Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications

Emilia Mendes
Computer Science Department
The University of Auckland, NZ

Nile Mosley
MXM Technology
Auckland, NZ

Abstract

To date studies using CBR for Web hypermedia effort prediction have not applied adaptation rules to adjust effort according to a given criterion. In addition, when applying n-fold cross-validation, their analysis has been limited to a maximum of three training sets, which according to recent studies, may lead to untrustworthy results.

This paper has therefore two objectives. The first is to further investigate the use of CBR for Web hypermedia effort prediction by comparing the prediction accuracy of eight CBR techniques, of which three have previously been compared. The second objective is to compare the prediction accuracy of the best CBR technique against stepwise regression, using a twenty-fold cross-validation. All prediction accuracies were measured using Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error, Prediction at level l (l=25%), and boxplots of the residuals.

One dataset was used in the estimation process and, according to all measures of prediction accuracy, stepwise regression showed the best prediction accuracy.

1: Introduction

Recently, growth of the Web as a delivery environment gave birth to a new research field - Web engineering, to apply engineering principles to develop quality Web applications [1]. A variety of technological solutions are available for Web developers to facilitate the delivery of quality Web applications and to bring them to market as quickly as possible, with typical durations ranging from 3 to 6 months [2]. There are no standardised development techniques or large datasets of historical data on Web development projects. Therefore, given the Web's fluidic scope, development effort prediction for Web applications, although important, is a challenging task [1].

The Web engineering literature is sparse when comparing the prediction accuracy of different effort prediction approaches, with emphasis placed on Case-based Reasoning (CBR), linear and stepwise regressions [3,4]. Favourable results have been obtained for both CBR and regression techniques. In the studies that used CBR, no adaptation rules to adjust effort according to a given criterion have been applied. In addition, when using n-fold cross-validation, their analysis has been limited to a maximum of three training sets, which according to recent studies, may lead to untrustworthy results [5]. According to [5] ideally at least 20 sets should be deployed.

This paper looks further into effort prediction for Web hypermedia applications [3,4,6-8], where the size measures used reflect current industrial cost estimation practices for developing multimedia and Web hypermedia applications [9,10]. Our goal is to propose and compare effort prediction models based on measures relevant to those who develop Web sites structured according to the hypermedia paradigm.

We differentiate between Web hypermedia application and Web software application. The former is a non-conventional application characterised by the structuring of information using nodes (chunks of information), links (relations between nodes), anchors, access structures (for navigation) and the delivery of this structure over the Web. The latter represents any conventional software application that depends on the Web or uses the Web's infrastructure for execution. Typical applications include legacy information systems such as databases, booking systems, knowledge bases etc. Many e-commerce applications fall into the latter category.

This paper therefore has two objectives. The first is to investigate further the use of CBR for Web hypermedia effort prediction by comparing the prediction accuracy of eight CBR techniques. Three of these techniques have been previously compared [3], and this study extends it by adding another five. The second objective is to compare, using an n-fold cross-validation [11], (n=20), the prediction accuracy of the best CBR technique against stepwise regression. All prediction accuracies were measured using Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MdMRE), Prediction at level l (Pred(25)), (l=25%), and boxplots of the residuals.

Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02)
0-7695-1796-X/02 $17.00 © 2002 IEEE

Our objectives are reflected in the following research questions:

Question 1: Will any of the CBR techniques that use adaptation rules, adjusting effort according to a given criterion, present statistically significantly better prediction accuracy than their counterparts not using adaptation rules?

Question 2: Which of the CBR techniques employed in this study gives the most accurate predictions for the dataset?

Question 3: Which of the two effort prediction approaches employed in this study (CBR vs. Stepwise Regression) yields the most accurate predictions for the dataset?

These questions are investigated using a dataset containing 37 Web hypermedia projects developed by postgraduate students. Several confounding factors, such as Web authoring experience, tools used and structure of the application developed, were controlled, so increasing the validity of the obtained data. Details on this dataset and threats to their validity are given in [7].

The remainder of the paper is organised as follows: Section 2 provides a literature review and places this paper in the context of existing research in Web engineering. Section 3 describes our research method. Sections 4 and 5 present respectively the results for the comparison of CBR approaches, and the comparison of CBR to stepwise regression. Section 6 presents our conclusions and comments on future work.

Readers are also guided towards [12] for an overview of effort prediction in software engineering.

2: Related Work

To our knowledge, there are relatively few examples in the literature of studies that compare effort prediction models generated using data from Web hypermedia applications [3,4,6-8]. Most research in Web/hypermedia engineering has focused on the proposal of methods, methodologies and tools as a basis for process improvement and higher product quality [13-16].

Mendes et al. [4] describes a case study evaluation involving the development of 76 Web hypermedia applications structured according to the Cognitive Flexibility Theory (CFT) [17] principles in which length and complexity metrics were collected. Several prediction models were generated (multiple linear regression, stepwise regression and case-based reasoning) for each of the four datasets employed and their predictive power was compared using the MMRE and MdMRE measures. Results showed that the best predictions were obtained using CBR. This study is limited in that it uses only one CBR technique, with no further validation of results by applying n-fold cross validation, coupled with the use of only two measures of accuracy (MMRE and MdMRE).

Mendes et al. [3] compares three different CBR techniques using two datasets of Web hypermedia projects. Best predictions for both datasets, measured using three measures of prediction accuracy (MMRE, MdMRE and Pred(25)), were obtained using the weighted Euclidean distance. Although their results converged, the limitation is this study does not validate their results by applying an n-fold cross validation and does not show the statistical significance of their findings.

Mendes et al. [7] describes a case study evaluation in which 37 Web hypermedia applications were used. These were structured according to the CFT principles. The measures collected were organised into five categories: length size, complexity size, reusability, effort and confounding factors. The size and reusability metrics were used to generate top down and bottom up prediction models using linear and stepwise regression techniques. They compared the predictive power of the regression models using the MMRE measure. Stepwise regression was not shown to be consistently better than multiple linear regression. A limitation of this study is that it only compared prediction models generated using algorithmic techniques and measured prediction accuracy using only MMRE.

Mendes et al. [8] presents a case study where size attributes of Web hypermedia applications were measured. Those attributes correspond to three size categories, namely Length, Complexity and Functionality. For each size category they generated prediction models using linear and stepwise regression. The accuracy of these predictions was compared using boxplots of the residuals. Results suggested that all models offered similar prediction accuracy. The limitation of this study is also that it compared prediction models generated using only algorithmic techniques and measured their prediction accuracy using only boxplots of the residuals.

Mendes et al. [3] compares the prediction accuracy of three CBR techniques to estimate effort for developing Web hypermedia applications. They also compare the best CBR technique against three commonly used prediction models, namely multiple linear regression, stepwise regression and regression trees. Prediction accuracy is measured using MMRE, MdMRE, Pred(25) and boxplots of the residuals and information on the statistical significance of their results is also given. Their findings suggest that both Multiple regression models and CBR presented the best prediction accuracy, depending on how prediction accuracy was measured: MMRE and MdMRE showed better prediction accuracy for Multiple regression models whereas boxplots showed better accuracy for CBR. The limitations of their work are that they did not use adaptation rules when applying CBR techniques and their n-fold cross-validation was restricted to three sets.

The study described in this paper, in addition to comparing eight CBR techniques for estimating Web hypermedia development effort, compares the best CBR

technique to Stepwise Regression by applying a 20-fold cross-validation.

3: Research Method

3.1: Dataset Description

The analysis presented in this paper was based on a dataset containing information from 37 Web hypermedia applications developed by postgraduate students.

Each Web hypermedia application provided 46 variables [3], from which we identified 8 (see Table 1), to characterise a Web hypermedia application and its development process. These form a basis for our data analysis. Total effort is our dependent/response variable and the remaining 7 are our independent/predictor variables. All variables were measured on a ratio scale.

The criteria used to select the attributes was [9]: 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.

Measure	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[1] 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

Table 1 - Size and Complexity Metrics

Table 2 outlines the properties of the dataset used. The original dataset of 37 observations had three outliers where total effort was unrealistic compared to duration. Those outliers were removed, leaving 34 observations. Collinearity represents the number of statistically significant correlations with other independent variables out of the total number of independent variables [18].

Cases	Features	Cat. features	outliers	collinearity
34	8	0	0	2/7

Table 2 - Properties of the dataset

[1] Subjects did not use external links to other Web hypermedia applications. All the links pointed to pages within the original application only.

Summary statistics for all the attributes are presented on table 3.

Attr.	Mean	Med.	Min	Max	Std. Dev.
PaC	55.2	53	33	100	11.2
MeC	24.8	53	0	126	29.2
PRC	0.4	0	0	5	1.0
RMC	42	42.5	0	112	31.6
RPC	0.2	0	0	8	1.3
COD	10.4	9.01	1.6	23.3	6.1
TPC	1.2	1	0	2.51	0.5
TE	111.9	114.6	58.3	153.7	26.4
Obs.	34	34	34	34	34

Table 3 - Summary statistics for all attributes

Excluding total effort, all measures collected were checked against the original Web hypermedia applications to ensure that attributes were precisely measured. More details on this dataset and a detailed description of threats and comments on the validity of the case study can be found at [7].

3.2: Effort Prediction Approaches Employed

Four types of effort prediction approaches have been compared in the Web engineering literature [3,4,6-8], namely multiple linear regression, stepwise regression, regression tree-based models (CART) and CBR.

For the scope of this paper we selected a subset based on similar criteria to that used in [11]:

- Can the approach be automated?
- Has the approach been used previously in Web engineering?
- Are the results easy to understand from a practitioner's point of view?

Similar to [11], we compute cross-validation mechanisms to calculate the accuracy values, opting for an automated mechanism.

We chose effort prediction approaches previously used in Web engineering to allow for the opportunity to compare results and look for convergence.

Finally, if effort prediction approaches are to be used by practitioners they should be easy to understand, so encouraging their use.

Based on the criteria aforementioned we chose the following approaches:

- Stepwise Regression
- CBR

We chose not to use Regression tree-based models as previous work [3] showed that this technique gave the worst results. In addition, both multiple linear regression and stepwise regression presented the same equations and adjusted R-squared values, so we opted to use stepwise regression only.

Stepwise Regression

Stepwise regression [19] builds a prediction model by, at each stage, adding to the model the variable that has the highest partial correlation with the response variable, taking into account all variables currently in the model. Its aim is to find the set of predictors that maximise *F*. *F* assesses whether the regressors, taken together, are significantly associated with the response variable. The criteria used to add a variable is whether it increases the *F* value for the regression by some specified amount *k*. When a variable reduces *F*, also by some specified amount *w,* it is removed from the model.

Stepwise regression has been frequently used as a benchmark in Software engineering and Web engineering [3,7,18,20,21] and is regarded by some as a good prediction technique [22].

Case-based Reasoning

CBR [23] provides estimates by comparing the current problem to be estimated against a library of historical projects. The similarity of features in the current problem description are compared to those in completed projects. Typically the development effort from the most similar completed project is retrieved and an estimate is calculated. Numerous techniques can be used for the similarity assessment, but in recent years, nearest neighbour algorithms [24] using a weighted Euclidean distance metric have been the most widely used both in software engineering and Web engineering.

3.3: Criteria Used to Evaluate Prediction Accuracy

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

- The Magniture of Relative Error (MRE) [25]
- The Mean Magnitude of Relative Error (MMRE) [20]
- The Median Magnitude of Relative Error (MdMRE) [26]
- The Prediction at level n (Pred(n)) [26]
- Boxplots of residuals [27]

The MRE is defined as:

$$MRE_i = \frac{\left| ActualEffort_i - \Pr edictedEffort_i \right|}{ActualEffort_i} \quad (1)$$

Where *i* represents each observation for which effort is predicted.

The mean of all MREs is the MMRE, which is calculated as:

$$MMRE = \frac{1}{n}\sum_{i=1}^{i=n}\frac{\left| ActualEffort_i - \Pr edictedEffort_i \right|}{ActualEffort_i} \quad (2)$$

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 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 [28] that *l* should be set at 25% and 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 MMRE and Pred(n) [27]. One such indicator is to use boxplots of the residuals (actual-estimate) [27].

The statistical significance of all the results, except boxplots, was tested using the T-test for paired MREs, generated using 1% and 5% levels of significance.

4: Comparing CBR techniques

4.1: Parameters to consider when comparing CBR techniques

The six parameters we considered to compare eight CBR techniques were as follows:

- Feature subset selection
- Similarity measure
- Scaling
- Number of analogies
- Analogy adaptation
- Adaptation rules to adjust the results

Feature subset selection

Feature subset selection involves determining the optimum subset of features that gives the most accurate estimation. CBR tools, such as ANGEL [20], offer this functionality by applying a brute force algorithm, searching for all possible feature subsets.

All our CBR results were obtained using CBR-Works [31], which does not offer the feature subset selection option.

Similarity Measure

Similarity Measure, as the name indicates, measures the level of similarity between cases. 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, weighted Euclidean distance and Maximum measure.

Unweighted Euclidean distance:

The Euclidean distance *d* between the points (x_0,y_0) and (x_1,y_1) is given by the formula:

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad (3)$$

Weighted Euclidean distance:

It is common in CBR for the features vectors to be weighted to reflect the relative importance of each feature. The weighted Euclidean distance *d* between the points (x_0,y_0) and (x_1,y_1) is given by the formula:

$$d = \sqrt{w_x(x_0 - x_1)^2 + w_y(y_0 - y_1)^2} \quad (4)$$

where w_x and w_y are the weights of *x* and *y* respectively.

Weight was calculated using two separate approaches:

1. We attributed weight=2 to attributes PaC, MeC and RMC and weight =1 to the remaining 4 attributes. Those attributes were chosen as they presented statistically significant correlation (α=0.01) with Total effort. In so doing we hoped to simulate the "Feature subset selection" option, provided by the ANGEL tool.
2. We measured the linear association between the predictors and response variables using a one-tailed Pearson's correlation, using coefficient values as weights.

Maximum measure:

Using the maximum measure, the maximum feature similarity defines the case similarity. For two points (x_0,y_0) and (x_1,y_1), the maximum measure *d* is equivalent to the formula:

$$d = \sqrt{\max((x_0 - x_1)^2, (y_0 - y_1)^2)} \quad (5)$$

This effectively reduces the similarity measure down to a single feature, although the maximum feature may differ for each retrieval episode.

Scaling or Standardisation

Standardisation represents the normalisation of attribute values according to a defined rule 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 [20]. This is the strategy used by ANGEL.

In this study we normalised all variables in the dataset to be between 0 and 1, by dividing every variable value by its maximum observed value.

Number of Analogies

The number of analogies refers to the number of most similar cases that will be used to generate the estimation. For Angelis and Stamelos [29] when small datasets are used it is reasonable to consider only a limited number of analogies.

In this study we have used 1, 2 and 3 analogies, similarly to other studies presented in Web engineering [3,4,6].

Analogy Adaptation

Once the most similar case(s) has/have been selected the next step is to decide how to generate the estimation. Choices of analogy adaptation techniques presented in the Software engineering literature vary from the nearest neighbour [11], the mean of the closest analogies [26], the median [29], inverse distance weighted mean and inverse rank weighted mean [30], to illustrate just a few. In the Web engineering literature, adaptations mostly used are the nearest neighbour and the mean of the closest analogies [3,4,6]. To our knowledge only one study has employed, in addition to the nearest neighbour and mean of the closest analogies, the median and the inverse rank weighted mean [3].

We opted for the mean, median and the inverse rank weighted mean.

Mean: Represents the average of k analogies, when k>1.

Inverse rank weighted mean: Allows higher ranked analogies to have more influence than lower ones. E.g., using 3 analogies, the closest analogy (CA) would have weight = 3, the second closest (SC) weight = 2 and the last one (LA) weight =1. The estimation would then be calculated as:

$$(3*CA + 2*SC + LA)/6 \quad (6)$$

Median: Represents the median of k analogies, when k>2.

Adaptation rules to adjust the results

Adaptation rules are used to adapt the estimated result, according to a given criterion, such that it reflects the case characteristics more closely. For example, in the context of effort prediction, the estimated effort to develop an application *app* would be adapted such that it would also take into consideration an *app*'s size values.

The adaptation rule employed was based on the linear size adjustment to the estimated effort [32], where we adapted the differences in size between target (estimated) and source (most similar) applications as:

$$Effort_{est} = \frac{1}{p}\sum_{p=1}^{p=7}\left(\frac{1}{n}\sum_{n=1}^{n=3}\frac{S_{est}.Effort_n}{S_n}\right) \quad (7)$$

where p is the number of size measures, n is the number of analogies, $Effort_{est}$ is the effort we wish to estimate, S_{est} is the size measure for the project which effort we wish to estimate, $Effort_n$ is the effort for the project corresponding to analogy n and S_n is the size measure for the project corresponding to analogy n.

4.2: Comparison of CBR Techniques

The results in Table 4 were obtained considering four similarity measures (unweighted Euclidean, weighted Euclidean using subjective weights, weighted Euclidean using Pearson's correlation coefficient weights and Maximum), three choices for the number of analogies (1, 2 and 3), three choices for the analogy adaptation (mean, inverse rank weighted mean and median) and two alternatives regarding the use or not of adaptation rules. Results obtained using Adaptation rules are identified as CBRAR. Not using adaptation rules are identified as CBRNAR.

Figures 1 to 4 present boxplots of residuals organised per similarity measure, comparing CBRAR to CBRNAR.

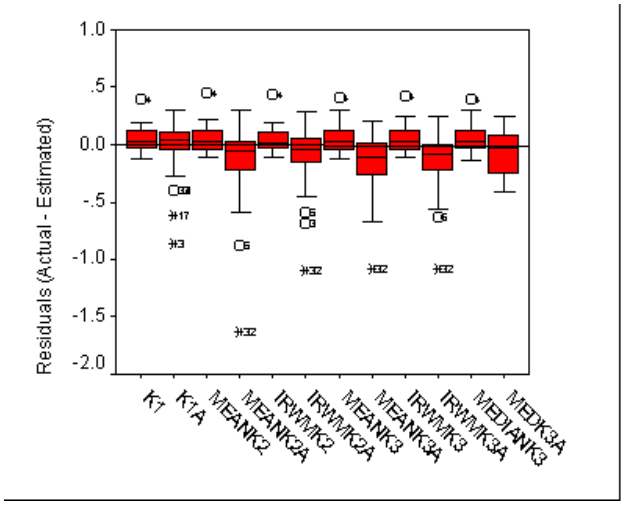

Figure 1 - Boxplots of residuals for Unweighted Euclidean

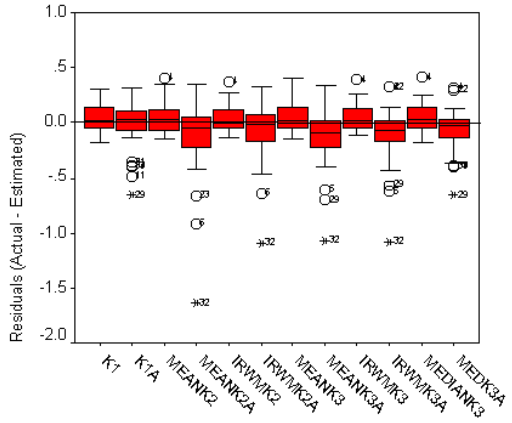

Figure 2 - Boxplots of residuals for Weighted Euclidean based on subjective weights

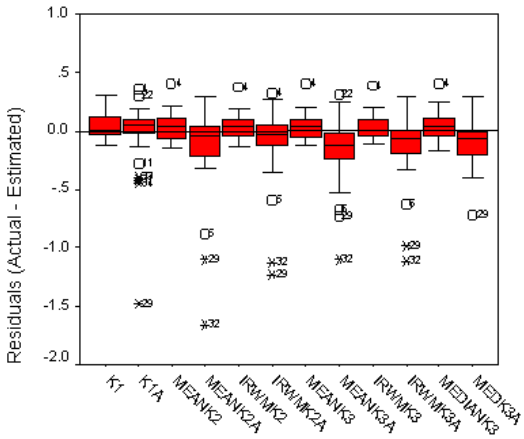

Figure 3 - Boxplots of residuals for Weighted Euclidean based on Pearson Correlation Coefficient weights

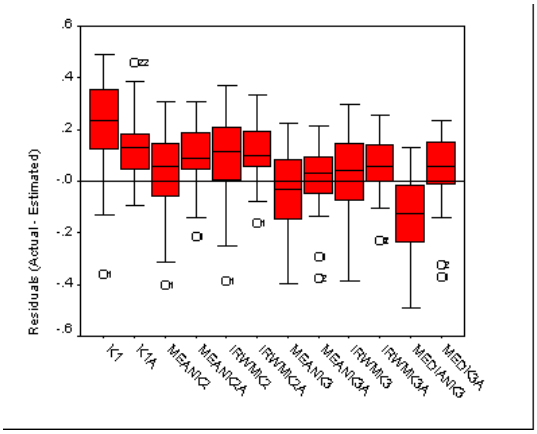

Figure 4 - Boxplots of residuals for Maximum Distance

For CBRAR (see Table 4), the maximum distance for 3 analogies gave the best MMRE, MdMRE and Pred(25).

Based on the threshold values for good prediction systems suggested by Conte et al. [28], CBRNAR predictions were better, for both MMRE and Pred(25), than those for CBRAR.

Our comments based on all four boxplots of residuals are as follows:

Boxplots of residuals for figure 1 suggest that K1 (one analogy without adaptation rules) gave the best results without adaptation (confirmed by corresponding MMRE, MdMRE and Pred(25)), and K1A (one analogy using adaptation rules) gave the best results for adaptation (confirmed by corresponding Pred(25)). In general, using the adaptation rule did not improve any of the results for UE.

For figure 2, boxplots of the residuals suggest that IRWMK2 (Inverse Rank Weighted Mean for 2 analogies without adaptation rules) gave slightly better predictions than K1 (one analogy without adaptation rules) (confirmed by corresponding Pred(25)). K1 presented better MMRE and MdMRE. Boxplots also suggest that K1A (one analogy using adaptation rules) gave the best predictions, corroborated by corresponding MMRE, MdMRE and Pred(25).

Boxplots of residuals for figure 3 suggest that IRWMK3 (Inverse Rank Weighted Mean for 3 analogies without adaptation rules) presented the best predictions, corroborated by corresponding Pred(25), however not corroborated by equivalent MMRE and MdMRE. Based on MMRE and MdMRE, K1 (one analogy without adaptation rules) gives the best prediction accuracy. These boxplots corroborate the results obtained for K1A (one analogy using adaptation rules) based on MMRE, MdMRE and Pred(25).

Boxplots of the residuals for figure 4 suggest that the best predictions were obtained for MEANK3 (mean of 3 closest analogies without adaptation rules), corroborated by corresponding Pred(25). For results generated using adaptation rules, the best predictions were obtained using MEANK3A (mean of 3 closest analogies), corroborated by corresponding MMRE, MdMRE and Pred(25).

The best results suggested by boxplots of residuals were also corroborated by their corresponding Pred(25), however these differed for MMRE and MdMRE.

To answer our first question we compared results obtained for CBRNAR and CBRAR, for each one of the four distances, using a T-test of paired MREs (see Tables 5 and 6) with α=0.01 and α=0.05.

Results without adaptation rules						Results with Adaptation Rules		
Distance	**K**	**Adaptation**	**MMRE**	**MdMRE**	**Pred(25)**	**MMRE**	**MdMRE**	**Pred(25)**
UE	1	CA	12	10	88.24	23	15	73.53
	2	Mean	15	12	82.35	32	14	67.65
		IRWM	13	11	85.29	28	13	70.59
	3	Mean	14	11	82.35	30	19	67.65
		IRWM	13	12	85.29	28	13	67.65
		Median	14	10	76.47	21	12	76.47
WESub	1	CA	10	09	94.12	21	12	76.47
	2	Mean	13	11	94.12	32	23	58.82
		IRWM	12	11	97.06	26	13	64.71
	3	Mean	13	09	88.24	31	23	55.88
		IRWM	12	12	94.12	27	19	64.71
		Median	14	10	82.35	23	18	61.76
WECorr	1	CA	11	09	88.24	24	09	73.53
	2	Mean	14	13	91.18	33	16	61.76
		IRWM	12	10	94.12	28	13	67.65
	3	Mean	13	11	94.12	32	19	58.82
		IRWM	12	10	97.06	28	14	64.71
		Median	15	10	79.41	24	16	64.71
MX	1	CA	32	34	26.47	20	18	67.65
	2	Mean	23	17	67.65	17	16	73.53
		IRWM	25	23	58.82	18	17	70.59
	3	Mean	25	15	76.47	14	10	85.29
		IRWM	23	16	67.65	15	12	85.29
		Median	31	17	58.82	16	12	79.41

UE – Unweighted Euclidean WESub - Weighted Euclidean based on subjective weights
WECorr – Weighted Euclidean using as weights Pearson's correlation Coefficients
MX – Maximum K - number of analogies CA – Closest Analogy IRWM – Inverted Rank Weighted Mean

Table 4 – Results for CBR Techniques with and without Adaptation Rules

	Unweighted Euclidean Distance			Weighted Euclidean Distance sub. weights		
Pairs	**Distances**	**t**	**Sig. (2-tailed)**	**Distances**	**t**	**Sig. (2-tailed)**
Pair 1	UEK1 - UEK1A	-2.465	.019*	WEK1 - WEK1A	-2.377	.023*
Pair 2	UEK2 - UEK2A	-2.052	.048*	WEK2 - WEK2A	-2.579	.015*
Pair 3	UEK2IRWM - UEK2IRA	-2.436	.020*	WEK2IRWM - WEK2IRA	-2.667	.012*
Pair 4	UEK3 - UEK3A	-2.616	.013*	WEK3 - WEK3A	-3.252	.003**
Pair 5	UEK3MD - UEK3MDA	-1.443	.158	WEK3MD - WEK3MDA	-2.537	.016*
Pair 6	UEK3IR - UEK3IRA	-2.403	.022*	WEK3IR - WEK3IRA	-2.908	.006**
** statistically significant at 1% * statistically significant at 5%						

Table 5 – Comparing the statistical significance for Unweighted Euclidean Distance and Weighted Euclidean Distance using subjective weights

	Weighetd Euclidean Distance pearson w.			Maximum Distance		
Pairs	**Distances**	**t**	**Sig. (2-tailed)**	**Distances**	**t**	**Sig. (2-tailed)**
Pair 1	WEK1 - WEK1A	-1.753	.089	MXK1 - MXK1A	3.231	.003**
Pair 2	WEK2 - WEK2A	-2.147	.039*	MXK2 - MXK2A	1.515	.139
Pair 3	WEK2IRWM - WEK2IRA	-2.130	.041*	MXK2IRWM - MXK2IRA	1.997	.054
Pair 4	WEK3 - WEK3A	-3.105	.004**	MXK3 - MXK3A	2.361	.024*
Pair 5	WEK3MD - WEK3MDA	-2.311	.027*	MXK3MD - MXK3MDA	2.538	.016*
Pair 6	WEK3IR - WEK3IRA	-2.452	.020*	MXK3IR - MXK3IRA	2.186	.036*
** statistically significant at 1% * statistically significant at 5%						

Table 6 – Comparison of statistical significance for Weighted Euclidean Distance using Pearson's correlation coefficients and for Maximum Distance

Unweighted Euclidean (see Table 5): T-test of paired MREs showed statistically significant better results for nearly all the CBRNAR results. These results were also corroborated by corresponding MMREs, MdMREs and Pred(25).

Weighted Euclidean using subjective weights (see Table 5): T-test of paired MREs showed that all the CBRNAR results were statistically significantly better than for CBRAR. These tests of significance were corroborated by corresponding MMREs, MdMREs and Pred(25).

Weighted Euclidean using Pearson's correlation coefficient weights (see Table 6): T-test of paired MREs showed that nearly all the CBRNAR results statistically significantly better than the CBRAR. These results were also confirmed by corresponding MMREs, MdMREs and Pred(25).

Maximum distance (see Table 6): T-test of paired MREs showed statistically significant better results for 1 and 3 analogies using adaptation rules. These results were also confirmed by corresponding MMREs, MdMREs and Pred(25). The Maximum distance simulates the situation in which only one size measure is used (the one with the highest similarity), although the size measure may differ for each retrieval episode. This may explain why results were better when applying the adaptation rule.

The inter-group comparison (CBRNAR vs. CBRAR) revealed that, except for maximum distance, the best predictions were obtained without applying the adaptation rule.

The answer to our first question was therefore, positive: One of the CBR techniques that used adaptation rules presented statistically significantly better prediction accuracy than their counterparts that did not use adaptation rules.

To answer our second question, we carried two intra-group comparisons, one for the CBRNAR group and another for the CBRAR group, also using T-tests for paired MREs and boxplots of the residuals. Results were omitted due to lack of space. Once the best CBR technique per group was obtained, both were compared in order to obtain the best CBR technique overall.

Maximum distance gave statistically significant worse accuracy, measured using MREs, than all other distances. All other distances seemed to present similar prediction accuracy.

Using boxplots of residuals, the Weighted Euclidean gave the best prediction 5 out of 6 times: 3 times based on subjective weights and twice based on Pearson's correlation coefficient weights. Unweighted Euclidean gave the best predictions 1 out of 6 times and maximum distance was the worst case on all 6 clusters.

Previous to using the Weighted Euclidean distance with weights based on Pearson correlation coefficients, the best result for CBRNAR was 1 analogy using the Weighted Euclidean distance with subjective weights (WEK1; MMRE=10; MdMRE=9; Pred(25)=94.12) [3]. A closer look at the boxplots of residuals (see Figure 5) revealed a subtle difference between WEK1 (weighted Euclidean distance using one analogy and subjective weights) and WEPIRK3 (weighted Euclidean using the Inverse rank weighted mean for 3 analogies and Pearson's correlation coefficient weights), where WEPIRK3 presented slightly more accurate predictions if based on boxplots of residuals and Pred(25) (MMRE=12; MdMRE=10; Pred(25)=97.06).

WEIRWM2 (weighted Euclidean using the Inverse rank weighted mean for 3 analogies and subjective weights) also presented good prediction accuracy (MMRE=12; MdMRE=11; Pred(25)=97.06), and even better than WEK1 if based solely on boxplots of residuals and Pred(25). We therefore chose WEPIRK3 as the best prediction for the CBRNAR group.

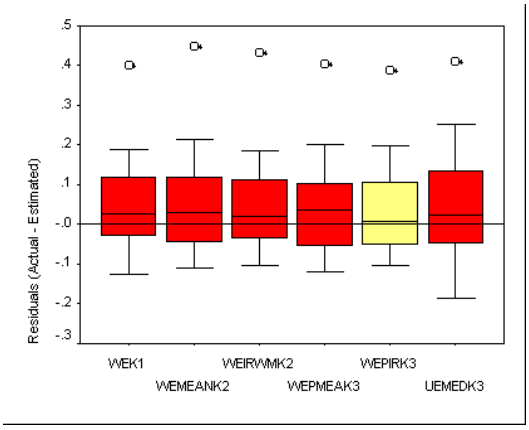

Figure 5 – Best predictions of each of the six clusters for CBRNAR.

Both maximum distance using the mean of the 3 closest analogies (MXK3A) and the inverse rank weighted mean of the closest 3 analogies (MXK3IRA) showed statistically significantly better accuracy, measured using MREs, than all other distances. These results are also corroborated by their corresponding MMREs, MdMREs and Pred(25).

According to the boxplots, for one analogy, WEPK1A (weighted Euclidean distance for one analogy using Pearson's correlation coefficient weights) gave the most accurate predictions (MMRE=24; MdMRE=9; Pred(25)=73.53). However, WEK1A (weighted Euclidean distance for one analogy with subjective weights) showed better prediction accuracy when measured using MMRE (21) and Pred(25) (76.47). For 2 and 3 analogies, the maximum distance gave the best results, also confirmed by their corresponding MMREs, MdMREs and Pred(25).

A closer look at boxplots of residuals, using the best result for each cluster (see figure 6) suggests that MXMEAK3A (maximum distance using the mean of the closest 3 analogies) gave the most accurate predictions. This results was also confirmed by the T-test of paired MREs, and corresponding MMRE, MdMRE and Pred(25).

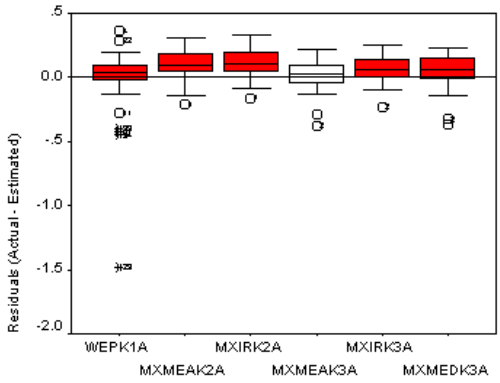

Figure 6 – Best predictions of each of the six clusters

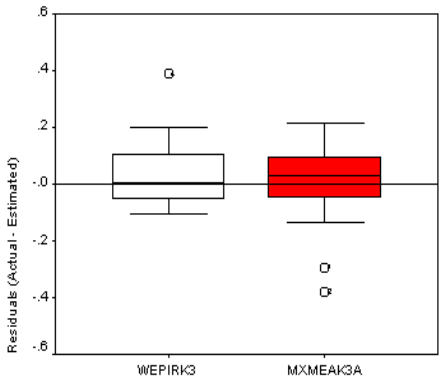

Figure 7 – Boxplots of residuals for best CBRNAR and CBRAR.

Finally, comparing the best CBRNAR technique to the best CBRAR technique gave the following result (see Table 7 and Figure 7):

Distances	t	Sig. (2-tailed)
WEPIRK3 - MXMEAK3A	1.378	.177

Table 7 – T-test of paired MREs for comparing the best CBRNAR to the best CBRAR

The answer to our second question was therefore: WEPIRK3. The CBR technique which presented the best prediction accuracy overall used the weighted Euclidean distance, the inverse rank weighted mean for three analogies and weights based on Pearson's correlation coefficients.

5: Comparing CBR to Stepwise Regression

To answer our third question, we measured the prediction accuracy of estimations
generated using the best CBR technique and Stepwise Regression.

To generate the estimations, we used a twenty-fold cross-validation approach [11]. Cross-validation involves dividing the whole dataset into multiple train and validation sets, calculating the accuracy for each validation set, and aggregating the accuracy across all validation sets. A twenty-fold cross-validation yields twenty different training-validation set combinations. We used a 66% split (23 observations in the train set and 11 in validation set). We therefore had in total twenty different combinations for each technique employed. All training sets were generated randomly. The prediction accuracies obtained are presented in Table 8, showing that all predictions were very good for CBR and excellent for Stepwise Regression. For CBR, all MMREs were below the 25% threshold and, except for v03 and v07, all Pred(25) were above the 75% threshold. In addition, boxplots of the residuals (see Figure 8) revealed that most predictions were not below or above 20% of their actuals, and few boxplots revealed residuals which were ± 10% from actual effort. For Stepwise Regression, all MMREs were far below the 25% threshold and boxplots of the residuals (see Figure 9) revealed that all residuals were ± 10% from actual effort.

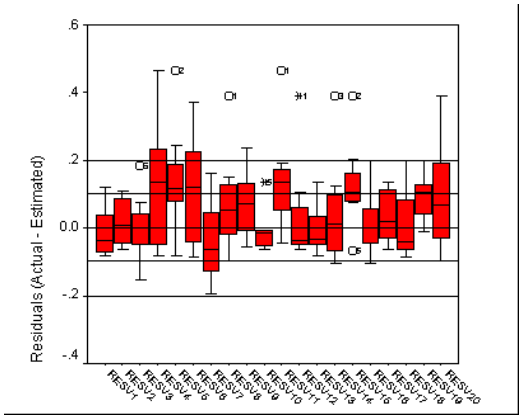

Figure 8 – Boxplots of residuals for twenty-fold cross-validation for the best CBR.

The final linear models for the stepwise regression were omitted for lack of space.

The results for the T-test of paired MREs per split version (see Table 9) show that Stepwise regression clearly gave statistically significantly better predictions than the best CBR. These results were of not surprise since all regression models presented very high adjusted R-squared.

	Results of the best CBR Technique			Results for Stepwise Regression		
Split	**MMRE (%)**	**MdMRE (%)**	**Pred(25) (%)**	**MMRE (%)**	**MdMRE (%)**	**Pred(25) (%)**
V01	12	11	87.50	1.56	0.96	100
V02	12	11	90.91	2.92	1.45	100
V03	15	09	66.67	2.98	2.61	100
V04	19	16	77.78	2.87	1.50	100
V05	18	13	88.89	1.69	0.98	100
V06	19	18	75.00	1.81	0.62	100
V07	22	16	60.00	3.85	3.65	100
V08	15	13	88.89	2.87	2.06	100
V09	12	12	90.00	3.43	1.70	100
V10	09	07	100.00	2.74	2.66	100
V11	16	14	90.00	3.26	1.24	100
V12	11	07	90.00	2.07	2.10	100
V13	11	10	100.00	2.59	2.29	100
V14	13	10	88.89	1.50	1.08	100
V15	16	12	80.00	2.34	1.48	100
V16	10	11	100.00	1.85	1.42	100
V17	10	08	100.00	1.95	1.24	100
V18	13	11	100.00	2.28	1.67	100
V19	13	15	100.00	2.42	0.99	100
V20	16	14	87.50	2.47	2.86	100

Table 8 – Prediction Accuracy for the twenty CBR and Stepwise Regression splits

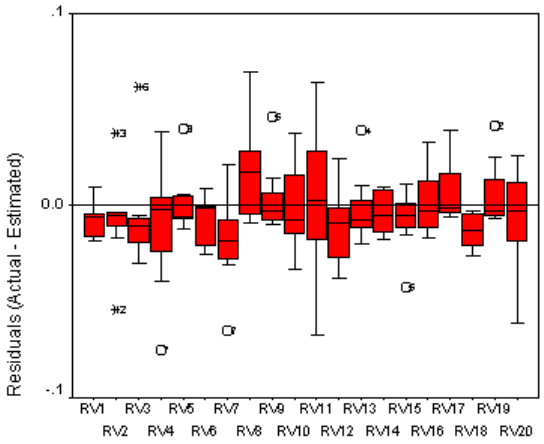

Figure 9 – Boxplots of the residuals for the Twenty-fold cross-validation for Stepwise Regression

CBR vs. SLR	t	Sig.	CBR vs. SLR	t	Sig.
V01	3.769	0.007**	V11	2.869	0.018*
V02	5.044	0.001**	V12	2.258	0.050*
V03	2.540	0.035*	V13	4.192	0.003**
V04	3.132	0.014*	V14	2.835	0.022*
V05	3.334	0.010*	V15	3.794	0.004**
V06	3.627	0.008**	V16	3.366	0.008**
V07	3.662	0.005**	V17	3.811	0.005**
V08	2.854	0.021*	V18	4.191	0.006**
V09	3.056	0.014*	V19	3.863	0.006**
V10	2.548	0.034*	V20	2.968	0.021*
** significant at 1%			* significant at 5%		

Table 9 – T-test comparing best CBR to Stepwise Regression

The answer to our third question was: Stepwise Linear Regression gave the most accurate predictions for our dataset.

6: Conclusions and Future Work

In this study we investigated three questions related to effort prediction models for Web hypermedia applications.

In addressing the first question, our results show that one CBR technique that used adaptation rules presented statistically significantly better prediction accuracy than its counterpart not using adaptation rules. The technique used the maximum distance, and used the mean of the closest 3 analogies.

In addressing the second question, our results show that the CBR technique that gave the best predictions used the weighted Euclidean distance, the inverse rank weighted mean for three analogies and weights based on Pearson's correlation coefficients.

Finally, in addressing the third question, the technique that gave the best prediction accuracy was Stepwise Regression, for all measures of prediction accuracy.

We have replicated part of this study using another dataset of Web hypermedia projects, addressing CBR-based effort predictions [6]. However in the future we also aim to answer the following questions [3]:

- What are the typical characteristics that may be found in a Web hypermedia project dataset?
- To what extent those datasets show similar characteristics to Web software project datasets and conventional software project datasets?

7: References

[1] Pressman, R.S. What a Tangled Web We Weave. *IEEE Software*, (Jannuary/February 2000), pp:18-21, 2000.

[2] Reifer, D.J. Web Development: Estimating Quick-to-Market Software, *IEEE Software*, (November/December 2000), pp:57-64, 2000.

[3] Mendes, E., Watson, I., Triggs, C., Mosley, N., and Counsell, S. A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications, to be published in *Proceedings of the IEEE Metrics Symposium*, 2002.

[4] Mendes, E., Counsell, S., and Mosley, N. Measurement and Effort Prediction of Web Applications, *Proc. Second ICSE Workshop on Web Engineering*, 4 and 5 June 2000; Limerick, Ireland, 2000.

[5] Kirsopp, C. and Shepperd, M. *Making Inferences with Small Numbers of Training Sets*, January 2001, TR02-01.

[6] Mendes, E., Mosley, N., and Watson, I. A Comparison of Case-based reasoning approaches to Web Hypermedia Project Cost Estimation, to be published in *Proceedings of the WWW'2002 Conference*, 2002.

[7] Mendes, E., Mosley, N., and Counsell, S. Web Metrics – Estimating Design and Authoring Effort. *IEEE Multimedia*, Special Issue on Web Engineering, (January/March 2001), pp:50-57, 2001.

[8] Mendes, E., Mosley, N., and Counsell, S. A Comparison of Length, Complexity & Functionality as Size Measures for Predicting Web Design & Authoring Effort, *Proceedings of the 2001 EASE Conference*, Keele, UK, pp:1-14, 2001.

[9] Cowderoy, A.J.C. 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, pp:423-431, 2000.

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

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

[12] Briand, L.C., and Wieczorek, I. Resource Estimation in Software Engineering, to be published in 2nd edition of the Encyclopaedia of Software Engineering, Wiley, J. Marciniak eds., 2002.

[13] Garzotto, F., Paolini, P., and Schwabe, D. HMD – A Model-Based Approach to Hypertext Application Design, *ACM Transactions on Information Systems*, 11, 1, January, 1993.
[14] Schwabe, D. and Rossi, G. From Domain Models to Hypermedia Applications: An Object-Oriented Approach, *Proceedings of the International Workshop on Methodologies for Designing and Developing Hypermedia Applications*, Edimburgh, September, 1994.
[15] Balasubramanian, V., Isakowitz, T., and Stohr, E.A. RMM: A Methodology for Structured Hypermedia Design, *Communications of the ACM*, 38, 8, August, 1995.
[16] Coda, F., Ghezzi, C., Vigna, G., and Garzotto, F. Towards a Software Engineering Approach to Web Site Development, *Proceedings of the 9th International Workshop on Software Specification and Design*, pp.8-17, 1998.
[17] Spiro, R. J., Feltovich, P. J., Jacobson, M. J., and Coulson, R. L. Cognitive Flexibility, Constructivism, and Hypertext: Random Access Instruction for Advanced Knowledge Acquisition in Ill-Structured Domains. In: L. Steffe & J. Gale, eds., *Constructivism*, Hillsdale, N.J.:Erlbaum, 1995.
[18] Kadoda, G., Cartwright, M., and Shepperd, M.J. Issues on the effective use of CBR technology for software project prediction, *Proceedings of the 4th International Conference on Case-Based Reasoning*, ICCBR 2001, Vancouver, Canada, (July/August 2001), pp: 276-290, 2001.
[19] Schroeder, L., Sjoquist, D., and Stephan, P. Understanding Regression Analysis: An Introductory Guide. No 57. In Series: Quantitative Applications in the Social Sciences, Sage Publications, Newbury Park, CA, USA, 1986.
[20] Shepperd, M.J., Schofield, C., and Kitchenham, B., Effort Estimation Using Analogy, in *Proceedings ICSE-18*, IEEE Computer Society Press, Berlin, 1996.
[21] Shepperd, M.J., and Kadoda, G. Using Simulation to Evaluate Prediction Techniques, in *Proceedings of the IEEE 7th International Software Metrics Symposium*, London, UK, pp: 349-358, 2001.
[22] Kok, P., Kitchenham, B. A., Kirakowski, J. The MERMAID Approach to software cost estimation, *ESPRIT Annual Conference*, Brussels, pp:296-314, 1990.
[23] Watson, I. *Applying Case-Based Reasoning: techniques for enterprise systems*. Morgan Kaufmann, San Francisco, USA, 1997.
[24] Okamoto, S., Satoh, K. An average-case analysis of k-nearest neighbor classifier. In *Case-Based Reasoning Research and Development*, Veloso, M., & Aamodt, A. (Eds.) Lecture Notes in Artificial Intelligence 1010 Springer-Verlag, 1995.
[25] Kemerer, C.F. An Empirical Validation of Software Cost Estimation Models, *Communications of the ACM*, 30, 5, pp:416-429, 1986.
[26] Myrtveit, I. and Stensrud, E. A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models, *IEEE Transactions on Software Engineering*, 25, 4, (July/August 1999), pp. 510-525, 1999.
[27] Kitchenham, B.A., Pickard, L.M., MacDonell, S.G., Shepperd, M.J. What accuracy statistics really measure, *IEE Proceedings - Software Engineering,* June 2001, 148, 3.
[28] Conte, S., Dunsmore, H., and Shen, V. *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, California, 1986.
[29] Angelis, L., and Stamelos, I. A Simulation Tool for Efficient Analogy Based Cost Estimation, *Empirical Software Engineering*, 5, pp:35-68, 2000.
[30] Kadoda, G., Cartwright, M., Chen, L., and Shepperd, M.J. Experiences Using Case-Based Reasoning to Predict Software Project Effort, in *Proceedings of the EASE 2000 Conference*, Keele, UK, 2000.
[31] Schulz, S. CBR-Works - A State-of-the-Art Shell for Case-Based Application Building, in *Proceedings of the German Workshop on Case-Based Reasoning*, GWCBR'99 (1999). Lecture Notes in Artificial Intelligence Springer-Verlag, 1999.
[32] Walkerden F., and Jeffery R. An Empirical Study of Analogy-based Software Effort Estimation. *Empirical Software Engineering*, 4,2, (June 1999), 135-158.