Case Based Reasoning Models in Management Application

Fariba Shadabi
Division of Health, Design and Science,
University of Canberra, ACT 2601,
Australia

Mehrdad Khodai-Joopari
School of Computer Science, University of
New South Wales (ADFA campus), ACT 2600,
Australia

Abstract

In this study we investigate the need for Case-Based Reasoning (CBR) model calibration. This paper also presents some guidelines on building CBR model tailored to a specific management application. To address some of the practical issues associated with the use of case-based reasoning or estimation by analogy, we conducted an experiment on software effort estimation using a well-known project effort dataset, namely Albrecht. We found that pruning the irrelevant features can improve the performance of CBR applications and it is essential to calibrate the prediction model carefully.

1. Introduction

Over the years a variety of techniques have been purposed for management application such as loan evaluation and diagnosis programs. In recent years there has been great amount of interest on the application of case-based reasoning (CBR) [1], [2], [3]. This technique solves new problems by adapting solutions that were used to solve old problems.

The CBR approach is easy to understand and apply. The model can be used in conjunction with expert judgment. End-user can participate in the prediction process and use his or her judgment to modify (adopt) the prediction. However there are some important design decisions that must be made in order to generate a reasonably accurate estimation [1]. Such decisions include choice of distance metrics or similarity measures in order to measure the level of similarities between cases, number of most similar cases that need to be used to generate the estimation and choice of optimum feature set that gives the most accurate estimation. Studying the importance of these design decisions was the motivation for this paper.

Research shows that case-based reasoning or estimation by analogy can be successfully adapted in software effort or cost estimation domain [1], [4], [5], [6]. ANGEL [7] and CBR-Works [8] are two examples of CBR tools, which can make use of previous projects to estimate effort for a new project (see [9] for other CBR tools currently available on the market). The goal of this paper is to present some of the practical issues associated with the use of CBR models in cost estimation as a prototypical management application.

2. Research Method

The idea of effort estimation by analogy has been around for a long time, however this method has not been widely used. Shepperd and Schofield (1996) presented the idea of estimation by analogy in the form of detailed estimation methodology and developed a CBR tool [10], called ANGEL (ANaloGy Estimation tooL).

In order to investigate the effects of certain parameters in accuracy of CBR applications we used ANGEL [7], CBR-Works 4.0 [8] and a CBR system that was developed during this study. The CBR-Works tool can be used for the development and maintenance of management applications in a variety of domains and environments [11]. Throughout this study, we used a well-known dataset, namely Albrecht.

In this section, the Albrecht dataset and the measurements used to evaluate CBR modeling techniques will be briefly described.

2.1 Dataset

For our investigation we used the Albrecht dataset, which is a publicly available dataset [12], [13]. Albrecht is a relatively small (24 cases) software development project dataset. During this investigation it was decided to keep the original dataset as intact as possible.

A statistical summary of all the features (metrics) in the Albrecht dataset is presented in Table 1.

Table 1. Summary statistics for Albrecht dataset

Feature	Count	Min	Max	Mean	Me-dian	St. Dev	Skew-ness
Effort	24	0.5	105	21.9	11.5	28.4	2.3
FP	24	100	1902	643.3	506	493	1.5
File	24	3	60	17.4	11.5	15.5	1.5
Input	24	7	193	40.3	33.5	36.9	3.3
Inquiry	24	0	75	16.9	13.5	19.3	2.1
Output	24	12	150	47.3	39	35.2	1.4
SLOC	24	3	318	61.1	51.7	63.7	3.1

2.2 Performance Evaluation

In order to study the accuracy of the estimation models, we adopted the jackknifing technique. This technique is also known as leave-one-out cross validation. Each completed project in turn was removed from the dataset and the remaining dataset was then used to estimate the effort for the removed project. MMRE (Mean Magnitude of Relative Error) and PRED (25) were computed from the jackknifing process [14]. MMRE is an error measurement method that has been used by various researchers [15], [16], [17]. Smaller MMRE value indicates better prediction model. MMRE is defined as:

$$100/n\sum_{i=1}^{n}\left(\frac{|\,ActualEffort - PredictedEffort\,|}{ActualEffort}\right)_i$$

Where n represent the number of projects in the dataset.

Pred (25) measures the proportion of predictions that are within 25% of the actual values. Clearly, the higher this value is, the better.

3. Experiment Results

During this study we had to decide upon five parameters as follows:

Scaling or Standardization: All feature's values for the projects can be standardized between 0 and 1. By performing standardization we can ensure that all features have equal influences to the measure of similarity and the method is immune to the choice of units.

Feature set: All collected features may not be helpful in finding a good estimation. By the use of brute force algorithm, ANGEL can automatically determinate the best subset of features. However the brute force algorithm uses exhaustive search, so as the number of features increase the process gets slower [1]. Unlike ANGEL, CBR-Works cannot determine the best possible feature set for a particular dataset.

Similarity measure: Here the question is how much the new project is similar to the other projects in the available dataset. One good feature of CBR-Works is that it provides a variety of retrieval algorithms such as:

- Euclidean distance: The formula is similar to the one used in ANGEL. The CBR users can assign different weights to the features in order to reflect the relative importance of each feature.
- Average: The average similarity of all attributes defines the case similarity.
- Maximum: The highest feature similarity defines the case similarity.
- Minimum: The lowest feature similarity defines the case similarity.

Number of analogies: Number of analogies refers to the number of closest projects that can be used to generate estimation for the new case. The simplest way is to consider only the effort of the most similar case as estimation for the new case (1 analogy). However there are other alternative strategies that can be considered [18]. In this study we used 1,2 and 3 analogies similar to [1].

Analogy adaptation: When the analogy projects are selected, the question is what would be the best analogy adaptation technique so that the best estimation for the new case can be generated? We decided to choose the same adaptation process that have been used previously by Schofield [1] as outlined below:

- One analogy: Estimation for new case is the effort from the closest analogy.
- Two analogies: Estimation for new case is the average of efforts for the closest two analogies.
- Two analogies (Weighted): Estimation for new case is the average of efforts for the two closest analogies; however the closest analogy is weighted double.
- Three analogies: Estimation for new case is the average of efforts for the closest three analogies.

Table 2 shows the MMRE and Pred (25) results obtained for the following scenarios:

1. When the full features available in the dataset were used through CBR-Works.

2. When optimum combination of features was generated for use via ANGEL.

The results suggest that for the Albrecht dataset ANGEL outperforms CBR-Works. The results reveal that the performance of CBR-Works is poor because it is not calibrated correctly.

Table 2. Comparing ANGEL and CBR-Works models

Feature	Analogy	MMRE %	Pred (25) %
Full set of features (Using CBR-Works)	One	96	16
	Two	74	33
	Two (W)	129	21
	Three	85	21
Optimum features Sub-set (Using ANGEL)	One	67	33
	Two	66	37
	Two (W)	**61***	**41***
	Three	62	33

In configuring the CBR-Works we had to make decisions concerning similarity function, retrieval algorithm and pruning the feature set. We tried to find the best feature set that minimize the mean absolute relative errors by assigning different weights to the features, based on their relative importance.

Table 3. Comparison of distance measures

Dist. Metric	Analogy	MMRE %	Pred (25) %
Euclidean (Un-weighted)	One	96.47	16.66
	Two	74.20	33.33
	Two (W)	129.47	20.83
	Three	85.01	20.83
Average (Un-weighted)	One	97.40	16.36
	Two	**71.25***	**33.33***
	Two (W)	126.05	20.83
	Three	83.38	20.83
Euclidean (Weighted)	One	78.47	16.66
	Two	75.80	33.33
	Two (W)	118.86	20.83
	Three	75.18	29.16
Average (Weighted)	One	99.27	12.50
	Two	75.56	33.33
	Two (W)	128.74	16.66
	Three	87.00	20.83

An obvious choice to determine the degree of importance for each feature was to choose only features that have a strong statistical influence on effort. The results suggest that different configurations of CBR-Works may produce different levels of accuracy. As can be seen from Table 3, two analogies, with average retrieval algorithm (unweighted) was found to be the most accurate prediction by predicting 33% of projects within 25% of their actual effort and with MMRE of 71%.

3.1 Practical Advice

Many CBR tools are currently available on the market [9]. In this paper, CBR-Works, 4.0, was used to implement different CBR techniques for estimating the development effort.

The ANGEL is another CBR tool that previously was used by Schofield [1] for the purpose of software cost estimation. Both estimation tools are provided with an easy-to-use interface that can support the stage of data collection, effort prediction and adoption rule. However the functionality of collecting and pasting specific items were not provided in neither of them. These functionalities can be a great help for the person responsible for reviewing the model and ensuring its accuracy.

A tool for estimating the development effort, based on analogy can be easily implemented in any programming language. Readers who are building their first CBR system and are interested in most important issues in building and maintaining CBR systems should refer to [9], [19] and [20] for more information. Figure 1 shows the prototype for automatic software estimation that we developed using MATLAB[1]. The purpose of this tool is to help project managers make important decisions regarding certain parameters (e.g. number of analogies or distance metrics) when analogy based estimation is required. We named the tool "Calibration of the Analogy Procedure". This tool needs relatively little effort in order to generate useful results.

[1] MATLAB is a commercial (Matrix Laboratory) package. More information also can be found on: http://WWW.math.ufl.edu/help/matlab-tutorial/matlab-tutorial.html

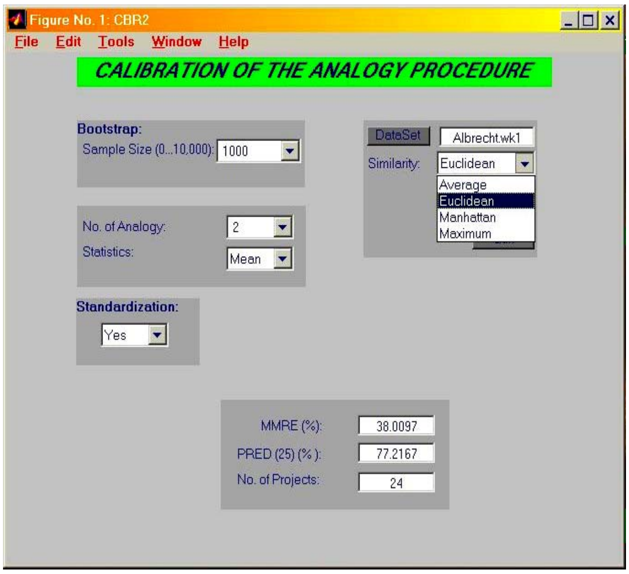

Figure 1. An example of Calibration of the Analogy Tool

The results in Table 4 have been obtained using four similarity measures, namely Euclidean (Un-weighted), Average (Un-weighted), Manhattan [18], Maximum, six choices for number of analogies (2,3,5,10,15 and 20), one choice of calculating the analogy (mean) and based on standardized (Std) /non-standardized feature values.

As can be seen from Table 4, two analogies, with average or Euclidean retrieval algorithm (non-normalized values) was found to be the most accurate prediction by predicting 67% of projects within 25% of their actual effort and with MMRE of 57%.

The results in Table 5 were generated with the help of a statistical simulation method, namely bootstrap in order to generate multiple samples for the accuracy indicators [18], [20]. This method can help the user to assess the accuracy of the tool or calibrate the analogy tool before its application to real projects. The results confirmed that two analogies, with average or Euclidean retrieval algorithm is the best choice.

Table 4. Comparison of various parameters in estimation by analogy

Distance	Analogy	Std	Statis-tic	MMRE (%)	PRED (25%)
Average	2	Yes	Mean	62.78	54.17
		No		**56.89**	**66.67**
	3	Yes	Mean	66.19	58.33
		No		66.18	58.33
	5	Yes	Mean	71.92	54.17
		No		83.48	20.83
Euclidean					
	2	Yes	Mean	62.78	54.17
		No		**56.89**	**66.67**
	3	Yes	Mean	66.19	58.33
		No		66.18	58.33
	5	Yes	Mean	71.92	54.17
		No		83.48	20.83
Manhattan					
	2	Yes	Mean	64.34	54.17
		No		62.02	58.33
	3	Yes	Mean	73.40	50
		No		73.96	50
	5	Yes	Mean	76.59	50
		No		83.77	25
Maximum					
	2	Yes	Mean	90.18	33.33
		No		62.02	58.33
	3	Yes	Mean	81.01	33.33
		No		73.96	50
	5	Yes	Mean	63.04	0
		No		83.77	25

Table 5. Comparison of various parameters in estimation by analogy (1000 bootstrap samples)

Distance	Analogy	Std	Statistic	MMRE (%)	PRED (25%)
Average	2	Yes	Mean	**38.12**	**76.72**
		No		40.65	75.21
	3	Yes	Mean	48.96	67.88
		No		52.14	61.45
	5	Yes	Mean	60.05	50.11
		No		70.66	51.34
Euclidean					
	2	Yes	Mean	**38.28**	**77.09**
		No		44.69	75.28
	3	Yes	Mean	49.09	67.64
		No		52.69	61.89
	5	Yes	Mean	59.95	49.70
		No		70.36	51.65
Manhattan					
	2	Yes	Mean	40.95	74.10
		No		43.19	73.33
	3	Yes	Mean	52.34	65.82
		No		54.21	59.30
	5	Yes	Mean	64.52	45.18
		No		71.38	52.37
Maximum					
	2	Yes	Mean	53	55.79
		No		42.99	73.19
	3	Yes	Mean	58.69	56.16
		No		54.54	59.37
	5	Yes	Mean	54.27	31.68
		No		70.93	50.67

Each organization applying analogy can design and implement a tool similar to the tool implemented in this study to meet their needs and generate useful results.

4. Conclusion and future work

Case-based reasoning or estimation by analogy is a relatively simple technique however it is essential to calibrate the prediction model carefully. Each organization applying analogy may design and implement a CBR tool tailored to their needs. Estimator should review the accuracy of CBR models at various time periods and configure the tool in best possible way (e.g. choosing the best combination of parameters). Clearly still there is a need for further investigation in this area.

As a continuation of our current work, we are currently developing a tool for predicting the outcomes of renal transplants and the type of graft rejection based on analogy. We also plan to compare the performance of CBR techniques in predicting medical outcomes against Artificial Neural Networks models.

References

[1] Schofield, C.: An Empirical Investigation Into Software Effort Estimation By Analogy**.** PhD thesis (1997)

[2] Kadoda, G., Cartwright M.: Issues on the effective use of CBR technology for software project prediction. Poole, UK, School of Design, Engineering Research Group, Bournemouth University (2001)

[3] Mendes, E., Watson, I.: A comparison of Development Effort Estimation Techniques for Web Hypermedia applications. International Software Metrics symposium (2002)

[4] Bisio, R., Malabocchia, F.: Cost Estimation of Software Projects Through Case Based Reasoning. International Conference on Case Based Reasoning, Sesimbra Portugal (1995)

[5] Finnie, G. R., Writtig, G. E.: A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models. J.SYSTEMS SOFTWARE (1997) 281–289

[6] Walkerden, F., Jeffery R.: An Empirical Study of Analogy-based software effort Estimation. The University of New South Wales, Technical Report (1998)

[7] http://dec.bmth.ac.uk/ESERG/ANGEL/

[8] CBR-Works, 4.0 beta. Research group.: Artificial Intelligence-Knowledge-Based System, University of Kaiserslautern http://wwwagr.informatik.uni-kl.de/~lsa/CBRatUKL.html

[9] Watson. I: Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann publishers Inc. (1997)

[10] Shepperd, M., Schofield C., Kitchenham, B.: Effort Estimation Using Analogies. Proceeding of 18th int. conference on software engineering, IEEE Com. Soc. Press (1996)

[11] Stefan, 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. Lecture Notes in Artificial Intelligence Springer-Verlag (1999)

[12] Albrecht, A. J.: Measuring Application Development Productivity. Proceedings of the IBM Applications Developments Symposium. (1979)

[13] Albrecht, A. J., Gaffney, J. E.: Software Function, Source Lines of Code, and Development Effort Prediction: A

Software Science Validation. IEEE Transactions on Software engineering 9(6) (1983) 639-648

[14] Conte, S., Dunsmore, H., Shen, V.Y.: Software Engineering Metrics and Models.” Menlo Park, Calif.: Benjamin Cummings (1986)

[15] Low, G. C., Jeffery R.: Function Points in the Estimation and Evaluation of the Software Process. IEEE Transactions on Software engineering 16(1) (1990) 64-71

[16] Kemerer, C.: An Empirical Validation of Software Cost Estimation Models. ACM 30(5): (1987) 416-429

[17] Mukhopadhyay, t., Vicinaza, S. S., Prietula, M. J.: Examining the feasibility of a case-based reasoning model for software effort estimation.” MIS Quarterly 16(2) (1992) 155–171

[18] Angelis, L., Stamelos, I.: A simulation Tool for Efficient Analogy Based Estimation. Empirical Software Engineering (2000) 35-68

[19] Heister, F., Wilke, W.: An Architecture for Maintaining Case-Based Reasoning Systems. Advances in Case-Based Reasoning, 221-232, Springer LNAI 1488, Berlin (1998)

[20] Efforn, B., Tibshirani, R.: An introduction to the Bootstrap (1998) New York: Chapman and Hall (1993) Gennari, J., et al., Templates for IEEE Proceedings, available at
<http://www.sheridanprinting.com/ieee/ieee.htm>