Using CBR to Estimate Development Effort for Web Hypermedia Applications

Ian Watson[1], Emilia Mendes[1], Chris Triggs[2], Nile Mosley[3] & Steve Counsell[3]

[1]Computer Science Dept. University of Auckland, New Zealand
[2]Statistics Dept. University of Auckland, New Zealand
[3]Computer Science Dept. Auckland University of Technology, New Zealand
[4]Computer Science Dept. Birkbeck College, University of London, UK
{ian | emilia}@cs.auckland.ac.nz

Abstract

Good estimates of development effort play an important role in the successful management of larger software development projects. This paper compares the prediction accuracy of three CBR techniques to estimate the effort to develop Web hypermedia applications. Most comparative studies have used one CBR technique. We believe this may bias the results, as there are several CBR techniques that may also be used for effort prediction. This paper shows that a weighted Euclidian similarity measure was the most accurate of the CBR techniques tested.

1 Introduction

Software practitioners recognise the importance of realistic estimates of effort to the successful management of software projects, the Web being no exception. Having realistic estimates at an early stage in a project's life cycle allow project managers and development organisations to manage resources effectively. Several techniques for cost and effort estimation have been proposed over the last 30 years, falling into three general categories, expert judgement, algorithmic models and machine learning [1]. Recently several comparisons have been made between the three categories of prediction techniques [2, 3 & 4]. However no convergence has been obtained to date.

Most comparisons in the literature measure the prediction accuracy of techniques using attributes (e.g. lines of code, function points) of conventional software. This paper looks at prediction accuracy based on attributes of Web hypermedia applications instead.

Our research focus is on proposing and comparing development effort prediction models for Web hypermedia applications [4]. Readers interested in effort estimation models for Web software applications are referred to [5 & 6]].

The metrics used in our study reflect current industrial practices for developing multimedia and Web hypermedia applications [7 & 8]. This paper compares the prediction accuracy of three CBR techniques to estimate the effort to develop Web hypermedia applications. As design decisions, when building CBR prediction systems, are influential upon the results [9], we wanted to reduce any bias that may hinder these results, before comparing them to other prediction models, the results of which are presented elsewhere. This objectives are reflected in the following research question: will different combinations of parameter categories for the CBR technique generate statistically significantly different prediction accuracy?

These issues are investigated using a dataset containing 37 Web hypermedia projects developed by postgraduate and MSc students studying a Hypermedia and Multimedia Systems course at the University of Auckland. Several confounding factors, such as Web authoring experience, tools used, structure of the application developed, were controlled, so increasing the validity of the obtained data. The remainder of the paper is organised as follows: Section 2 describes our research method. Section 3 presents the results for the comparison of CBR approaches and Section 4 presents our conclusions.

2 Research Method

2.1 Dataset

All analysis presented in this paper was based on a dataset containing information for 37 Web hypermedia applications developed by postgraduate students. The data set is described in detail in the companion paper Each Web hypermedia application provided 46 pieces of data [4], from which we identified 8 attributes, shown in Table 1, to characterise a Web hypermedia application and its development process. These attributes form a basis for our data analysis. Total effort is our dependent/response variable and the other 7 attributes are our independent/predictor variables. All attributes were measured on a ratio scale.

The criteria used to select the attributes was [7]: i) practical relevance for Web hypermedia developers; ii) metrics which are easy to learn and cheap to collect; iii) counting rules which were simple and consistent.

The original dataset of 37 observations had three outliers where total effort was unrealistic. Those outliers were removed from the dataset, leaving 34 observations. Total effort was calculated as:

$$Total\text{-}effort = \sum_{i=1}^{i=n} PAE + \sum_{j=0}^{j=m} MAE + \sum_{k=0}^{k=o} PRE \qquad (1)$$

where PAE is the page authoring effort, MAE the media authoring effort and PRE the program authoring effort [4]. A detailed description of threats and comments on the validity of the case study is presented in [4].

Table 1 - Size and Complexity Metrics

Metric	Description
Page Count (PaC)	Number of html or shtml files used in the application.
Media Count (MeC)	Number of media files used in the application.
Program Count (PRC)	Number of JavaScript files and Java applets used in the application.
Reused Media Count (RMC)	Number of reused/modified media files.
Reused Program Count (RPC)	Number of reused/modified programs.
Connectivity Density (COD)	Total number of internal links divided by *Page Count*.
Total Page Complexity (TPC)	Average number of different types of media per page.
Total Effort (TE)	Effort in person hours to design and author the application

2.2 Evaluation Criteria

The most common approaches to assessing the predictive power of effort prediction models are:

- The Magnitude of Relative Error (MRE) [10]
- The Mean Magnitude of Relative Error (MMRE) [11]
- The Median Magnitude of Relative Error (MdMRE) [12]
- The Prediction at level n (Pred(n)) [13]
- Boxplots of residuals [14]

The MRE is defined as:

$$MRE_i = \frac{\left|ActualEffort_i - PredictedEffort_i\right|}{ActualEffort_i} \qquad (2)$$

Where *i* represents each observation for which effort is predicted. The mean of all MREs is the MMRE, which is calculated as:

$$MMRE = \frac{1}{n}\sum_{i=1}^{i=n} \frac{\left|ActualEffort_i - PredictedEffort_i\right|}{ActualEffort_i} \qquad (3)$$

The mean takes into account the numerical value of every observation in the data distribution, and is sensitive to individual predictions with large MREs. An option to the mean is the median, which also represents a measure of central tendency, however it is less sensitive to extreme values. The median of MRE values for the number i of observations is called the MdMRE. Another indicator which is commonly used is the Prediction at level l, also known as Pred(l). It measures the percentage of estimates that are within l% of the actual values. Suggestions have been made [15] that l should be set at 25% and that a good prediction system should offer this accuracy level 75% of the time. In addition, other prediction accuracy indicators have been suggested as alternatives to the commonly used MMRE and Pred(n) [14]. One such indicator is to use boxplots of the residuals (actual-estimate) [16].

The statistical significance of all the results, except boxplots, was tested using the T-test for paired MREs and MMREs and the Wilcoxon Rank Sum Test or Mann-Whitney U Test for MdMREs. Both were generated using 1% and 5% levels of significance.

3 Comparing CBR Approaches

During the process of applying case-based reasoning users may need to choose five parameters, as follows:

1. Feature subset selection
2. Similarity measure
3. Scaling
4. Number of retrieved cases
5. Case adaptation

Each parameter in turn can be split into more detail, and incorporated or not for a given CBR tool. Based on that, the question asked here is: will different combinations of parameter categories for the CBR technique generate statistically significantly different prediction accuracy? In answer, we compared the prediction accuracy of several estimations generated using different categories for a given parameter. Estimations were generated using two CBR tools, namely ANGEL [17] and CBR-Works [18].

ANGEL was developed at Bournemouth University. An important feature is its ability to determine the optimum combination of attributes for retrieving analogies (cases). ANGEL compares similar projects by using the unweighted Euclidean distance using variables that have been standardised between 0 and 1 [17].

CBR-Works is a state-of-the-art commercial CBR environment [18]. It was a product of years of collaborative European research by the INRECA I & II projects [19]. It is available commercially from Empolis (www.tecinno.com). The tool provides a variety of retrieval algorithms (Euclidean, weighted Euclidean, Maximum Similarity,) as well as fine control over individual feature similarity metrics. In addition, it provides sophisticated support for symbolic features and taxonomies hierarchies as well as providing adaptation rules and formulae.

3.1 Feature subset selection

Feature subset selection involves determining the optimum subset of features that gives the most accurate estimation. ANGEL optionally offers this functionality by applying a brute force algorithm, searching for all possible feature subsets. CBR-Works does not provide similar functionality.

Table 2 - Comparing FSS to NFSS

	Used FSS			Did not use FSS		
	k=1	k=2	k=3	k=1	k=2	k=3
1.1.1.1.1 MMRE	0.09	0.11	0.12	0.15	0.15	0.15
MdMRE	0.08	0.09	0.10	0.12	0.11	0.13
Pred(25)	97	94	88	76	82	82

To investigate if the feature subset selection would help achieve better prediction accuracy, we used the ANGEL tool, and leave-one-out cross-validation. The results are summarised on Table 2 and a boxplot of the residuals is presented on Figure 1. On Table 2 Kn represents the number of retrieved cases (K1,K2,K3), FSS stands for "Feature Subset Selection" and NFSS for "No Feature Subset Selection". It was observed that the prediction accuracy for estimations based on FSS were more accurate than those based on all seven attributes. The boxplots of the residuals show that the best predictions were obtained using 1 retrieved case (K1) + FSS option, followed by two cases (K2) + FSS, and 3 cases (K3) + FSS. These results were also confirmed by the values for MMRE, MdMRE and Pred(25).

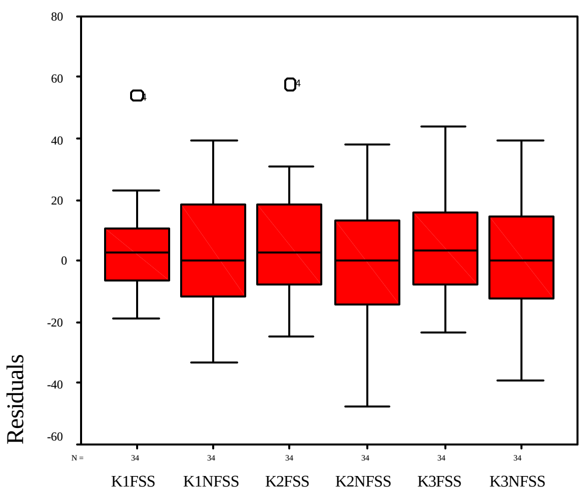

Figure 1 - Boxplots of the Residuals for FSS and NFSS

For k=1 case, the MRE for FSS was significantly less than that for NFSS (a=0.01), using a T-test. For k=2 and 3 cases the difference between FSS and NFSS was not statistically significant. Comparing these results to the boxplots of residuals suggests that for k=1 the feature subset selection may indeed affect the accuracy of the prediction obtained

3.2 Similarity Measure

To our knowledge, the similarity measure most frequently used in Software engineering and Web engineering literature, is the unweighted Euclidean distance. In the context of this investigation we have used three measures of similarity, namely the unweighted Euclidean distance, the weighted Euclidean distance and the Maximum measure.

3.3 Scaling or Standardisation

Standardisation represents the transformation of attribute values such that all attributes are measured using the same unit. One possible solution is to assign zero to the minimum observed value and one to the maximum observed value [9]. This is the strategy used by ANGEL and was the strategy chosen for part of the analysis carried out using CBR-Works.

3.4 Number of Retrieved Cases

The number of retrieved cases refers to the number of retrieved most similar cases that will be used to generate the estimation. For Angelis and Stamelos [20] when small sets of data are used it is reasonable to consider only a small number of cases. In this study we have used 1, 2 and 3 retrieved cases, similarly to [3, 17 & 20].

Dist.	K	Adpt.	SV?	MMRE	MdMRE	Pred(25)
UE	1	Mean	Yes	0.12	0.10	88.24
			No	0.11	0.09	91.18
	2	Mean	Yes	0.15	0.12	82.35
			No	0.13	0.11	88.24
		IRWM	Yes	0.13	0.11	85.29
			No	0.12	0.11	91.18
	3	Mean	Yes	0.14	0.11	82.35
			No	0.12	0.10	91.18
		IRWM	Yes	0.13	0.12	85.29
			No	0.11	0.08	91.18
		Median	Yes	0.14	0.10	76.47
			No	0.14	0.09	82.35
WE	1	Mean	Yes	0.10	0.09	94.12
			No	0.11	0.09	94.12
	2	Mean	Yes	0.13	0.11	94.12
			No	0.13	0.11	94.12
		IRWM	Yes	0.12	0.11	97.06
			No	0.11	0.11	97.06
	3	Mean	Yes	0.13	0.09	88.24
			No	0.12	0.09	88.24
		IRWM	Yes	0.12	0.12	94.12
			No	0.12	0.12	94.12
		Median	Yes	0.14	0.10	82.35
			No	0.13	0.10	82.35
MX	1	Mean	Yes	0.32	0.34	26.47
			No	0.32	0.33	26.47
	2	Mean	Yes	0.23	0.17	67.65
			No	0.23	0.17	67.65
		IRWM	Yes	0.25	0.23	58.82
			No	0.25	0.23	58.82
	3	Mean	Yes	0.25	0.15	76.47
			No	0.24	0.15	76.47
		IRWM	Yes	0.23	0.16	67.65
			No	0.23	0.16	67.65
		Median	Yes	0.31	0.17	58.82
			No	0.31	0.16	61.76
Dist. = distance UE = Unweighted Euclidean WE = Weighted Euclidean MX = Maximum				K = # of retrieved cases Adpt. = adaptation SV? = Standardised Variable?		

Table 3 - Comparison of CBR Techniques.

3.5 Case Adaptation

Once the most similar case(s) has/have been retrieved the next step is to decide how to generate the estimation. Choices of case adaptation techniques presented in the software engineering literature vary from the nearest neighbour [3], the mean of the closest cases [13], the median [20], inverse distance weighted mean and inverse rank weighted mean [9], to illustrate just a few. We opted for the mean (the average of k retrieved cases, when k>1), median (the median of k retrieved cases, when k>2) and the inverse rank weighted mean, which allows more similar cases to have more influence than less similar ones(e.g., if we use 3 cases, for example, the closest case would have weight = 3, the second closest weight = 2 and the last one weight =1).

3.6 Comparison of techniques

The first question we wanted to answer was if there were any statistically significant differences between results obtained using Standardised and Non-standardised variables. A T-test (for MMREs) and a Mann-Whitney U Test (for MdMREs), for a=0.01 and a=0.05 did not reveal any statistically significant differences.

The second question was if there were any statistically significant differences between results obtained using different distances (Unweighted Euclidean, Weighted Euclidean and Maximum). This time we restricted our analysis to results obtained using standardised variables. Both T-test (for MMREs and Pred(25)) and a Wilcoxon Signed Rank Test (for MdMREs), using a=0.01 and a=0.05 were performed (see Table 4).

Table 4 - Comparison of Distances

Distance	T-test	Wincoxon test
UE x WE	3.796 *	-1.633
UE x MX	-6.982 **	-2.207*
WE x MX	-7.652 **	-2.207*

UE = Unweighted Euclidea
WE = Weighted Euclidean
MX = Maximum
*** statistically significant at 1%*
** statistically significant at 5%*

Table 5 - Comparison of Euclidean Distances

	1 analogy	2 analogies	3 analogies
UE x WE	1.338	-2.400*	0.610

WE = Weighted Euclidean
UE = Unweighted Euclidean
** statistically significant at 5%*

It was no surprise to obtain statistically significant results when comparing the Maximum distance to any other type, as it gave much worse results than the other two. The Weighted Euclidean (WE) showed statistically significant better results (a=0.01) than the Unweighted Euclidean (UE), for MMREs (Table 4) and paired MREs (Table 5), however none when we used MdMREs. Boxplots of the residuals (Figure 2) corroborate the results obtained using the T-test. The answer to our question was therefore, positive: there are statistically significant differences between results obtained using different distances.

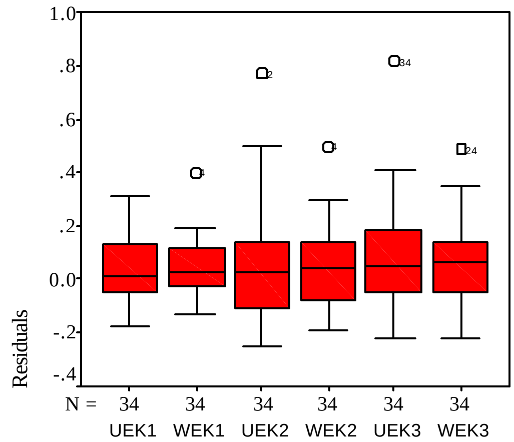

Figure 2 - Boxplots of the Residuals for Euclidean distances

Consequently, the answer to our general question: will different combinations of parameter categories for the CBR technique generate statistically significantly different prediction accuracy? was, at least for the dataset used, positive. Different combinations of parameter categories for the CBR technique gave statistically significantly different prediction accuracy.

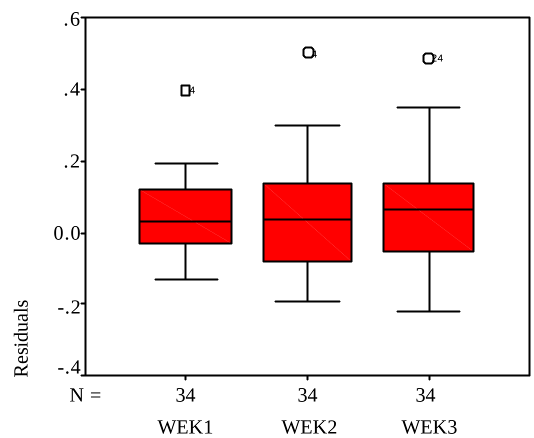

Figure 3 - Boxplots of the Residuals for Weighted Distances

Our next step was to choose the WE combination that gave the best prediction accuracy, and to assess whether different prediction accuracies would be statistically significant or not. To decide, we compared paired MREs for one, two and three retrieved cases using a T-test (Table 6). Boxplots for their residuals (Figure 3) confirmed the results obtained by the T-test, ie., one retrieved case (the most similar) gave the best results, which were statistically significantly better than those for two and three retrieved cases. Consequently, the technique, which gave the best prediction accuracy, used one retrieved case, based on a weighted Euclidian distance.

Table 6 - Comparison Weighted Euclidean Distances

k=1 vs k=2	k=1 vs. k=3	k=2 vs. k=3
-3.290**	-3.290**	0.294
** statistically significant at 1%		

4 Conclusions

In this study we investigated two questions related to effort prediction models for Web hypermedia applications, which were:

1. Will different combinations of parameter categories for the CBR technique generate statistically significantly different prediction accuracy?
2. Which of the techniques employed in this study gives the most accurate predictions for the dataset?

In addressing the first question, our results show that the CBR technique which gave the most accurate results used a Weighted Euclidean distance similarity measure to retrieve a single most similar case (k=1). We do accept that our results may obviously be dependent on the data set that we used and future work will seek to extend the data sets that we use.

5 References

[1] M.J. Shepperd, C. Schofield, and B. Kitchenham, "Effort Estimation Using Analogy." Proc. ICSE-18, IEEE Computer Society Press, Berlin, 1996.

[2] A.R. Gray, and S.G. MacDonell. A comparison of model building techniques to develop predictive equations for software metrics. Information and Software Technology, 39: 425-437, 1997.

[3] L.C. Briand, K.El-Emam, D. Surmann, I. Wieczorek, and K.D. Maxwell, An Assessment and Comparison of Common Cost Estimation Modeling Techniques, Proceedings of ICSE 1999, Los Angeles, USA, p:313-322, 1999.

[4] Mendes, E., Mosley, N., and Counsell, S. Web Metrics – Estimating Design and Authoring Effort. IEEE Multimedia, Special Issue on Web Engineering, January-March, 50-57, 2001.

[5] M. Morisio, I. Stamelos, V. Spahos and D. Romano, “Measuring Functionality and Productivity in Web-based applications: a Case Study”, Proceedings of the Sixth International Software Metrics Symposium, 1999, pp. 111-118.

[6] D.J. Reifer, Web Development: Estimating Quick-to-Market Software, IEEE Software, November/December 2000, p:57-64.

[7] Cowderoy, Measures of size and complexity for web-site content, Proceedings of the Combined 11th European Software Control and Metrics Conference and the 3rd SCOPE conference on Software Product Quality, Munich, Germany, p:423-431.

[8] A.J.C. Cowderoy, A.J.M. Donaldson, J.O. Jenkins, A Metrics framework for multimedia creation, Proceedings of the 5th IEEE International Software Metrics Symposium, Maryland, USA, 1998.

[9] G. Kadoda, M. Cartwright, L. Chen, and M.J. Shepperd, Experiences Using Case-Based Reasoning to Predict Software Project Effort, Proceedings of the EASE 2000 Conference, Keele, UK, 2000.

[10] C.F. Kemerer, An Empirical Validation of Software Cost Estimation Models, Communications of the ACM, v.30:5. P:416-429.

[11] M.J. Shepperd, C. Schofield, and B. Kitchenham, "Effort Estimation Using Analogy." Proc. ICSE-18, IEEE Computer Society Press, Berlin, 1996

[12] 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, No. 4, Jul./Aug. 1999, pp. 510-525.

[13] M.J. Shepperd, and C. Schofield, “Estimating Software Project Effort Using Analogies.” IEEE Transactions on Software Engineering, Vol. 23, No. 11, pp. 736-743, 1997.

[14] B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, M.J. Shepperd, " What accuracy statistics really measure", IEE Proceedings - Software Engineering June 2001, Vol. 148 Issue 3, p: 107.

[15] S. Conte, H. Dunsmore, and V. Shen, Software Engineering Metrics and Models. Benjamin/Cummings, Menlo Park, California, 1986.

[16] L.M. Pickard, B.A. Kitchenham, and S.J Linkman, An investigation of analysis techniques for software datasets, Proceedings of the 6th International Symposium on Software Metrics (Metrics99), IEEE Computer Society Press, Los Alamitos, California, 1999.

[17] Schofield, C. An empirical investigation into software estimation by analogy, PhD thesis, Dept. of Computing, Bournemouth Univ., UK, (1998).

[18] Schulz S. CBR-Works - A State-of-the-Art Shell for Case-Based Application Building, Proceedings of the German Workshop on Case-Based Reasoning, GWCBR'99 (1999). Lecture Notes in Artificial Intelligence Springer-Verlag 1995.

[19] Bergmann, R. Highlights of the INRECA Projects. In Case-Based Reasoning Research & Development. Aha, D. & Watson, I. (Eds.) pp. 1-15. Springer Lecture Notes in AI 2080. Berlin. 2001.

[20] L. Angelis, and I. Stamelos, A Simulation Tool for Efficient Analogy Based Cost Estimation, Empirical Software Engineering, 5, 35-68, 2000.