Proceedings of the 2010 IEEE
International Conference on Information and Automation
June 20 - 23, Harbin, China

Assessing the Weighted Sum Algorithm for Automatic Generation of Probabilities in Bayesian Networks

Simon Baker
The University of Auckland
Computer Science Department
Private Bag 92019, Auckland, New Zealand
sbak030@aucklanduni.ac.nz

Emilia Mendes
The University of Auckland
Computer Science Department
Private Bag 92019, Auckland, New Zealand
emilia@cs.auckland.ac.nz

***Abstract* — A Bayesian Network (BN) is a probabilistic reasoning technique, which to date has been used in a broad range of applications. One of the key challenges in constructing a BN is obtaining its Conditional Probability Tables (CPTs). CPTs can be learnt from data (when available), elicited from domain experts, or a combination of both. Eliciting from domain experts provides more flexibility; however, CPTs grow in size exponentially, thus making their elicitation process very time consuming and costly. Previous work proposed a solution to this problem using the weighted sum algorithm (WSA) [9]; however no empirical results were given on the algorithm's elicitation reduction and prediction accuracy. Hence the aim of this paper is to present two empirical studies that assess the WSA's efficiency and prediction accuracy. Our results show that the estimates obtained using the WSA were highly accurate and make significant reductions in elicitation.**

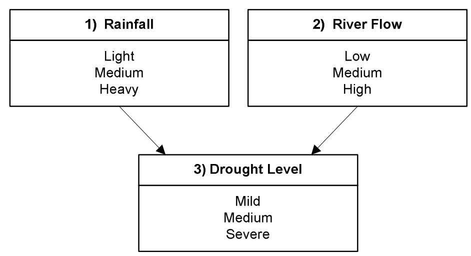
Bayesian Network; Conditional Probability; Weighted Sum Algorithm; Knowledge Elicitation, CPT Elicitation, Empirical study

I. INTRODUCTION

A Bayesian Network (BN) (also known as a Bayesian Belief Network) is a probabilistic modelling technique that allows for reasoning under uncertainty. BNs have been applied in many areas including: forecasting, estimation, classification, recognition, and inference [1, 2]. A BN consists of two components: The first is a Direct Acyclic Graph (DAG) that represents factors of interest (as nodes) and associated causal relations (as edges). The second component is a set of Conditional Probability Tables (CPTs), one for each node. For example, Figure 1 shows a naive BN for predicting drought level for a hypothetical location. The BN consists of three nodes two can be considered as causes (Rainfall and River Flow), and an effect node: Drought Level.

The left side of the Drought Level CPT lists all possible configurations of its parents' states, while on the left side it holds the corresponding probability values for each configuration, for example, the right most cell in the first row is the probability that the Drought Level is Severe given that rainfall and river flow are both low. Every row in the child nodes' CPT must contain values that sum to exactly one.

A BN's DAG and CPTs can be automatically learnt from data [3], or by domain expert elicitation [4, 5], where experts knowledgeable in a given domain are asked to visualize the relations and factors that constitute the causal structure of the model, as well as the probability values required for the CPTs, or using a combination of both. Domain expert elicitation is often favoured when there is no data available for automatic learning, or when the model is qualitative in nature, and hence, difficult to record measurable data [5, 6].



Rainfall CPT	
Low	0.25
Med	0.50
High	0.25

River Flow CPT	
Low	0.30
Med	0.55
High	0.15

Drought Level CPT				
Rainfall	***River Flow***	***Mild***	***Med***	***Severe***
Low	**Low**	0.10	0.10	0.80
Low	**Med**	0.25	0.45	0.30
Low	**High**	0.20	0.50	0.30
Med	**Low**	0.50	0.25	0.25
Med	**Med**	0.25	0.60	0.15
Med	**High**	0.20	0.50	0.30
High	**Low**	0.30	0.50	0.20
High	**Med**	0.45	0.45	0.10
High	**High**	0.85	0.15	0.0

Figure 1. : Example Baysian Network with its CPTs

Although domain expert elicitation is seen as providing a more pragmatic approach, the reality is that eliciting thousands of probabilities manually becomes unmanageable, because this process becomes extremely laborious and time consuming. CPTs grow exponentially with respect to the number of parental states, for example, if we add an additional state to each parent in the BN shown in Figure 1, then the child node (Drought Level) CPT size would increase from 27 to 48

978-1-4244-5704-5/10/$26.00 ©2010 IEEE

parameters. Therefore, even a seemingly small network (e.g. under 20 nodes) can potentially contain very large CPTs.

II. Motivation

Eliciting probability values for CPTs is one of the chief concerns for Bayesian Network practitioners [5, 6]. During the last three decades a few techniques were proposed to help mitigate this problem. Pearl's Noisy-OR Gate technique [7] is perhaps one of the more established techniques, followed by its generalization (Noisy-MAX Gate) [7, 8]. However, these techniques often make fundamental assumptions about the BN: they assume that parent nodes must be independent, and that each node must have an "absent" state. In many domains, however, these assumptions cannot be satisfied. To remedy many of these constraints Das [9] proposed a technique known as the weighted sum algorithm (WSA). However, unlike the Noisy-OR and Noisy-MAX techniques, there are no empirical studies, as far as we know that assess the WSA's efficiency and estimation accuracy.

Therefore, the aim of this paper is to empirically assess the WSA's efficiency and estimation accuracy using as benchmark manual domain expert elicitation. The main contribution of this paper is to provide empirical evidence on the use of the WSA for CPT probability generation.

The remainder of this paper is organised as follows: The next Section introduces the WSA technique, followed by a description of the methodology used in the empirical assessment, our results and threats to their validity, and finally our conclusions and comments on future work.

III. The Weighted Sum Algorithim

The WSA is based on two heuristics originally proposed by Kahnman and Tversky [10, 11]. The first states that the more cognitively accessible an event is, the more likely it is perceived to occur (known as the availability heuristic). The second heuristic is to mentally simulate a scenario to assess the ease with which different results are produced given an initial set of parameters and operating constraints (known as the simulation heuristic). Using as basis these two heuristics, Das introduces the notion of compatible parental configuration [9], to be used to elicit the more cognitively accessible scenarios from the experts, and later to generate the remaining CPT using a weighted sum calculation. To describe the weighted sum algorithm we will refer to the BN in Figure 2.

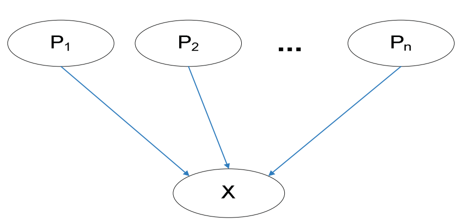


Figure 2. Generic naive Bayesian Network

We represent a node's individual states in superscript, so $\{x^1, x^2, \dots x^n\}$ are all the states of the child node x; for the parent nodes, we'll use numbers in subscript to distinguish between different parents. For example $\{p_1^1,\ p_1^2, \dots\ p_1^n\}$ are all the states of the first parent P_1.

To formally define the notion of parental compatibility, let the parent P_i be assigned an arbitrary state p_i^v i.e. $P_i = p_i^v$, and let P_j be another parent, such that P_j is considered compatible with $P_i = p_i^v$ only when P_j is in some state (p_j^w) that is most likely, according to the expert's knowledge, to coexist with $(P_i = p_i^v)$. Therefore, we will use the notation $Comp[P_i = p_i^v]$ to represent the set of states that are compatible with $P_i = p_i^v$ for all parents.

$$Comp[P_i = p_i^v] = \left\{ p_j^w, \forall j \neq i \;\middle|\; \max_{w=1 \dots |P_j|} \mathrm{P}\left(p_j^w \middle| p_i^v\right) \right\} \tag{1}$$

The expert is asked to elicit the probabilities of compatible parental configurations for each state in each parent node (whenever the combination of states is meaningful (i.e. compatible) to the expert), and then the WSA algorithm is used to generate the probabilities of a child node's CPT. For example, if we refer to the BN in Figure 1, the expert might find it easier to estimate the probabilities for the drought levels for the case when the river flow is low, which would likely coincides when the rainfall is also low, so therefore $Comp[RiverFlow = low]$ would contain the parental state $Rainfall = low$, i.e. $Comp[P_2 = p_2^1] = \{p_1^1\}$ in this example.

Therefore, what makes this algorithm unique, compared to other techniques, is that it focuses on asking experts questions that are easy to visualize and simulate because they relate to more realistic probabilities. Once this is done, then the algorithm generates the remaining probabilities that are likely harder to estimate. However, an important assumption must be made when applying this algorithm: the expert must be capable of identifying realistic compatible parental configurations of each state for each parent. If the expert fails to do so, the algorithm will generate less trustworthy probabilities.

The algorithm takes as input:

1. The conditional probabilities corresponding to the compatible parental configurations for every parental state.
2. A relative weight value (between zero and one) for each parent node, denoting the degree of influence a parent has on a child node. The relative weights for all parent nodes must add up to exactly one. A relative weight equal to zero means that a parent has no influence at all over a child node, and can therefore be omitted from the network; conversely, a relative weight equal to one indicates that a parent node is the only determinant of the conditional probabilities in the child node.

The above inputs are then used as weighted sum corresponding to each state configuration in the child node's CPT, as follows:

$$P(X = x^i | p_1^a, \quad p_2^b, \quad p_3^c, \dots \ p_n^z) =$$
$$w_1 P(X = x^i | Comp[P_1 = p_1^a]) + w_2 P(X = x^i | Comp[P_2 = p_2^b])$$
$$+ w_3 P(X = x^i | Comp[P_3 = p_3^c]) \dots + w_n P(X = x^i | Comp[P_n = p_n^z]) \quad (2)$$
$$for\ i = 1 \dots |x|$$

Where w is the relative weight value for a parent, e.g. w_1 is the relative weight value for P_1.

To illustrate it's usage, we refer back to our example in Figure 1, and we intended to calculate an arbitrary cell in the CPT that has not been elicited, for instance, the probability that the drought level is severe given that the rainfall is low and river flow is medium. Assuming that both parents are equally influential (i.e. both have an equal relative weight value of 0.5), then the weighted sum calculation for this cell is as follows:

$$\begin{aligned} &P(DroughtLevel = Severe \mid Rainfall = low, RiverFlow = Medium) \\ &= 0.5 \times P(DroughtLevel = Severe \mid Comp[Rainfall = low]) \\ &+ 0.5 \times P(DroughtLevel = Severe \mid Comp[RiverFlow = Medium]) \end{aligned} \quad (3)$$

If different parents' states share the same compatible parental configuration it is possible to reduce the number of elicitation questions due to this overlap. In addition, it also makes the computation of the algorithm more efficient; however, in most cases, this computation will be negligible. For example, if there are 5 parent nodes each with 3 states, but none of the states share any parental configuration, then 15 elicitation questions are needed to satisfy the first input of the algorithm, whereas if we take the opposite extreme, when there is an optimum compatible parental configuration overlap, only 3 questions are required. Although such optimum overlap is unlikely to be frequent, it is still beneficial to detect overlaps in order to reduce the elicitation effort.

There are situations when the domain expert is unable to decide on exactly one compatible parental configuration for a given parental state assignment, and there might be a set of compatible parental configurations for that given state. Das does not provide a solution for such situations. However, during our experimentation with this method, we proposed a possible solution for this problem. We suggested an extension of the algorithm [12] such that if a part of the given CPT configuration can be matched with a compatible parental configuration, then this value would be used (which is the default situation presented in Equation 2); otherwise, an average of all valid compatible parental configurations' probabilities would be used. Our proposal was later discussed with the WSA's author, who agreed with our suggestion.

To state the proposed extension formally, let $\Omega[P_i = p_i^v]$ be a subset of $Comp[P_i = p_i^v]$ such that it contains all of the compatible states that share the same parent, in other words, it is the set of states that have at least one other state from the same parent that is also compatible with $P_i = p_i^v$, i.e.:

$$\begin{aligned} &\Omega[P_i = p_i^v] = \\ &\{\forall\, p_j^a \in Comp[P_i = p_i^v] \mid \exists\, p_j^b \in Comp[P_i = p_i^v] : a \neq b\} \end{aligned} \quad (4)$$

And let ω be any subset of $\Omega[P_i = p_i^v]$ such that each state has a unique parent (i.e. set of states that do not share the parent with another state). We can define ω as follows:

$$\omega = \{S \subset \Omega[P_i = p_i^v], S \neq \emptyset \mid \forall p_j^m, p_k^n \in S : j \neq k\} \quad (5)$$

We therefore extend the original notation for representing the set of compatible parental configurations for the parental assignment $P_i = p_i^v$ (Equation 1) to be conditional on ω; we define this conditional parental configuration as the following:

$$Comp\left[\frac{P_i = p_i^v}{\omega}\right] = \left\{p_j^w \,\middle|\, \max_{w = 1 \dots |P_j|} P\left(p_j^w \mid p_i^v, \omega\right)\right\} \quad (6)$$

which is the set of all states that are compatible with $P_i = p_i^v$ and all of the compatible parental assignments represented by ω, Using this definition, we can extend the Weighted Sum Algorithm (Equation 2) to handle all situations when there are more than one compatible parental configurations for a given parental assignment, as follows:

$$P(X = x^i | Config) = \sum_{p_i^v \in Config} w_i\ f(p_i^v, Con) \quad (7)$$

where $Config$ is a given configuration in the child CPT that we are generating the probability for, w_i is the weight associated with parent i and $f(p_i^v, Config)$ is a conditional function defined as follows:

$$f(p_i^v, Config) = \begin{cases} P\left(X = x^c \middle| Comp\left[\frac{P_i = p_i^v}{Config}\right]\right) & if\ Config \subseteq \Omega[P_i = p_i^v] \\ \frac{\sum_{\omega \in \Omega[P_i = p_i^v]} P\left(X = x^c \middle| Comp\left[\frac{P_i = p_i^v}{\omega}\right]\right)}{|\Omega[P_i = p_i^v]|} & otherwise \end{cases} \quad (8)$$
$$for\ c = 1 \dots |x|$$

In other words, if a part of the given configuration can be matched with a compatible parental configuration, then its elicited probabilities would be used, otherwise, an average of the closest (in similarity) compatible parental configuration probabilities would be used.

Note that our solution greatly increases the number of probabilities elicited using the WSA method because for each additional parental state that is considered to be compatible for a given parent, the number of elicited probabilities using this method grows exponentially, which may approach manual elicitation.

IV. METHODOLOGY

The BNs used in our empirical assessment of the WSA method were elicited from experts in software and Web development. These BNs were built to forecast development costs for Web companies. Two Web effort estimation BNs were used, thus leading to two separate case studies. The following factors were fixed in each of these studies:

- A single domain expert participated in eliciting all required information. By fixing the domain expert we eliminated any potential discrepancies that could be introduced by multiple domain experts.
- An expert-driven BN model constructed using only manual elicitation was used as benchmark because it had already been validated using real data and it was confirmed by the domain expert(s) to fully reflect their

beliefs. Therefore, the benchmark model was used as basis to measure the accuracy of the WSA algorithm.

- The input probabilities for the WSA algorithm were taken directly from the benchmark BN, and were not elicited again. This was done to avoid discrepancies in probabilities used in the models being compared (i.e. if probabilities were elicited twice, once using manual elicitation and once using the WSA algorithm, the expert could erroneously provide different probability values for exactly the same cell in the CPT). Therefore, the only inputs that were elicited for the WSA algorithm were the selection of the compatible parental configurations, and the weights for each parent.

We evaluated the weighted sum algorithm by:

- Determining reductions in elicited probablities that it achieves. This is done by simply counting the required input probablities required by the algorithm for a given CPT and then comparing it with the CPT size (i.e. what would be required if one would use manual elicitation).
- Measuring the accuracy of the algorthim by calculating the absolute error (Euclidean distance) between every cell in the generated CPT and the reference CPT in the benhmark BN model. Here we used the same technique previousl used to evaluate CPT generation techniques (e.g. [13]). The Wilcoxon signed ranked test was employed to check the statistical significance of the results ($\alpha = 0.05$). A non-parametric test was chosen because we could not guarantee that the data from some of the CPTs in the benchmark BN were normally distributed.

V. RESULTS

In this section we summarise the results for the two case studies conducted.

Case Study 1

The benchmark model contains 15 nodes, where four were considered to be nontrivial in terms of probability parameters (highlighted in Figure 3) and were therefore used as reference CPTs. A CPT is considered as nontrivial whenever it has at least two parent nodes, each presenting at least two states [12, 14, 15]. The sizes of the four CPTs are listed in Table I.

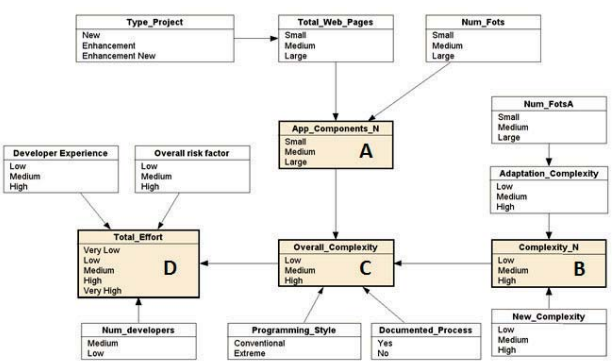


Figure 3. Benchmark Bayesian Network model for case study 1

TABLE I. THE SIZE OF CPTS IN CASE STUDY 1

CPT	Size	Number of Parents
A	27	2
B	27	2
C	108	4
D	270	4

The domain expert was confident in identifying the compatible parental configurations for CPTs A, and B. However this was not the case for the remaining CPTs, given that the expert was unable to select exactly one compatible parental configuration for most parent states in CPT C, and to a lesser extent in CPT D. we therefore applied our proposed extension (described in Section III). We elicited all computable parental configurations that the expert could identify for a given parental state, and later averaged their probability values. The percentage reduction in probability elicitation for each of the four CPTs (see Figure 4) shows that CPT C achieved only 33.33% reduction in elicitation due to the expert's indecisiveness in selecting compatible parental configurations.

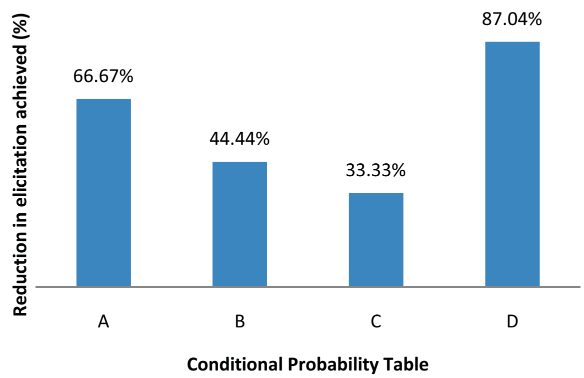


Figure 4. Elicitation reductions achieved in case study 1

With regards to accuracy, Figure 5 illustrates the distribution of absolute error values for each generated CPT. CPT C achieved nearly perfect accuracy (most values presenting an absolute error equal to zero), which is also due to the proposed extension of the algorithm, where higher accuracy is achieved due to further elicitation of probabilities. With the exception of CPT D, all median absolute error values were under 0.05, i.e. median accuracy of 95%, which is a low error rate in comparison to other studies assessing other CPT generation techniques [13, 16, 17].

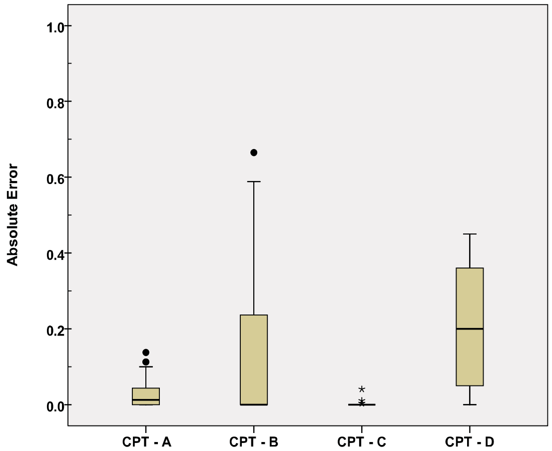


Figure 5. Accuracy results for case study 1

Despite the abovementioned results, no statistically significant differences in accuracy were observed between the two models being compared.

Case Study 2

The benchmark model used in this study contained 16 nodes; seven were considered nontrivial in terms of probability parameters (highlighted in Figure 6) and were used as reference CPTs. Table 2 lists the seven reference CPT sizes.

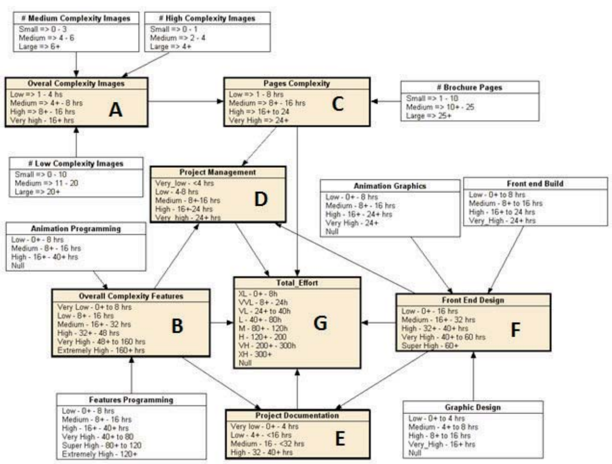


Figure 6. Benchmark Bayesian Network model for case study 2

TABLE II. THE SIZE OF CPTS IN CASE STUDY 2

CPT	Size	Number of Parents
A	108	3
B	144	2
C	48	2
D	600	3
E	120	2
F	500	3
G	21,600	5

During this case study we chose not to apply our proposed extension to the WSA algorithm in order to be able to assess the weighted sum algorithm as it was originally proposed; thus, whenever the domain expert was encountering difficulties in selecting a single compatible parental configuration from a set of valid configurations, they were asked to arbitrarily select any configuration from the valid set.

A significant reduction in elicitation effort was achieved in case study 2, specifically in CPT G (the largest CPT used in both case studies), where only 72 probabilities were needed by the WSA algorithm, compared to 21,600 probabilities obtained using manual elicitation (see Figure 7). The reductions achieved in this case study clearly show the linear asymptotic growth that the WSA algorithm can achieve compared to the exponential growth of manual elicitation.

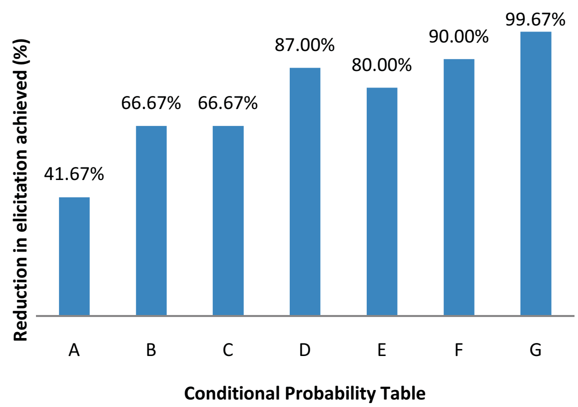


Figure 7. Elicitation reduction reduction achieved in case study 2

With regards to the accuracy, one can observe that, with the exception of CPT D and G, the median absolute error was under 0.1, with CPT-B being the most accurate while CPT D the least (shown in Figure 8). Although the accuracy results in case study 2 were slightly lower than those for case study 1, it still presents relatively more accurate results when compared to other techniques (e.g. interpolation, and noisy gates).

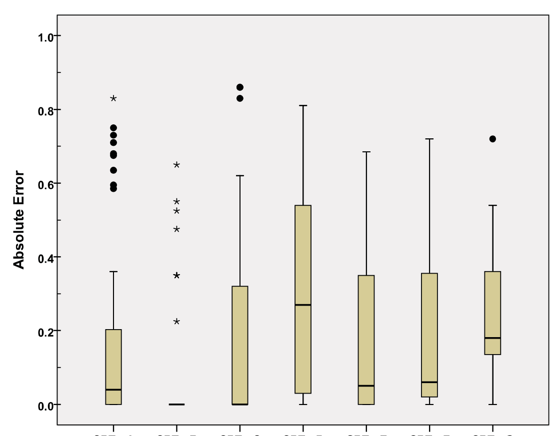


Figure 8. Accuracy results for case study 2

These accuracy results also suggest that even though the expert was required to select exactly one compatible parental configuration for each parental state, the results achieved were not significantly more accurate than those for case study 1. This implies that our proposed extension to the WSA algorithm did not seem to significantly increase the accuracy whenever a domain expert could not select exactly one parental configuration. This was perhaps unexpected, as one would assume additional elicited information would result in much higher accuracy.

TABLE III.
WILCOXON SIGNED RANKED TEST RESULTS FOR CASE STUDY 2

CPT	Wilcoxon Z-value
A	-0.409
B	-0.687
C	-0.213
D	**-2.441**
E	-0.552
F	**-1.986**
G	**-22.404**

Table III shows the seven Wilcoxon signed ranked test results. Three of which (emphasized in bold) fall outside the ±1.96 range, implying statistically significant differences in the obtained estimates. Note that the three CPTs (D,F, and G) that have resulted in statistically significant differences all have sizes greater or equal to 500, whereas the remaining CPTs are much smaller in size (as described earlier in Table II). Thus, the results in this case study suggest that large CPT estimates tend to yield statistically significant differences compared to probability parameters obtained using manual elicitation.

VI. THREATS TO VALIDITY

Although we believe that the results of the two case studies described herein provide useful empirical evidence regarding the benefits of the WSA algorithm, there are threats to the validity of these results that must be taken into account:

The first threat relates to the number of experts who participated in each case study, given that a greater number of experts could perhaps increase the level of certainty relating to identifying compatible parental configuration for each parental state. Clearly further investigation is needed in order to confirm whether the accuracy changes whenever the number of experts increases.

Another threat relates to the size of the BN models employed in both case studies. Although both BN models did not seem to be small, when compared to the size of other BN models in different areas (e.g. ecology [20] and medicine[21]), further investigation using large and very large BN models is also needed to confirm/refute the patterns observed herein.

Another related point is the applicability of the WSA to a particular domain. As stated previously, both cases studies were based in the Web engineering domain. Certain intricacies specific to this domain might affect the performance of the WSA, for example, it might be more difficult to identify parental compatible configurations in other domains compared to the one used herein.

VII. CONCLUSIONS

Eliciting probability parameters for Conditional Probability Tables is one of the major impediments for Bayesian Network practitioners [5, 6]. CPT sizes grow exponentially relative to the number of parental states. Thus, a seemingly undersized Bayesian Network model can still contain very large CPTs. Therefore, abating this exponential growth is a key requisite in order to construct larger and more comprehensive Bayesian Networks.

During the last three decades few techniques have been pioneered to help ease the problem of eliciting large CPTs such as Noisy gates [7, 8], and linear and piecewise interpolation techniques [17]. However, most of these techniques require certain assumptions to be made about the BN model at hand, for example, they often require that parent nodes variables to be independent, or they constrain the type of states they can include, or simply make assumptions about the underlying probability distributions. These constraints can be a serious obstacle in applying these techniques. The weighted sum algorithm makes no such assumptions and therefore can be applied in broader environments and domains.

The weighted sum algorithm relies on the notion of compatible parental configurations in order to elicit probabilities for scenarios that are easier to recall by the expert, and hence likely to be easier to elicit and be more reliable. The algorithm utilizes a very simple calculation based on elicited parental weights, and a weighted sum. However, the original algorithm proposed by Das does not describe how to deal with situations where the expert cannot select a single compatible parental configuration for a given parental state. Therefore, we proposed an extension for the algorithm that remedies this situation by averaging the probabilities of valid compatible parental configurations that expert might select.

The objective of our work was therefore to empirically assess the weighted sum algorithm in terms of elicitation reduction and accuracy. Another secondary goal was to see how effective our proposed extension to algorithm was.

We assessed the weighted sum algorithm in two case studies. The first case study consisted of four nontrivial CPTs, two CPTs to which our proposed extension to the original WSA algorithm was applied to. The outcome of the first case study showed that overall the WSA algorithm achieved a significant reduction in the effort eliciting probabilities, and achieved very high accuracy, with a median absolute error value less than 0.05 for three of the four CPTs.

The second case study used a larger benchmark BN model with seven nontrivial CPTs; however, in this case study, the proposed extension to the algorithm was not applied, and instead, the expert was asked to arbitrarily select a compatible parental configuration in the event that there is more than one valid configuration compatible with a given prenatal state.

The results of the second case study showed slightly less accurate results, however, still very much comparable to that of the first case study, perhaps suggesting that our proposed extension might yield slightly better accuracy, but not necessarily worth the tradeoff of eliciting more probabilities. The elicitation reduction results in the second case study were even more significant than the first. With the largest CPT of (21,600 parameters) being reduced by 99.67% to only 72 parameters, which illustrates the linear growth of the input probabilities required by the algorithm, compared to the exponential growth of manual elicitation. The Wilcoxon signed ranked test results also suggest that the algorithm only yields statistically significant results when the CPTs are large (on the scale of 500 or more parameters).

In conclusion, the weighted sum algorithm has demonstrated to significantly mitigate the CPT elicitation burden in both case studies, while yielding high accuracy levels. The algorithm's simplicity and lack of constraints it assumes makes it an attractive technique for solving the CPT elicitation burden, and should be considered by BN practitioners.

In our future work, we will further investigate the weighted sum algorithm, and directly compare it with other CPT generation techniques such as interpolation.

ACKNOWLEDGEMENTS

We like to thank all the participating companies and experts in this research for donating their in-kind time and expertise This work was sponsored by the Royal Society of New Zealand (Marsden research grant 06-UOA-201).

REFERENCES

[1] Stephenson, T.A., An Introduction to Bayesian Network Theory and Usage. 2000: bIDIAPRR.

[2] Kjærulff, U.B. and A.L. Madsen, Probabilistic Networks—An Introduction to Bayesian Networks and Influence Diagrams. Aalborg University, 2005.

[3] Neapolitan, R.E. Learning Bayesian networks. in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007: ACM New York, NY, USA.

[4] Mendes, E., C. Polino, and N. Mosley, Building an Expert-based Web Effort Estimation Model using Bayesian Networks. 13th International Conference on Evaluation & Assessment in Software Engineering, 2009.

[5] Hu, X.-x., H. Wang, and S. Wang, Using Expert's Knowledge to Build Bayesian Networks. Proceedings of the 2007 International Conference on Computational Intelligence and Security Workshops, 2007(1333212): p. 220-223.

[6] van der Gaag, L.C., et al., How to elicit many probabilities, in Uncertainty in Artificial Intelligence. 1999, Morgan Kaufmann. p. 647--654.

[7] Pearl, J., Probabilistic reasoning in intelligent systems: networks of plausible inference. 1988: Morgan Kaufmann.

[8] Heckerman, D. and J.S. Breese, Causal independence for probability assessment and inference usingBayesian networks. IEEE Transactions on Systems, Man and Cybernetics, Part A, 1996. 26(6): p. 826-831.

[9] Das, B., Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem. CoRR, 2004. cs.AI/0411034.

[10] Tversky, A. and D. Kahneman, Judgment under uncertainty: Heuristics and biases. Judgment under Uncertainty: Hureistics and Biases, 1982: p. 3-20.

[11] Tversky, A. and D. Kahneman, Availablity: A heuristic for judging frequency and probability. Judgment under Uncertainty: Hureistics and Biases, 1982: p. 163-178.

[12] Baker, S., Towards the Construction of Large Bayesian Networks for Web Cost Estimation, in Department of Computer Science. 2009, University of Auckland: Auckland.

[13] Zagorecki, A. and M. Druzdzel. An empirical study of probability elicitation under Noisy-OR assumption. in Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS. 2004).

[14] Jensen, F.V., An Introduction to Bayesian Networks. 1996, London: UCL Press.

[15] Woodberry, O., et al. Parameterising Bayesian Networks. in Australian Conference on Artificial Intelligence. 2004.

[16] Onisko, A., M.J. Druzdzel, and H. Wasyluk. An experimental comparison of methods for handling incomplete data in learning parameters of Bayesian networks. 2002: Physica Verlag.

[17] Zhong, T. and M. Brenda, Developing Complete Conditional Probability Tables from Fractional Data for Bayesian Belief Networks. Journal of Computing in Civil Engineering, 2007. 21(4): p. 265-276.

[18] Mendes, E., The Use of Bayesian Networks for Web Effort Estimation: Further Investigation. Proceedings of the 2008 Eighth International Conference on Web Engineering 2008(1441569): p. 203-216.

[19] Mendes, E. and N. Mosley, Bayesian Network Models for Web Effort Prediction: A Comparative Study. Software Engineering, IEEE Transactions on, 2008. 34(6): p. 723-737.

[20] McCann, R.K., B.G. Marcot, and R. Ellis, Bayesian belief networks: applications in ecology and natural resource management. Canadian Journal of Forest Research, 2006. 36(12): p. 3053-3062.

[21] Twardy, C.R., A.E. Nicholson, and K.B. Korb, Knowledge engineering cardiovascular Bayesian networks from the literature.