, K. and M. T. Keane, 'The adaptation knowledge bottleneck: how to ease it by learning from cases', *2nd Intl. Conf. on Case-Based Reasoning*, 1997.

9. Kadoda, G., M. Cartwright, L. Chen, and M. Shepperd, 'Experiences using case-based reasoning to predict software project effort'. *4th International Conference on Empirical Assessment and Evaluation in Software Engineering*, 17-19 April 2000, Keele University, UK.
10. Khan, A. S. and A. Hoffmann, 'A new approach for the incremental development of adaptation functions for CBR'. Advances in Case-Based Reasoning, Proceedings, Lecture Notes in Artificial Intelligence, No. 1898, pp260-272, Springer-Verlag, 2001.
11. Kirsopp, C. and M. Shepperd, 'Making inferences with small numbers of training sets', *IEE Proceedings - Software* 149(5), 2002.
12. Kirsopp, C., M. Shepperd and J. Hart, 'Search heuristics, case-based reasoning and software project effort prediction', *Genetic and Evolutionary Computation Conference*, New York, USA, July 9-13, 2002.
13. Kirsopp,C. and M. Shepperd, 'Case and Feature Subset Selection in Case-Based Software Project Effort Prediction', Research and Development in Intelligent Systems XIX, Springer-Verlag, 2002.
14. Kitchenham, B.A., S.G. MacDonell, L. Pickard and M.J. Shepperd, 'What accuracy statistics really measure', *IEE Proceedings - Software Engineering* 148(3): pp81-85, 2001.
15. Mendes, E., S. Counsell, and N. Mosley, 'Investigating the use of case-based reasoning adaptation rules for web project cost estimation'. submitted paper, World Wide Web conference 2003.
16. Munoz-Avila, H., D.W. Aha, L.A. Breslow, D.S. Nau and R. Weber, 'Integrating conversational case retrieval with generative planning', in Advances in *Case-Based Reasoning, Proceedings*, Blanzieri, E. and Portinale, L., (Editors), Springer-Verlag. pp210-221, 2001.
17. Prietula, M.J., S.S. Vincinanza and T. Mukhopadhyay 'Software Effort Estimation With a Case-Based Reasoner', *J. of Experimental & Theoretical Artificial Intelligence* 8, pp341-363, 1996.
18. Scott, S., H. Osborne and R. Simpson, 'Assessing Case Value in Case-Based Reasoning with Adaptation'. *World Multiconference on Systemics, Cybernetics and Informatics* (IIIS-99), Orlando, Florida, 1999.
19. Shepperd, M., C. Schofield and B.A. Kitchenham 'Effort estimation using analogy'. *18th Intl. Conf. on Softw. Eng.*, Berlin, IEEE Computer Press, 1996.
20. Shepperd, M. and C. Schofield.'Estimating software project effort using analogies', *IEEE Transactions on Software Engineering* 23(11) pp736-743, 1997.
21. Shiu, S. C. K., D. S. Yeung, C. H. Sun and X. Z. Wang, 'Transferring case knowledge to adaptation knowledge: An approach for case-base maintenance.' *Computational Intelligence* 17(2) pp95-314, 2001.
22. Symons, C.R., 'Function point analysis: difficulties and improvements', *IEEE Transactions on Software Engineering* 14(1): pp2-11, 1988.
23. Voss, A. and R. Oxman. 'A study of case adaptation systems'. *Artificial Intelligence in Design*, pp173-189, Kluwer, 1996.
24. Walkerden, F and R. Jeffery. 'An empirical study of analogy-based software effort estimation'. *Empirical Software Engineering* 4(2),pp135-158, 1999.
25. Wilke, W. and R. Bergmann, 'Techniques and knowledge used for adaptation during case-based problem solving'. *Tasks and Methods in Applied Artificial Intelligence*, LNAI 1416, pp497-505, Springer-Verlag: Berlin, 1998.