The Use of a Bayesian Network for Web Effort Estimation

Emilia Mendes

The University of Auckland, Private Bag 92019
Auckland, New Zealand, 0064 9 3737599 ext. 86137
emilia@cs.auckland.ac.nz

Abstract. The objective of this paper is to describe the use of a probabilistic approach to Web effort estimation by means of a Bayesian Network. A Bayesian Network is a model that embodies existing knowledge of a complex domain in a way that supports reasoning with uncertainty. Given that the causal system relative to Web effort estimation has an inherently uncertain nature the use of Bayesian model seemed a reasonable choice. We used a cross-company data set of 150 industrial Web projects volunteered from Web companies worldwide, which are part of the Tukutuku database. Results showed that the effort estimates obtained using a Bayesian Network were sound and significantly superior to the prediction based on two benchmark models, using the mean and median effort respectively.

Keywords: Web effort estimation, Bayesian Networks, Effort accuracy, Web cost estimation.

1 Introduction

A cornerstone of Web project management is sound resource estimation, the process by which resources are estimated and allocated effectively, enabling projects to be delivered on time and within budget. Resources are factors, such as cost, effort, quality, 'problem size', that have a bearing on a project's outcome. Within the scope of resource estimation, the causal relationship between factors is not deterministic and has an inherently uncertain nature. E.g. assuming there is a relationship between development effort and an application's quality, it is not necessarily true that increased effort will lead to improved quality. However, as effort increases so does the *probability* of improved quality. Resource estimation is a complex domain where corresponding decisions and predictions require reasoning with uncertainty.

In Web project management the complete understanding of what factors affect a project's outcome and the causal relationships between factors is unknown. In addition, as Web development differs substantially from software development 0, there is very little research on resource estimation for software projects that can be readily reused.

Web development, despite being a relatively young industry, initiated just 13 years ago, currently represents a market that increases at an average rate of 20% per year, with Web e-commerce sales alone surpassing 95 billion USD in 2004 (three times the

L. Baresi, P. Fraternali, and G.-J. Houben (Eds.): ICWE 2007, LNCS 4607, pp. 90–104, 2007.

revenue from the world's aerospace industry)[1][33]. Unfortunately, in contrast, most Web development projects suffer from unrealistic project schedules, leading to applications that are rarely developed on time and within budget [33].

To understand resource estimation for Web projects, previous studies have developed models that use as input, factors such as the size of a Web application, and cost drivers (e.g. tools, developer's quality, team size), and provide an effort estimate as output. The differences between these studies were the number and type of size measures used, choice of cost drivers and occasionally the techniques employed to build resource estimation models. Despite previous studies, to date no complete understanding of which factors affect a Web project's outcome, their causal relationships, and the uncertainty inherent to such relationships has been achieved.

Important reasons for this gap in knowledge are: i) the use of techniques to build resource estimation models that fail to represent the causal relationship between factors and their corresponding uncertainty, and require the use of large amounts of data that is often difficult to obtain; ii) a strong reliance on obtaining the "correct" causal model using simple statistical models, which are inadequate to accommodate complex relationships between all the relevant factors [8]; iii) until recently, the non-existence of appropriate algorithms and corresponding software tools [11] to enable the building of large causal models that allow for uncertainty and probabilistic reasoning; iv) the relatively new research area of resource estimation for Web projects with the first study published in 2000 [19].

There have been numerous attempts to model resource estimation of Web projects, but none yielded a complete causal model incorporating all the necessary component parts. Mendes and Counsell [19] were the first to investigate this field by building a model that used machine-learning techniques with data from student-based Web projects, and size measures harvested late in the project's life cycle. Mendes and collaborators also carried out a series of consecutive studies [10],[18],[19]-[28] where models were built using multivariate regression and machine-learning techniques using data on industrial Web projects. Recently they also proposed and validated size measures harvested early in the project's life cycle, and therefore better suited to resource estimation [22].

Other researchers have also investigated resource estimation for Web projects. Reifer [34] proposed an extension of an existing software engineering resource model, and a single size measure harvested late in the project's life cycle. None were validated empirically. This size measure was later used by Ruhe et al. [35], who further extended a software engineering hybrid estimation technique to Web projects, using a small data set of industrial projects, mixing expert judgement and multivariate regression. Later, Baresi et al. 0,[2], and Mangia et al. [17] investigated effort estimation models and size measures for Web projects based on a specific Web development method. Finally, Costagliola et al. [4] compared two types of Web-based size measures for effort estimation.

The goal of our research is therefore to create and evaluate a large-scale Bayesian network [11] (BN) that represents the causal model for resource estimation of Web projects, incorporating all the fundamental factors and their causal relationships. A BN is a model that embodies existing knowledge of a complex domain in a way that supports reasoning with uncertainty [11][31]. It is a representation of a joint probability

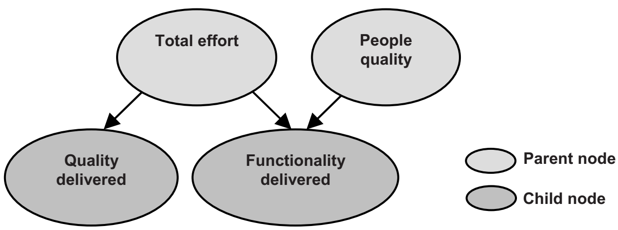
---

[1] http://www.aia-erospace.org/stats/aero_stats/stat08.pdf
http://www.tchidagraphics.com/website_ecommerce.htm

distribution over a set of variables, and is made up of two parts. The first, the qualitative part, represents the structure of a BN as depicted by a directed acyclic graph (digraph) (see Fig. 1). The digraph's nodes represent the relevant variables (factors) from the domain being modelled, which can be of different types (e.g. observable or latent, categorical, numerical). A digraph's arcs represent probabilistic relationships, i.e. they represent the causal relationships between variables [11][29][41]. The second, the quantitative part, associates a node probability table (NPT) to each node, its probability distribution. A parent node's NPT describes the relative probability of each state (value) (Fig. 1 "NPT for node Total Effort"); a child node's NPT describes the relative probability of each state conditional on every combination of states of its parents (Fig. 1 "NPT for node Quality delivered"). So, for example, the relative probability of Quality delivered (QD) being 'Low' conditional on Total effort (TE) being 'Low' is 0.8, and represented as:

- p(QD = 'Low' | TE = 'Low') = 0.8

Each column in a NPT represents a conditional probability distribution and therefore its values sum up to 1 [11].



NPT for node Total Effort (TE)	
Low	0.2
Medium	0.3
High	0.5

NPT for node Quality Delivered (QD)			
Total Effort	Low	Medium	High
Low	0.8	0.2	0.1
Medium	0.1	0.6	0.2
High	0.1	0.2	0.7

Fig. 1. A small BN model and two NPTs

Formally, the relationship between two nodes is based on Bayes' rule [11][31]:

$$p(X \mid E) = \frac{p(E \mid X)\, p(X)}{p(E)} \tag{1}$$

where:

- $p(X \mid E)$ is called the *posterior* distribution and represents the probability of *X* given evidence *E*;
- $p(X)$ is called the *prior* distribution and represents the probability of *X* before evidence *E* is given;
- $p(E \mid X)$ is called the *likelihood* function and denotes the probability of *E* assuming *X* is true.

Once a BN is specified, evidence (e.g. values) can be entered onto any node, and probabilities for the remaining nodes are automatically calculated using Bayes' theorem [31][41]. Therefore BNs can be used for different types of reasoning, such as predictive and "what-if" analyses to investigate the impact that changes on some nodes have upon others [37].

The BN described and validated in this paper focuses on Web effort estimation. This BN comprises a subset of a more complete BN, and was chosen since this is the only BN within the scope of our research that was built from data on Web projects, as opposed to being elicited from interviews with domain experts. We had the opportunity to gather data on 150 industrial Web projects as part of the Tukutuku Benchmarking project [22], and to use these data to build and validate the BN presented herein. The project data characterises Web projects using size measures and cost drivers targeted at effort estimation. Since we had a dataset of real industrial Web projects, we were also able to compare the accuracy of our Web effort BN to that provided using a mean and median effort models, which are used here as a benchmark. To do so we computed point forecasts for the BN using the method described in [32], to be detailed later.

Prediction accuracy was measured using de facto measures such as the Mean Magnitude of Relative Error (MMRE), Median Magnitude of Relative Error (MdMRE) and Prediction at 25% (Pred(25)) [4].

The remainder of the paper is organised as follows: Section 2 describes the procedure used to build and validate the Web effort BN. Section 3 presents the results using for the Web effort BN, and for the mean and median effort models. Finally, conclusions and comments on future work are given in Section 4.

2 Building the Web Effort BN

2.1 Dataset Description

The analysis presented in this paper was based on data from 150 Web projects of the Tukutuku database [22], which aims to collect data from completed Web projects, to be used to develop Web cost estimation models and to benchmark productivity across and within Web Companies. The Tukutuku includes data on 150 Web hypermedia systems and Web applications [3] where:

- Projects come from 10 different countries, mainly New Zealand (56%), Brazil (12.7%), Italy (10%), Spain (8%), United States (4.7%), England (2.7%), and Canada (2%).
- Project types are new developments (56%) or enhancement projects (44%).
- The applications are mainly Legacy integration (27%), Intranet and eCommerce (15%).
- The languages used are mainly HTML (88%), Javascript (DHTML/DOM) (76%), PHP (50%), Various Graphics Tools (39%), ASP (VBScript, .Net) (18%), and Perl (15%).

Each Web project in the database was characterized by 25 variables, related to the application and its development process (see Table 1). These size measures and cost drivers have been obtained from the results of a survey investigation [22], using data from 133 on-line Web forms aimed at giving quotes on Web development projects. In addition, these measures and cost drivers have also been confirmed by an established Web company and a second survey involving 33 Web companies in New Zealand. Consequently it is our belief that the 25 variables identified are measures that are meaningful to Web companies and are constructed from information their customers can provide at a very early stage in project development.

Table 1. Tukutuku database variables

Variable name	Scale	Description
COMPANY DATA		
Country	Categorical	Country company belongs to.
Established	Ordinal	Year when company was established.
nPeopleWD	Ratio	Number of people who work on Web design and development.
PROJECT DATA		
TypeProj	Categorical	Type of project (new or enhancement).
nLang	Ratio	Number of different development languages used
DocProc	Categorical	If project followed defined and documented process.
ProImpr	Categorical	If project team involved in a process improvement programme.
Metrics	Categorical	If project team part of a software metrics programme.
Devteam	Ratio	Size of project's development team.
Teamexp	Ratio	Average team experience with the development language(s) employed.
TotEff	Ratio	Actual total effort in person hours used to develop the Web application.
estEff	Ratio	Estimated total effort in person hours necessary to develop the Web application.
Accuracy	Categorical	Procedure used to record effort data.
WEB APPLICATION		
TypeApp	Categorical	Type of Web application developed.
TotWP	Ratio	Total number of Web pages (new and reused).
NewWP	Ratio	Total number of new Web pages.
TotImg	Ratio	Total number of images (new and reused).
NewImg	Ratio	Total number of new images created.
Fots	Ratio	Number of features reused without any adaptation.
HFotsA	Ratio	Number of reused high-effort features/functions adapted.
Hnew	Ratio	Number of new high-effort features/functions.
totHigh	Ratio	Total number of high-effort features/functions
FotsA	Ratio	Number of reused low-effort features adapted.
New	Ratio	Number of new low-effort features/functions.
totNHigh	Ratio	Total number of low-effort features/functions

Within the context of the Tukutuku project, a new high-effort feature/function requires at least 15 hours to be developed by one experienced developer, and a high-effort adapted feature/function requires at least 4 hours to be adapted by one experienced developer. These values are based on collected data.

Table 2 summarises the number and percentages of projects for the categorical variables, and summary statistics for the numerical variables from the Tukutuku database are given in Table 3.

Table 2. Summary of Number of Projects and Percentages for Categorical variables

Variable	Level	Num. Projects	% Projects
TypeProj	Enhancement	66	44
	New	84	56
DocProc	No	53	35.3
	Yes	97	64.7
ProImpr	No	77	51.3
	Yes	73	48.7
Metrics	No	85	56.7
	Yes	65	43.3

Table 3. Summary Statistics for Numerical variables

	Mean	**Median**	**Std. Dev.**	**Min.**	**Max.**
nlang	3.75	3.00	1.58	1	8
DevTeam	2.97	2.00	2.57	1	23
TeamExp	3.57	3.00	2.16	1	10
TotEff	564.22	78.00	1048.94	1	5000
TotWP	81.53	30.00	209.82	1	2000
NewWP	61.43	14.00	202.78	0	1980
TotImg	117.58	43.50	244.71	0	1820
NewImg	47.62	3.00	141.67	0	1000
Fots	2.05	0.00	3.64	0	19
HFotsA	12.11	0.00	66.84	0	611
Hnew	2.53	0.00	5.21	0	27
totHigh	14.64	1.00	66.59	0	611
FotsA	1.91	1.00	3.07	0	20
New	2.91	1.00	4.07	0	19
totNHigh	4.82	4.00	4.98	0	35

As for data quality, we asked companies how their effort data was collected (see Table 4). At least for 83% of Web projects in the Tukutuku database effort values were based on more than guesstimates.

Table 4. How Effort data was gathered

Data Collection Method	**# of Projects**	**% of Projects**
Hours worked per project task per day	93	62
Hours worked per project per day/week	32	21.3
Total hours worked each day or week	13	8.7
No timesheets (guesstimates)	12	8

2.2 Procedure Used to Build the BNs

The BN presented in this paper was built and validated using an adapted Knowledge Engineering of Bayesian Networks (KEBN) process [6][16][41] (see Fig. 2). In Fig. 2

arrows represent flows through the different processes, depicted by rectangles. Such processes are executed either by people – the Knowledge Engineer (KE) and the Domain Experts (DEs) [41] (white rectangles), or automatic algorithms (dark grey rectangles). Within the context of this research project this author is the knowledge engineer, and Web project managers from Web companies in Rio de Janeiro and Auckland are the domain experts.

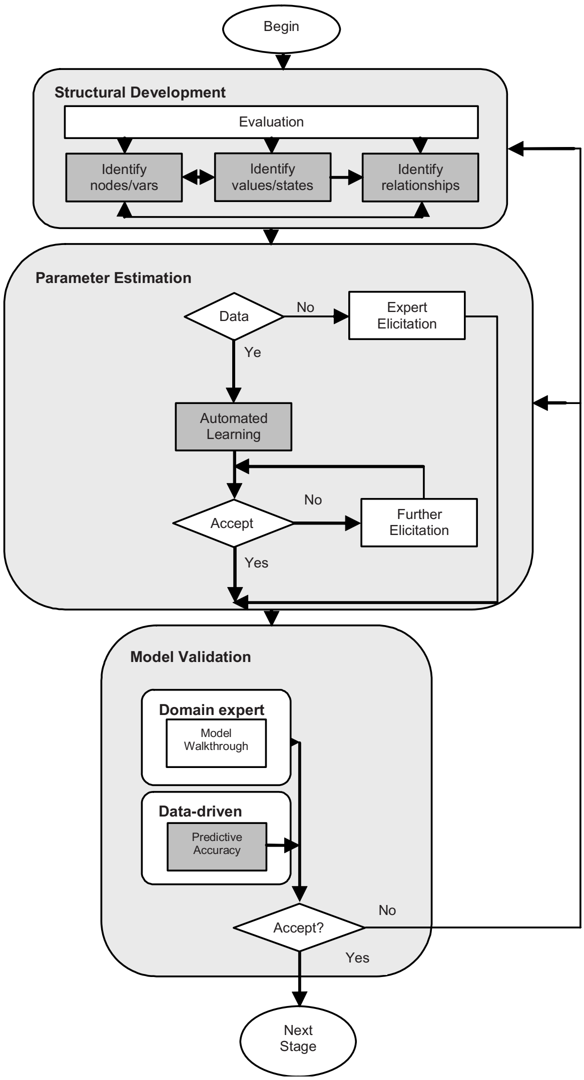


Fig. 2. KEBN, adapted from [40]

The three main steps within our KEBN process are the Structural Development, Parameter Estimation, and Model Validation. This process iterates over these steps until a complete BN is built and validated. Each of these three steps is detailed below:

Structural Development: This step represents the qualitative component of a BN, which results in a graphical structure comprised of, in our case, the factors (nodes, variables) and causal relationships identified as fundamental for resource estimation of Web projects. This is an iterative process where independent BN's sub-models are identified. This model construction process has been validated in previous studies [7][9][16][29][41] and uses the principles of problem solving employed in data modelling and software development [39]. Also, the BN tool we used (Hugin Expert) allows for the representation of sub-models, thus facilitating the application of our modelling approach. Existing literature in Web resource estimation, data from the Tukutuku database and current knowledge from domain experts are employed to elicit the BN's structure. In the context of this paper we have used data from the Tukutuku database and current knowledge from a domain expert who works in a well-established Web company in Rio de Janeiro (Brazil). The identification of nodes, values and relationships was initially obtained automatically using Hugin, and later modified once feedback was obtained from the domain expert and the conditional independences were checked. In addition to identifying variables, their types (e.g. query variable, evidence variable) and relationships, domain experts in general also choose what states (values) each variable should take, and if they are discrete or continuous. In practice, currently available BN tools require that continuous variables be discretised by converting them into multinomial variables [14], also the case with Hugin Expert. Hugin offers two discretisation algorithms – equal-width intervals [36], whereby all intervals have equal size, and equal-frequency intervals, whereby each interval contains n/N data points where n is the number of data points and N is the number of intervals (this is also called maximal entropy discretisation [40]). We used equal-frequency intervals as suggested in [13], and five intervals. Automatic discretisation frees domain experts and knowledge engineers from having to statically discretise variables manually [30]. Throughout this step the knowledge engineer also evaluated the structure of the BN in two stages. The first entailed checking if [14]: variables and their values have a clear meaning; all relevant variables for that cycle have been included; variables are named conveniently; all states are appropriate (exhaustive and exclusive); a check for any states that can be combined. The second stage entailed reviewing the graph structure of the BN to make sure any identified d-separation dependencies comply with the types of variables used and causality assumptions. D-separation dependencies are used to identify variables influenced by evidence coming from other variables in the BN [11][31]. Once the BN structure is assumed to be close to final we may still need to optimise this structure to reduce the number of probabilities that need to be assessed for the network. If optimisation is needed then we employ techniques that change the graphical structure (e.g. divorcing [11]) and the use of parametric probability distributions (e.g. noisy-OR gates [7][31]). In the case of the Web effort BN we changed its original graphical structure to maintain the conditional independence of the nodes (see Section 2.3), however divorcing was not employed in order to keep only nodes that had been elicited from the Tukutuku data.

Parameter Estimation: This step represents the quantitative component of a BN, which results in conditional probabilities, obtained via Expert Elicitation or automatically, which quantify the relationships between variables [11][14]. For the Web effort BN, they were obtained using two steps: first, by automatically fitting a sub-network to a subset of the Tukutuku dataset (Automated learning); second, by obtaining feedback from the domain expert regarding the suitability of priors and conditional probabilities that were automatically fitted. No previous literature was used in this step since none reported probabilistic information. Of the 150 projects available in the Tukutuku database we used 120 (80%) to build the Web effort BN and later employed the remaining 30 for the Model Validation step to assess the BN's effort prediction accuracy.
Model Validation: This step validates the BN that results from the two previous steps, and determines whether it is necessary to re-visit any of those steps. Two different validation methods are used - Model Walkthrough and Predictive Accuracy, which specifically verifies if resource predictions provided by a BN are, on average, better than those currently used by Web companies. Predictive Accuracy is normally carried out using quantitative data, thus this was the validation approach we employed to validate the Web effort BN. Accuracy was measured using de facto measures such as the Mean MRE, median MRE and Pred(25), and estimated effort for each of the 30 projects in the validation set was obtained using a point forecast, computed using the method described in [32]. This method calculates the joint probability distribution of effort using the belief distribution [31], and computes estimated effort as the sum of the probability of a given effort scale point multiplied by its related mean effort. Within the context of our Web effort BN, effort was discretised using a five-scale point (see Section 2.3).

Model walkthrough represents the use of real case scenarios that are prepared and used by domain experts to assess if the predictions provided by a BN, or BN's submodel, correspond to the predictions experts would have chosen based on their own expertise. Success is measured as the frequency with which the BN's predicted value for a target variable (e.g. quality) that has the highest probability corresponds to experts' own assessment. We did not employ a model walkthrough to validate the Web effort BN because we had already carried out a Predictive accuracy procedure using real data volunteered by numerous Web companies worldwide.

2.3 The Web Effort BN

Fig. 3(a) shows the original Web effort BN obtained from fitting the data on Web projects. We used the entire Tukutuku database when building the structure, however for parameter estimation we only employed the 120 projects in the training set, otherwise the point estimates would be biased.

Once this structure was obtained using the Necessary Path Condition (NPC) algorithm [38], it was validated with a domain expert, resulting in the structure presented in Figure 3(b). The main changes to the original structure were related to node *TypeProj*, from which all causal relationships, except for *TotalEffort*, were removed. There were also several changes relating to the three categorical variables *Documented Process*, *Process Improvement* and *Use Metrics*. In the validated structure (see Figure 3(b)), *Process Improvement* presents a relationship with both *Use Metrics* and *Documented Process*, indicating that it is an important factor determining whether a Web company adheres to the use of metrics and to the use of a documented process. This structure also relates *Use Metrics* to *Documented Process*, indicating that companies that measure attributes to some extent document their processes. The number of languages to be used in a project (*numLanguages*) and the average number of years of experience of a team (*Team Experience*) are also related with the size of the development team (*sizeDevTeam*). The nodes relative to Web size measures (e.g. *NewWP*) remained unchanged as the data already captured the strong relationship between size and effort.

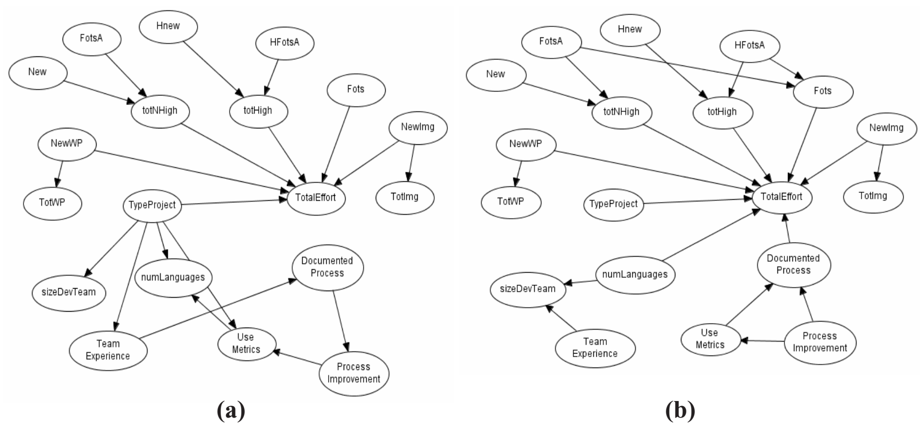


Fig. 3. Original BN (a) and BN after evaluation with DE (b)

Once the structure had been validated, our next step was to ensure that the conditionally independent variables (nodes) in the Web effort BN were really independent of each other [31]. Whenever two variables were significantly associated we also measured their association with effort, and the one with the strongest association was kept. For example, *Process improvement* was significantly associated with *Fots,* so one of these nodes had to be removed from the BN. Given that *Process Improvement* had a significant association with *TotalEffort* stronger than the association between *TotalEffort* and *Fots*, we kept *Process Improvement* in the model. This was an iterative process given that once nodes are removed (e.g. *FotsA*, *New*), other nodes become conditionally independent (e.g. *totNHigh*) and so need to be checked as well. The associations between the numerical variables were assessed using a non-parametric test - Spearman's rank correlation test; the associations between numerical and categorical variables were checked using the one-way ANOVA test, and the associations between categorical variables were checked using the Chi-square test. All tests were carried out using SPSS 12.0.1 and $\alpha = 0.05$.

Fig. 4 shows the Web effort BN after all conditional independences were checked. This was the Web effort BN used as input to the *Parameter estimation* step, where prior and conditional probabilities were automatically generated using the EM-learning algorithm [15], and later validated by the DE.

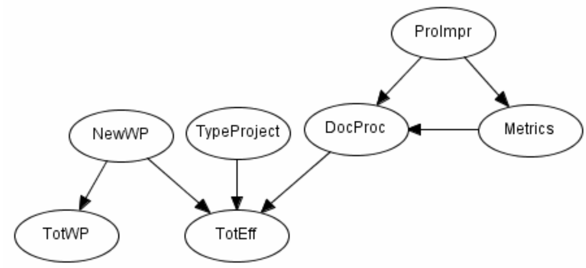


Fig. 4. BN after conditional independences were checked

Effort was discretised into five discrete approximations, described in Table 5.

Table 5. How Effort data was gathered

Categories	Range (person hours)	Mean Effort
Very low	<= 12.55	5.2
Low	> 12.55 and <= 33.8	22.9
Medium	> 33.8 and <= 101	63.1
High	> 101 and <= 612.5	314.9
Very High	> 612.5	2,238.9

TotWP and NewWP were also discretised into five discrete approximations. There are no strict rules as to how many discrete approximations should be used. Some studies have employed three [32], others five [9], and others eight [37]. We chose five. However, further studies are necessary to determine whether a different number of approximations leads to results significantly different. The NPTs for the seven nodes used in the Web effort BN are not presented here due to lack of space.

3 Measuring the Prediction Accuracy of the Web Effort BN

The 30 Web projects in the validation set were used to measure the prediction accuracy of the Web effort BN model. In addition, we also used the mean (526.9) and median (59.1) effort models as benchmark. Prediction accuracy was measured using MMRE, MdMRE, and Pred(25) and Table 6 shows that the MMRE and MdMRE obtained using the BN model was very close to the baseline predictions suggested in the literature (MMRE and MdMRE <= 25%). However, Pred(25) was lower than the suggested baseline of 75% or above. In addition, Table 6 also shows that the prediction accuracy for the Web Effort BN model was superior to the accuracy obtained with either the mean or median effort models.

In order to assess if the difference in accuracy between the Web Effort BN model and the mean & median models was not due to chance we also used a statistical significance test to compare the absolute residuals (actual effort – estimated effort) between these three models. Since none of the residuals were normally distributed, confirmed used the One-Sample Kolmogorov-Smirnov Test, they were compared using the non-parametric Wilcoxon Signed Paired Test. This test confirmed that the predictions obtained using the Web Effort BN model were significantly superior to the predictions from both the median and mean models. In addition, this test also showed that there were no significant differences between the median and mean effort models.

Table 6. Accuracy Measures for the Web Effort BN and Benchmarking models

Accuracy (%)	BN model	Mean model	Median model
MMRE	34.26	1106.31	132.76
MdMRE	27.42	252.36	85.90
Pred(25)	33.33	6.67	10.00

Fig. 5 shows boxplots of absolute residuals for the Web effort BN (ResBN), mean (Mean) and median (Median) models. The median of 'ResBN' is much lower than the median of 'Mean', and also lower than the median of 'Median'. All boxplots present outliers, however those for 'Mean' and 'Median' are much worse than the ones for 'ResBN'. The boxes for 'ResBN' and 'Mean' are flatter than the box for 'Median'. What these results suggest is that using a model that allows the representation of uncertainty, which is inherent in effort estimation, can outperform other commonly used benchmarking models, based on the mean or median effort. In addition, these results also suggest that Web companies that either volunteered projects to the Tukutuku database, or develop similar projects to those in that database, would benefit from using a Bayesian Network to obtain effort estimates, compared to simply relying on estimated based on the mean or median effort of past projects.

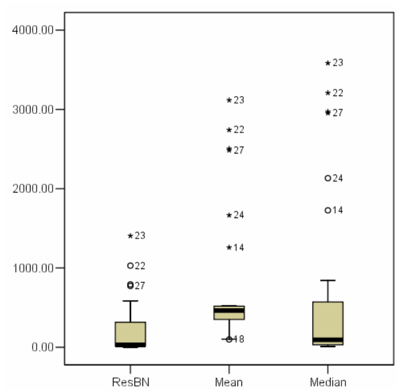


Fig. 5. Boxplots with distribution of residuals

The Web effort BN model presented in this paper is a very simple model, built using a dataset that does not represent a random sample of projects, therefore these results have to be interpreted with care. In addition, we chose to use only the nodes identified using the Tukutuku dataset, i.e., other nodes that could have been identified by the DE were not included. We also wanted to investigate to what extent a BN model and probabilities generated using automated algorithms available in HUGIN would provide predictions comparable to those obtained using mean and median models.

There are several issues regarding the validity of our results: i) the choice of discretisation, structure learning, parameter estimation algorithm, and the number of categories used in the discretisation all affect the results and there are no clear-cut guidelines on the best combination to use. This means that further investigation is paramount; ii) the Web effort BN presented in this study might have been quite different had it been entirely elicited from DEs, and this is part of our future work; iii) the decision as to what conditional independent nodes to retain was based on their strength of association with TotalEffort, however other solutions could have been used, e.g. ask a DE to decide; iv) obtaining feedback from more than one DE could also have influenced the BN structure in Fig. 3(b), and this is also part of our future work .

Finally, the use of BN tools by practitioners may still prove to be a challenge given that there are still many interface and technical issues that do not make their use straightforward.

4 Conclusions

This paper has presented the results of an investigation where a dataset containing data on 120 Web projects was used to build a Bayesian model, and the predictions obtained using this model were compared to those obtained using the mean and median effort models, based on a validation set with 30 projects.

The predictions obtained using the Web effort BN was significantly superior to the median-based and mean-based predictions, despite the use of a simple BN model. Future work entails: the building of a second Web effort BN based solely on domain experts' knowledge, to be compared to the BN presented in this paper; aggregation of this BN to our large Web resource BN, to obtain a complete causal model for Web resource estimation.

Acknowledgments. We would like to thank all the Web companies that volunteered the data used in this study. We would also like to thank Dr. N. Mosley for his comments on a previous version of this paper and Associate Professor Filomena Ferrucci for volunteering data on 15 projects to the Tukutuku project. This work is sponsored by the Royal Society of New Zealand, under the Marsden Fund research grant 06-UOA-201 MIS.

References

[1] Baresi, L., Morasca, S., Paolini, P.: An empirical study on the design effort for Web applications. In: Proceedings of WISE 2002, pp. 345–354 (2002)

[2] Baresi, L., Morasca, S., Paolini, P.: Estimating the design effort for Web applications. In: Proceedings of Metrics 2003, pp. 62–72 (2003)

[3] Christodoulou, S.P., Zafiris, P.A., Papatheodorou, T.S.: WWW2000: The Developer's view and a practitioner's approach to Web Engineering. In: Proc. Second ICSE Workshop on Web Engineering, 4 and 5 June 2000, Limerick, pp. 75–92 (2000)

[4] Conte, S.D., Dunsmore, H.E., Shen, V.Y.: Software Engineering Metrics and Models, Benjamin-Cummins (1986)
[5] Costagliola, G., Di Martino, S., Ferrucci, F., Gravino, C., Tortora, G., Vitiello, G.: Effort estimation modeling techniques: a case study for web applications. In: Procs. Intl. Conference on Web Engineering (ICWE'06), pp. 9-16 (2006)
[6] Druzdzel, M.J., Onisko, A., Schwartz, D., Dowling, J.N., Wasyluk, H.: Knowledge engineering for very large decision-analytic medical models. In: Proceedings of the 1999 Annual Meeting of the American Medical Informatics Association, pp. 1049–1054 (1999)
[7] Druzdzel, M.J., van der Gaag, L.C.: Building Probabilistic Networks: Where Do the Numbers Come From? IEEE Trans. on Knowledge and Data Engineering 12(4), 481–486 (2000)
[8] Fenton, N., Krause, P., Neil, M.: Software Measurement: Uncertainty and Causal Modeling, IEEE Software, pp. 116–122 (2002)
[9] Fenton, N., Marsh, W., Neil, M., Cates, P., Forey, S., Tailor, M.: Making Resource Decisions for Software Projects. In: Proc. ICSE'04, pp. 397-406 (2004)
[10] Fewster, R., Mendes, E.: Measurement, Prediction and Risk Analysis for Web Applications. In: Proceedings of IEEE Metrics Symposium, pp. 338–348 (2001)
[11] Jensen, F.V.: An introduction to Bayesian networks. UCL Press, London (1996)
[12] Kitchenham, B.A., Mendes, E.: A Comparison of Cross-company and Single-company Effort Estimation Models for Web Applications. In: Proceedings EASE 2004, pp. 47–55 (2004)
[13] Knobbe, A.J., Ho, E.K.Y.: Numbers in Multi-Relational Data Mining. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, Springer, Heidelberg (2005)
[14] Korb, K.B., Bayesian, A.E.N: Artificial Intelligence. CRC Press, USA (2004)
[15] Lauritzen, S.L.: The EM algorithm for graphical association models with missing data. Computational Statistics & Data Analysis 19, 191–201 (1995)
[16] Mahoney, S.M., Laskey, K.B.: Network Engineering for Complex Belief Networks. In: Proc. Twelfth Annual Conference on Uncertainty in Artificial Intelligence, pp. 389–396 (1996)
[17] Mangia, L., Paiano, R.: MMWA: A Software Sizing Model for Web Applications. In: Proc. Fourth International Conference on Web Information Systems Engineering, pp. 53–63 (2003)
[18] Mendes, E., Kitchenham, B.A.: Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications. In: Proc. IEEE Metrics, pp. 348–357 (2004)
[19] Mendes, E., Counsell, S.: Web Development Effort Estimation using Analogy. In: Proc. 2000 Australian Software Engineering Conference, pp. 203–212 (2000)
[20] Mendes, E., Mosley, N., Counsell, S.: Web Effort Estimation. In: Mendes, E., Mosley, N. (eds.) Web Engineering, pp. 29–73. Springer-Verlag, Heidelberg (2005)
[21] Mendes, E., Mosley, N.: Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications. In: Proc. ACM/IEEE ISESE, Nara, Japan, pp. 79–90 (2002)
[22] Mendes, E., Mosley, N., Counsell, S.: Investigating Web Size Metrics for Early Web Cost Estimation. Journal of Systems and Software 77(2), 157–172 (2005)
[23] Mendes, E., Mosley, N., Counsell, S.: A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation. In: Proc. ISESE, pp. 100–109 (2003)
[24] Mendes, E., Mosley, N., Counsell, S.: Early Web Size Measures and Effort Prediction for Web Costimation. In: Proceedings of the IEEE Metrics Symposium, pp. 18–29 (2003)
[25] Mendes, E., Mosley, N., Counsell, S.: Comparison of Length, complexity and functionality as size measures for predicting Web design and authoring effort. In: IEE Proc. Software 149(3), 86–92 (2002)

[26] Mendes, E., Mosley, N., Counsell, S.: Web metrics - Metrics for estimating effort to design and author Web applications. IEEE MultiMedia, pp. 50–57 (January-March, 2001)
[27] Mendes, E., Mosley, N., Watson, I.: A Comparison of Case-Based reasoning Approaches to Web Hypermedia Project Cost Estimation. In: Proc. WWW'02 (2002)
[28] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S.: A Comparative Study of Cost Estimation Models for Web Hypermedia Applications. ESE 8(2), 163–196 (2003)
[29] Neil, M., Fenton, N., Nielsen, L.: "Building Large-scale bayesian networks", The knowledge Engineering Review. KER 15(3), 257–284 (2000)
[30] Neil, M., Tailor, M., Marquez, D., Fenton, N., Hearty, P.: Modeling Dependable Systems using Hybrid Bayesian Networks. In: Proc. BND Workshop, pp. 817–823 (2006)
[31] Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
[32] Pendharkar, P.C., Subramanian, G.H., Rodger, J.A.: A Probabilistic Model for Predicting Software Development Effort. IEEE Trans. Software Eng. 31(7), 615–624 (2005)
[33] Reifer, D.J.: Web Development: Estimating Quick-to-Market Software. IEEE Software, 57–64 (2000)
[34] Reifer, D.J.: Ten deadly risks in Internet and intranet software development. IEEE Software, 12–14 (2002)
[35] Ruhe, M., Jeffery, R., Wieczorek, I.: Cost estimation for Web applications. In: Proceedings ICSE 2003, pp. 285–294 (2003)
[36] Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, Sydney (1986)
[37] Stamelos, I., Angelis, L., Dimou, P., Sakellaris, E.: On the use of Bayesian belief networks for the prediction of software productivity. Information and Software Technology 45(1), 51–60 (2003)
[38] Steck, H., Tresp, V.: Bayesian Belief Networks for Data Mining. In: Proceedings of The 2nd Workshop on Data Mining und Data Warehousing, Sammelband (September 1999)
[39] Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data & Knowledge Engineering 25, 161–197 (1998)
[40] Wong, A.K.C., Chiu, D.K.Y.: Synthesizing Statistical Knowledge from Incomplete Mixed-mode Data. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9(6), pp. 796–805 (1987)
[41] Woodberry, O., Nicholson, A., Korb, K., Pollino, C.: Parameterising Bayesian Networks. In: Proc. Australian Conference on Artificial Intelligence, pp. 1101–1107 (2004)