The Consistency of Empirical Comparisons of Regression and Analogy-based Software Project Cost Prediction

Carolyn Mair and Martin Shepperd
Brunel University, UK
{carolyn.mair, martin.shepperd}@brunel.ac.uk

Abstract

OBJECTIVE - to determine the consistency within and between results in empirical studies of software engineering cost estimation. We focus on regression and analogy techniques as these are commonly used.
METHOD – we conducted an exhaustive search using predefined inclusion and exclusion criteria and identified 67 journal papers and 104 conference papers. From this sample we identified 11 journal papers and 9 conference papers that used both methods.
RESULTS – our analysis found that about 25% of studies were internally inconclusive. We also found that there is approximately equal evidence in favour of, and against analogy-based methods.
CONCLUSIONS – we confirm the lack of consistency in the findings and argue that this inconsistent pattern from 20 different studies comparing regression and analogy is somewhat disturbing. It suggests that we need to ask more detailed questions than just: "What is the best prediction system?"

1. Introduction

Software engineering as a discipline demands theories grounded in real world evidence, but unfortunately researchers frequently find that investigations into similar phenomena conducted in different studies produce conflicting results. These differences may arise from a number of issues including heterogeneous sampling methods, measurement or reporting which makes comparing and combining results problematic.
This study focuses upon empirical studies that have compared the accuracy levels of two competing software project cost prediction systems, namely regression analysis and analogy. We seek to answer the question to what extent are the empirical results consistent between and also within studies.

Empirical studies in software engineering are used to investigate the efficacy of methods and the impact of various factors on productivity, quality or cost. Hence, 'the reliable combination of results from independent experiments is an essential building block in any discipline attempting to build a solid empirical foundation' [1]. However, different studies of the same phenomenon frequently report different findings perhaps because the data have been collected or analysed in a non-standardised manner. This problem is compounded because of difficulties defining the population to which the results can generalize and thus in obtaining a representative sample [2]. In addition, the selection of studies is influenced by heterogeneity of measures, environmental factors, publication bias and the 'file drawer problem' [3]. Therefore in order to perform 'a study of studies'[4], a meta-analysis [5], a systematic and documented procedure needs to be used to search and screen relevant studies, code results, and provide a quantitative summary of the findings [1, 6].

The existence of context variables poses a serious challenge to forming a body of knowledge in empirical software engineering. Basili, Shull *et al.* [7] suggest researchers should build models using a common framework for data collection to represent common observations that would allow generalization [8]. However these would involve replication of individual yet comparable studies in which results are refined rather than combined. In such cases, authors typically dismiss seemingly contradictory results rather than use them [9].
The need for empirical validation of different and often competing software project effort prediction systems, has led to hundreds of studies being conducted. Yet there remains a lack of synthesised findings. Thus, to the best of our knowledge, this investigation is the first systematic comparison of the empirical evidence for two competing prediction systems.

The remainder of this paper is organised as follows. The next section, Section 2, briefly describes regression and analogy techniques as applied in effort prediction, and goes on to summarise work in which both regression and analogy-based approaches are used. This body of work comprises the papers from conference proceedings and refereed software engineering journals identified using our search strategy which is documented in Section 3. Section 4 details the results of the analysis, while the conclusions are given in Section 5.

2. Related work

An effort prediction system can be derived using a number of techniques for example, expert judgement, statistical (e.g. regression), and more recently machine learning approaches (such as Artificial Neural Networks and Case-Based Reasoning (CBR)). These prediction systems typically have a primary cost factor such as size (typically lines of code (LOC) or function points (FP)) and a number of adjustment factors (cost drivers) which characterise the project and influence effort. Cost drivers are used to adjust the preliminary estimate provided by the primary cost factor [10].

As the 'best' technique frequently varies among studies, some researchers recommend at least two prediction approaches are needed in order to reduce risk [11]. This can be through the application of multiple techniques to different subsets of the data, or by using more than one technique to produce a range of estimated values [12]. Despite adopting these recommendations, to date no converging results have been obtained [e.g. 13, 14]. The most commonly applied prediction methods for software effort prediction systems are regression [15] and analogy-based [16]. In this paper, we investigate the consistency of results when using regression and analogy-based prediction methods. In the following subsections we define regression and analogy-based techniques, then briefly synthesise outline results from papers included in this study.

(i) Regression-based techniques are the most popular way of building prediction systems because they are simple and easy to use. However, the model must be specified first. For example, the dependent variable, effort is plotted against one or more independent variables, the cost drivers, to calculate the line of best fit among the data points. Standard regression refers to regression modelling using least squares in which the data is fitted to the pre-specified model in an attempt to minimise the overall sum of squared errors. However, standard regression models are sensitive to outliers which are common in software project data sets.

(ii) Analogy-based prediction, for example case-based reasoning (CBR), is a problem solving technique [17] which solves new problems by using and adapting solutions that have solved previous problems. In order to perform analogy-based prediction for software effort estimation, analogues, completed projects that are most similar to the new project are identified. With the help of experts or through automated tools, the user selects relevant cost drivers, similarity/distance functions (most frequently standardised Euclidean distance), and the number of analogues to consider. Examples of CBR tools for software prediction include Estor [18] and ANGEL [e.g. 16].

Varying results have been found within and among studies using multiple techniques and measures, and many researchers [19] suggest that the success of a technique is dependent on data set characteristics. For example, Myrveit and Stensrud [20] concluded that results are sensitive to experimental design after finding contradictory results when replicating previous studies which had claimed analogy outperformed regression. Similarly, Briand *et al.* [21] found analogy-based prediction systems were less robust than regression models when using data external to the organisation for which the model is built. In contrast, Mendes and Kitchenham [22] suggest that CBR predicts better across a large heterogeneous data set, and regression is better for within-company predictions. Finnie, Wittig and Desharnais [23, 24] claimed that because of the complexities involved in software development projects, regression models were less effective than analogy-based prediction systems which benefit from human judgement and intuition. Shepperd and Schofield [16] found that analogy gave better results than regression in terms accuracy, and Angelis and Stamelos [25] also found analogy-based methods superior except when using transformed data derived from non-parametric bootstrap methods. Mendes *et al.* [26] and Jeffery *et al.* [27] found that overall, stepwise regression outperformed CBR. However, in later work, Mendes [28] claimed that CBR gave better prediction accuracy. Briand *et al.* [21] concluded that using OLS is 'probably sufficient' (p.385). Niessink and van Vliet [29] proposed that analogy offers an alternative to regression models when applied to a heterogeneous data set.

The studies outlined above provide clear evidence of a lack of synthesis and consistency when attempting to determine which is the better prediction method: regression or analogy. In order to address this, we attempt to compare the consistency of results and conclusions within and between these studies in which regression and analogy-based prediction techniques are used. The rationale for our methods is described in the following section.

3. Method

As stated previously, this study compares results within and between empirical studies that use both regression and analogy prediction techniques for software project cost estimation. In order to clarify the evidence and to address the issues of inconsistency we have restricted the search to using predefined criteria described below. This study builds on our previous work [30] in which we identified publications that described empirical studies of effort prediction systems from three journals. Here we extend the search to include all software engineering journals plus conference proceedings.

In order to systematically compare the individual studies we define the following terminology and illustrate the structure of the data in Figure 1. A *study* is an empirical investigation that compares the performance of different prediction systems (in the case of this paper we focus upon comparisons of regression and analogy-based prediction systems). A study is described by one or more *papers*. Where the study is reported more than once, we use only the most recent description in order to avoid over-counting a single investigation. A study utilizes one or more *data sets* (in our investigation between 1 and 16 data sets) and uses one or more, *accuracy indicators*, typically MMRE and medianMRE (in our investigation between 1 and 7 accuracy measures). Next, for each data set there will be one or more *runs* depending upon the cross-validation mechanism, for example *n*-fold validation implies *n* runs. Finally, some studies were constructed so that there were additional accuracy comparisons that didn't easily fit into any of the above. We termed these *other comparisons*. The total number of *comparisons* for a study is the number of pairs of accuracy indicators reported and is the product of the count of data sets, accuracy indicators and runs plus any other comparisons. The comparisons are then interpreted in terms of preference relations: prefer analogy, prefer regression, indifferent. For example, if a study reports that for a particular comparison, accuracies of MMRE=40% for analogy and MMRE=38% for regression we record this as an occurrence of prefer

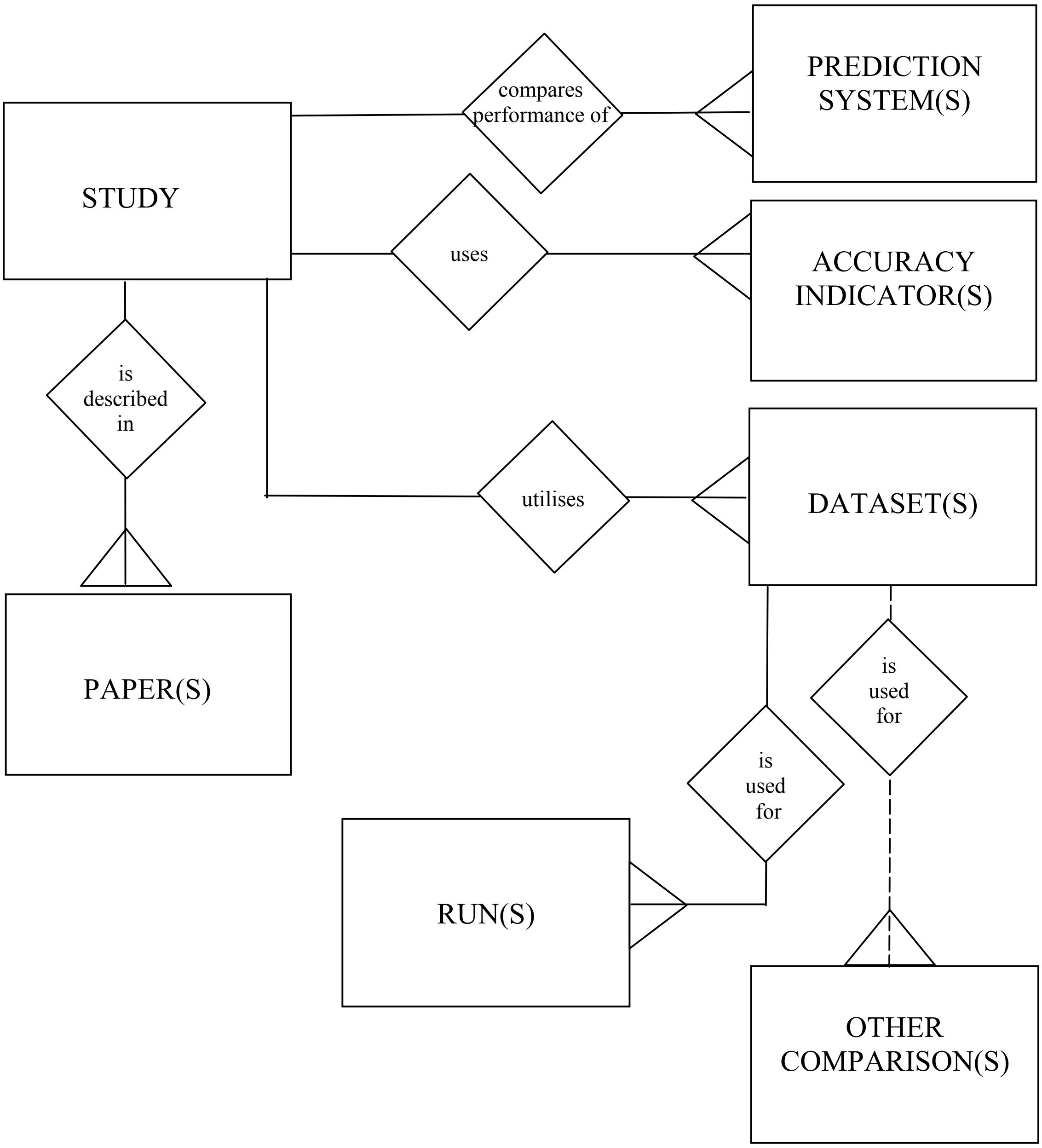

Figure 1 Entity relationship diagram showing the structure of the data

regression. The studies included in this study come from the population defined by these parameters:

- the study was peer reviewed either in a journal or in conference proceedings;
- the work is published in English;
- the study uses more than a single software project;
- the purpose of the estimation is cost (effort);
- the prediction techniques include both regression and analogy in order for us to make a comparison.

We conducted a search for refereed journal and conference papers using a personal informal bibliographic database, the Web of Knowledge (wok.mimas.ac.uk/), ScienceDirect (sciencedirect.com), IEEE Explore (ieeexplore.ieee.org) and Google (google.co.uk).

Table 1. Published journal and conference papers using regression and analogy-based software project effort prediction techniques

Authors	Publication	Year
Finnie, Wittig and Desharnais [24]	JSS	1997
Shepperd and Schofield [16]	TSE	1997
Finnie, Wittig and Desharnais [23]	ICCBR97	1997
Niessink and van Vliet [29]	ICSM97	1997
Hughes, Cunliffe and Young-Martos [31]	IEEE Proc. Software	1998
Myrtveit and Stensrud [20]	TSE	1999
Briand, El Emam, Surmann, Wieczorek and Maxwell [21]	ICSE99	1999
Angelis and Stamelos [25]	ESE	2000
Mair, Kadoda, Lefley, Phalp, Schofield, Shepperd and Webster [32]	JSS	2000
Jeffery, Ruhe and Wieczorek [33]	IST	2000
Briand, Langley and Wieczorek [34]	ICSE00	2000
Burgess and Lefley [35]	IST	2001
Shepperd and Kadoda [36]	TSE	2001
Jeffery, Ruhe and Wieczorek [27]	METRICS01	2001
Mendes and Mosley [26]	ISESE02	2002
Wieczorek and Ruhe [37]	METRICS02	2002
MacDonell and Shepperd [11]	JSS	2003
Mendes, Watson, Triggs, Mosley and Counsell [38]	ESE	2003
Mendes, Mosley and Counsell [39]	METRICS03	2003
Mendes and Kitchenham [22]	METRICS04	2004

This search identified 67 journal papers and 104 conference papers. From these studies, we identified 11 journal papers and 9 conference papers which used *both* regression and analogy-based techniques to predict software project cost on the *same* data set(s). A number of these studies covered additional techniques to regression and analogy; however, for the purpose of this analysis we extracted only results relevant to the main research question.

One difficulty was making judgements about what exactly constitutes regression and analogy since there are many variants of both techniques. In the end, because of most common usage, we counted any least squares regression method but excluded robust and logistic regression. Similarly we defined analogy as essentially being a variant of a k-nearest neighbour algorithm. Thus, for example, although the study conducted by Prietula *et al*. [40] was retrieved using our search criteria, it is not included in this investigation. Their approach, Estor, is a hybrid technique developed from rule-based reasoning and CBR.

We have not included unpublished or unreviewed material in our search strategy. One reason for this decision is the assumption that studies that have been subjected to peer review will be of higher quality. However, we recognise that our results might thus be subject to publication bias, and therefore, in future work we will attempt to retrieve all investigations published or otherwise. To summarise, our search identified 20 relevant papers: 11 journal papers and 9 conference proceedings. These cover the period 1997 to 2004 and are tabulated in Table 1.

4. Results

In this section we consider the results from combining the 20 individual studies identified using the inclusion criteria and terminology described previously. The overall results are given in Table 2. This shows for each study how many separate comparisons were reported between the accuracy of the regression and analogy-based prediction. Each comparison is then classified as: ‘prefer analogy’, ‘prefer regression’ or ‘indifferent’ (when it is not possible to differentiate in terms of accuracy). The counts are in columns 7 to 9 and the rightmost column gives the ratio of counts for analogy to regression, hence values greater than one imply support for analogy and values less than one imply support for regression-based predictors Note that DIV/0! arises when there is no support for regression.

By categorizing the ratios from Table 2 into support, i.e. two thirds or more of the comparisons prefer one or other of the prediction techniques; we observe mixed results in that there is no strong evidence to favour either technique. Alternatively taking an even simpler view, and using unity as the split point we have 12 studies tending towards Analogy and 8 for Regression. Again this is not compelling evidence. In other words, when we combine the 20 studies it is very difficult to decide which technique to prefer. **erw**

In addition, 6 of the papers (30%) also made use of statistical inference (e.g. *t*-test or Wilcoxon) in order to arrive at conclusions concerning the relative performance of regression and analogy-based prediction systems. Table 4 indicates that of the studies that carried out inferential tests comparing the residuals arising from each type of prediction, there is more support for regression-based models, in other words a slightly contradictory result to the analysis of Tables 2 and 3. However, this is a minority of the studies so again the overall significance is not entirely clear.

Table 2. Overall results of journal and conference papers using regression and analogy-based prediction methods

Study Ref.	# Data sets	# Accuracy indicators	# Comparisons			Empirical result in support of each prediction method			Ratio A/R
			Runs	Others	Total	Analogy	Regression	Indifferent	
[24]	1	1	1	0	1	1	0	0	#DIV/0!
[16]	9	2	1	19	37	30	3	4	10
[23]	1	1	1	0	1	1	0	0	#DIV/0!
[29]	1	2	12	0	24	24	0	0	#DIV/0!
[31]	1	7	3	2	23	14	9	0	1.56
[20]	1	4	3	0	12	0	12	0	0
[21]	3	3	2	0	18	6	12	0	0.50
[25]	2	2	1	0	4	2	1	1	2
[32]	1	1	3	0	3	2	1	0	2
[33]	2	3	4	0	24	0	21	3	0
[34]	3	2	1	0	6	0	6	0	0
[35]	1	6	2	0	12	3	9	0	0.33
[36]	16	1	2	0	32	20	12	0	1.67
[27]	3	4	2	2	26	12	13	1	0.92
[26]	1	3	20	0	60	0	54	6	0
[37]	6	3	2	0	36	23	9	4	2.56
[11]	1	3	2	0	6	1	5	0	0.2
[38]	2	2	3	0	12	9	3	0	3
[39]	2	2	3	0	12	9	3	0	3
[22]	4	5	1	0	20	12	7	1	1.71

Table 3. Summary of Study Support for Analogy and Regression-Based Prediction

Ratio of #Analogy/#Regression	Frequency
≤0.5 (Support for Regression)	7
0.51-1.99	4
≥2 (Support for Analogy)	9

Table 4. Summary of Study Support from Inferential Tests for Analogy and Regression-Based Prediction

Ratio of #Analogy/#Regression	Frequency
≤0.5 (Support for Regression)	4
0.51-1.99	1
≥2 (Support for Analogy)	1

Table 5. Summary of Qualitative Analysis of Analogy and Regression-Based Prediction Studies

Study Ref.	Conclusions	Quotation
[24]	++	Regression models performed poorly...
[16]	++	Estimation by analogy tends to be the more accurate prediction method…produces superior predictive performance in all cases when measured by MMRE and in seven out of nine cases for the Pred(25).
[23]	++	CBR has significant potential...it performs much better than a regression model. The prediction accuracy is as at least as good as any other models in use.
[29]	+	...the heterogeneity of the dataset remains in conflict with the 'one model fits all' flavour of regression type models. In such cases analogy-based estimation seems to offer an interesting alternative.
[31]	?	CBR worked well on both data sets but there appeared to be a tendency to underestimate. It would not be appropriate to draw too many conclusions...based on relatively small data sets. ... adopt a dual approach using LSR and CBR techniques.
[20]	-	MMRE favours analogy. R^2 favours regression...Adjusted results (outliers removed) favour regression using MMRE.
[21]	?	Analogy based models do not seem as robust when using data external to the organisation for which the model is built.
[25]	+	..analogy gives better results compared to both regression models in the case of the Albrecht data set. In the case of the Abran-Robillard data set, the analogy method performs better than the regression model with the original variables, while the regression model built on the transformed variables gives the best values of MMRE and Pred(25). ...we can see an obvious disadvantage of linear regression based on the original variables, i.e. the negative values in the estimates and in the lower bound of confidence intervals.
[32]	?	Regression analysis is a well established technique with good tool support...little effort needs to be expended in building a satisfactory regression model. Likewise CBR needs relatively little effort.
[33]	?	OLS regression performed as well as analogy-based estimation when using company-specific data. Using multiple-company data, OLS provided significantly more accurate results.
[34]	-	OLS provides the best results overall. Analogy does not seem to bring any specific advantage over other modelling techniques.
[35]	?	CBR needs relatively little work though more might be gained by relative weighting of the inputs.
[36]	+	...CBR was the best technique since it outperformed the other techniques on most occasions (11 out of 32). However, this would lead to choosing the wrong prediction system two thirds of the time.
[27]	?	OLS was more accurate when using multi-company [ISBSG] data. When using a company's own data...there was no significant difference between OLS and analogy.
[26]	- -	The technique that gave the best prediction accuracy was stepwise regression.
[37]	?	NB The paper focuses upon different research questions and therefore makes no direct comment upon regression vs. analogy.
[11]	-	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.
[38]	-	Stepwise regression showed the most accurate predictions...most of the time.
[39]	++	Regarding the prediction accuracy of CBR, compared to that for SWR, CBR clearly gave better predictions …
[22]	?	CBR was better for predictions across the large heterogeneous data set, but regression was better for within-company predictions.

Thus far we have considered only the quantitative aspects of the studies. However, authors also provide narrative interpretation of their research and comment on the significance of their findings. Therefore, we now move from a quantitative analysis to a more qualitative approach and examine the text — with a particular emphasis upon the abstract, discussion and conclusion sections — of the papers describing the studies in order to determine what conclusions the authors derive.

In Table 5 we have assigned a class indicating the strength of the conclusions in support of analogy-based prediction ranging from "++", "+", "?", "-" to "- -". Where "++" indicates strong support for analogy; "+"indicates support for analogy; "?"indicates indifference; "-"indicates support for regression and "- -" indicates strong support for regression. This classification is based on a reading of the text of the research paper and is a somewhat subjective process. Therefore we include quotations from the paper that we believe support our classification. One interesting observation is that since the studies in Table 5 are organized chronologically it is the earliest studies that tend to argue strongest support for analogy contrasting with a certain degree of ambivalence from the more recent studies.

Table 6 summarises the overall conclusions from the entire set of 20 papers. It is interesting that the modal class of study is non-committal and few studies (25%) make strong conclusions (in either direction) about the relative performance of the prediction systems.

Table 6. Summary of Study Conclusion Strengths

Conclusions strength	Frequency
++	4
+	3
?	8
-	4
- -	1

This classification seems to indicate a small inconsistency between the qualitative evidence and the kinds of conclusions drawn by the authors, specifically, there is a slight tendency for researchers to over interpret evidence in favour of the "new" technique of analogy-based prediction since regression is generally used as a benchmark.

In order to ascertain which variables if any were the best predictors of effort, we conducted a discriminant analysis. Using features: year, publication type, number of projects, number of data sets and prediction type, we found that no significant level of discrimination.

5. Conclusions

Over recent years there has been considerable activity by the empirical software engineering research community to investigate and compare competing software project cost prediction systems. In this paper we have examined two approaches that have received significant attention over the past ten years, namely regression and analogy-based techniques. A systematic review of the literature identified 20 empirical studies comparing the relative accuracy levels yielded by the two approaches. Unfortunately these studies yield little clear evidence as to which technique should be preferred with 45% offering some support for analogy, 35% for regression and 20% undecided.

The problem we need to address is: why are the results inconsistent? One might expect different results when models are generated from different data sets, however in some cases results were inconsistent despite utilising the same data set and the same prediction techniques. For example, [33], [36], and [22] each used the Desharnais [41] data set, but found conflicting results. In [33], CBR outperformed LSR in 2 out of the 3 comparisons performed in terms of MMRE, whereas LSR outperformed analogy in both comparisons in [36] in terms of MMRE and Pred(25). And although CBR resulted in a lower MMRE, Pred(25) was greater in comparisons with both regression models in [22]. In [33] and [22], projects with missing values were excluded; in [36], missing data were replaced by 'random samples from the other projects' (p.864). In [33] the holdout sample comprised 15% of the projects, in [36] it was 22%, whereas in [22] the entire data set was used to generate the regression models. Given the inconsistencies within and between these papers, we might conclude that variations in method are responsible.

The lack of standardisation in software engineering research methodology leads to heterogeneous sampling, measurement, and reporting techniques. These are compounded by the impact of different contexts and variants of the methods. From our results there would seem to be some evidence of both. As commented previously there are conflicting results from the *same* data set and therefore presumably the same context. On the other hand this is evident in a minority of results and other researchers have reported conflicting results conducted *within* the same study (e.g. [31, 36, 37]). The likely impact of context upon comparative accuracy of prediction techniques has also been noted in the simulation work of Shepperd and Kadoda [36]. Indeed it is hardly surprising that different data set characteristics will favour different techniques.

In order that our findings may be generalised, we need to consider the validity of our work. Because the validity of our study is contingent on the validity of the studies on which it is based, we firstly consider threats to validity within these papers. Studies such as these cannot be conducted blind, therefore researcher bias needs be considered. For example, a research group might have greater expertise in a particular method, or they may have pioneered a particular technique. Either or both of these could manifest in a disproportionate time to be spent on a 'pet technique' [42] in comparison to others. Clearly, this can lead to inconsistent results when comparing results from seemingly similar studies derived from different research groups. One solution might be to investigate the background of each research group with the aim of identifying whether or not they are pioneering, or are simply in favour of, a particular technique. Alternatively, the paper could state where the research group's interest s lie. Furthermore, as protocols were rarely described, replicating these studies would be problematic in most cases. For the purposes of this study, we assumed the data were unbiased.

If this were not the case, the validity of our work might be affected. A further possible threat is that we included only papers published in English. However, we do not consider this a serious threat. More importantly we made judgements as to what constituted regression and analogy-based techniques when each is in fact 'a family' of techniques. Therefore we grouped and treated equally all regression, except log linear, techniques and ignored any transformations of skewed variables. For analogy-based techniques we ignored parameters such as feature subset, adaptation, number of analogies and distance measures used. Had we used narrower definitions for each technique our results would likely have been more consistent. However, we took the view that if a technique is so sensitivity to minor variants in its deployment, the community needs to be aware of this factor.

The discriminant analysis failed to find a significant level of discrimination among the variables. Hence we conclude factors other than those accounted for in the present study, might contribute to the inconsistencies within and between results. We thus conclude that using available evidence, and broad definitions of each technique, neither dominates in the sense of always being preferable. Therefore, as a starting point researchers should ask questions such as when might it be better to use technique A rather B, as opposed to is technique A better than B?

Acknowledgement

This work is supported by the Engineering and Physical Sciences Research Council of the UK under grant GR/S45119.

References

[1] J. Miller, "Can results from software engineering experiments be safely combined?," presented at METRICS, 1999.

[2] L. Pickard, B. Kitchenham, and S. Linkman, "An investigation of analysis techniques for software data sets," *IEEE METRICS*, pp. 130-142, 1999.

[3] R. Rosenthal, "The "file-drawer" problem and tolerance for null results," *Psychological Bulletin*, vol. 86, pp. 638-641, 1979.

[4] J. S. Armstrong, "Principles of forecasting: A handbook for researchers and practitioners," in *International series in operations research & management science*. Boston: Kluwer Academic Publishers, 2001, pp. XII, 849 s. : ill.

[5] G. V. Glass, "Primary, Secondary and Meta-analysis of Research," *Educational Researcher*, vol. 5, pp. 3-8, 1976.

[6] J. Miller, "Applying meta-analytical procedures to software engineering experiments.," *Journal of Systems and Software*, pp. 29-39, 2000.

[7] V. R. Basili, F. Shull, and F. Lanubile, "Building knowledge through families of experiments," *IEEE Transactions on Software Engineering*, vol. 25, pp. 456-473, 1999.

[8] W. Hayes, "Research synthesis in software engineering: a case for meta-analysis," *IEEE METRICS*, pp. 143-151, 1999.

[9] N. E. Fenton and S. L. Pfleeger, *Software Metrics – A Rigorous and Practical Approach.*, 2nd Edition ed. Boston, USA:: PWS Publishing Company., 1997.

[10] C. Mair, M. Shepperd, and M. Jorgensen, "An Analysis of Data Sets Used to Train and Validate Cost Prediction Systems," presented at ICSE'05, St. Louis, USA, 2005.

[11] S. G. MacDonell and M. J. Shepperd, "Combining techniques to optimize effort predictions in software project management," *Journal of Systems and Software*, vol. 66, pp. 91-98, 2003.

[12] A. R. Gray and S. G. MacDonell, " A comparison of techniques for developing predictive models of software metrics," *Information & SoftwareTechnology*, vol. 39, pp. 425-437, 1997.

[13] E. Mendes, N. Mosley, and S. Counsell, "Using an engineering approach to understanding and predicting Web authoring and design," presented at Proceedings of Hawaii International Conference on System Sciences. HICSS-34. 3-6 Jan. 2001 Maui, HI, USA [Univ. Hawaii College of Bus. Adm], 2001.

[14] E. Mendes, N. Mosley, and I. Watson, "A Comparison of Case-Based Reasoning Approaches to Web Hypermedia Project Cost Estimation," presented at WWW2002, Honolulu, Hawaii, USA., 2002.

[15] F. Walkerden and R. Jeffery, "An empirical study of analogy-based software effort estimation," *Journal of Empirical Software Engineering*, vol. 4, pp. 135-158, 1999.

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

[17] T. Mukhopadhyay, S. S. Vicinanza, and M. J. Prietula, "Examining the feasibility of a case-based reasoning model for software effort

estimation," *MIS Quarterly*, vol. 16, pp. 155-171, 1992.

[18] T. Mukhopadhyay and S. Kekre, "Software Effort Models for Early Estimation of Process Control Applications," *IEEE Transactions on Software Engineering*, vol. 18, pp. 915-924, 1992.

[19] M. Shepperd and G. Kadoda, "Using simulation to evaluate prediction techniques," presented at International Software Metrics Symposium, London, England, 2001.

[20] I. Myrtveit and E. Stensrud, "A controlled experiment to assess the benefits of estimating with analogy and regression models," *IEEE Transactions on Software Engineering*, vol. 25, pp. 510-525, 1999.

[21] L. C. Briand, K. El Emam, D. Surmann, I. Wieczorek, and K. D. Maxwell, "An assessment and comparison of common software cost estimation modeling techniques," presented at The 21st International Conference on Software Engineering, Los Angeles, California, 1999.

[22] E. Mendes and B. Kitchenham, "Further Comparison of Cross-Company and Within-Company Effort Estimation Models for Web Applications," presented at Software Metrics, 10th International Symposium on (METRICS'04), Chicago, Illinois, 2004.

[23] G. R. Finnie, G. E. Wittig, and J. M. Desharnais, "Estimating software development effort with case-based reasoning," presented at ICCBR, Rhode Island, USA, 1997.

[24] G. R. Finnie, 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," *Journal of Systems and Software*, vol. 39, pp. 281-289, 1997.

[25] L. Angelis and I. Stamelos, "A simulation tool for efficient analogy based cost estimation," *Empirical Software Engineering*, vol. 5, pp. 35-68, 2000.

[26] E. Mendes and N. Mosley, "Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications," presented at 2002 International Symposium on Empirical Software Engineering (ISESE'02), Nara, Japan, 2002.

[27] D. R. Jeffery, M. Ruhe, and I. Wieczorek, "Using public domain metrics to estimate software development effort," presented at Seventh International Software Metrics Symposium., 2001.

[28] E. Mendes and S. Counsell, "Early Web Size Measures and Effort Prediction for Web Costimation," presented at Ninth International Software Metrics Symposium (METRICS'03), Sydney, Australia, 2003.

[29] F. Niessink and H. van Vliet, "Predicting Maintenance Effort with Function Points," presented at International Conference on Software Maintenance (ICSM '97), Bari, ITALY, 1997.

[30] B. Kitchenham, *Measurement for Software Process Improvement*. Oxford: Blackwell, 1996.

[31] R. T. Hughes, A. Cunliffe, and F. Young-Martos, "Evaluating software development effort model-building techniques for application in a real-time telecommunications environment," *IEE Proceedings Software*, vol. 145, pp. 29-33, 1998.

[32] C. Mair, G. Kadoda, M. Lefley, K. Phalp, C. Schofield, M. Shepperd, and S. Webster, "An investigation of machine learning based prediction systems," *The journal of Systems and Software*, vol. 53, pp. 23-29, 2000.

[33] R. Jeffery, M. Ruhe, and I. Wieczorek, "A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data," *Information and Software Technology*, vol. 42, pp. 1009-1016, 2000.

[34] L. C. Briand, T. Langley, and I. Wieczorek, "A replicated assessment and comparison of common software cost modeling techniques," presented at International Conference on Software Engineering, Limerick, Ireland, 2000.

[35] C. J. Burgess and M. Lefley, "Can genetic programming improve software effort estimation? A comparative evaluation," *Information and Software Technology*, vol. 43, pp. 863-873, 2001.

[36] M. Shepperd and G. Kadoda, "Comparing software prediction techniques using simulation," *IEEE Transactions on Software Engineering*, vol. 27, pp. 1014-1022, 2001.

[37] I. Wieczorek and M. Ruhe, "How valuable is company-specific data compared to multi-company data for software cost estimation?," in *Eighth IEEE Symposium on Software Metrics, Proceedings*, 2002, pp. 237-246.

[38] E. Mendes, I. Watson, C. Triggs, N. Mosley, and S. Counsell, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications," *Empirical Software Engineering*, vol. 8, pp. 163-196, 2003.

[39] E. Mendes, N. Mosley, and S. Counsell, "Early Web size measures and effort prediction for Web costimation," in *Ninth International Software Metrics Symposium, Proceedings*, 2003, pp. 18-29.

[40] M. J. Prietula, S. S. Vicinanza, and T. Mukhopadhyay, "Software-effort estimation with a case-based reasoner," *Journal of*

Experimental and theoretical artificial intelligence, vol. 8, pp. 341-363, 1996.

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

[42] D. J. Spiegelhalter, C. C. Taylor, and J. Campbell, *Machine learning, neural and statistical classification*. Upper Saddle River, NJ, USA: Ellis Horwood, 1995.