Improving the Applicability of the Ranked Nodes Method to build Expert-Driven Bayesian Networks

João Nunes*[1], Luiz Silva†[1], Mirko Perkusich‡[1], Kyller Gorgonio§[1], Hyggo Almeida¶[1], and Angelo Perkusich‖[1]

[1]Embedded and Pervasive Computing Laboratory, Federal University of Campina Grande, Campina Grande, Brazil

Abstract

One challenge in constructing a Bayesian network (BN) is defining the node probability tables (NPTs), which can be learned from data or elicited from domain experts. In practice, for large-scale BN it is common not to have enough data for learning and elicitation from experts is unfeasible. Previous work proposed a solution to this problem: the Ranked Nodes Method (RNM). However, this solution needs to be applied by a RNM expert who, through the elicitation of expert judgement, identifies the necessary parameters for the RNM algorithm to generate the NPTs. Hence, this paper presents a novel approach to define NPT using the RNM with no ranked nodes-specific knowledge. The solution is named Simulated Bayesian Network Expert (SBNE). It consists of eliciting a subset of the NPT from the domain experts which is used as input to an algorithm that estimates the optimal parameters for the RNM to generate the NPTs. To validate our solution, we conducted an experiment with multiple domain experts and compared the results with other methods. Our solution outperformed the other methods (producing NPTs at least 12% more accurate) and is, therefore, a promising approach to apply RNM without relying on RNM experts.

Bayesian networks; expert systems; knowledge acquisition; ranked nodes.

*joao.bezerra@embedded.ufcg.edu.br
†luiz.silva@embedded.ufcg.edu.br
‡mirko.perkusich@embedded.ufcg.edu.br
§kyller@embedded.ufcg.edu.br
¶hyggo@embedded.ufcg.edu.br
‖perkusic@embedded.ufcg.edu.br

1 Introduction

Bayesian Network (BN) is a mathematical model that graphically and numerically represents the probabilistic relationships between random variables through Bayes theorem. Recently, given the evolution of the computational capacity, which enabled the calculation of complex BNs, it has become a popular technique to assist on decision-making [7] and it has been applied in several areas such as large-scale engineering projects [12], software engineering [16, 14], and sports management [4].

The challenges for the construction of BNs can be divided into two sub-problems: (i) construct the directed acyclic graph (DAG) and (ii) define the NPTs. In this research, we focus on (ii). In cases where there is historical data with enough information about the domain to be modelled it is possible to automate the process of NPT definition through batch learning [10].

Unfortunately, in practice, in most cases, there is not enough data [7] to apply batch learning. In such cases, it is necessary to manually define the NPTs through the elicitation of domain experts knowledge. However, given that the complexity of building NPTs grows exponentially, depending on the number of parents and states, the manual definition of the NPT becomes unfeasible.

To reduce the complexity of manually defining a NPT through the elicitation of knowledge from domain experts, Fenton et al. [7] proposed the Ranked Nodes Method (RNM). This method is limited to nodes (i.e., random variables) with an ordinal scale (e.g., “Good”, “Medium”, “Bad”), which are called ranked nodes.

In ranked nodes, the ordinal scale is mapped into a scale monotonically ordered in the interval $[0, 1]$. The solution is based on a Normal distribution truncated between $[0, 1]$

DOI reference number: 10.18293/SEKE2019-190

(i.e., TNormal) to represent the NPTs. Hence, the NPT of a child node is a TNormal calculated as the mixture of the TNormals of its parent nodes. There are four expressions to model the mixture's mean (μ): weighted mean (*WMEAN*), weighted minimum (*WMIN*), weighted maximum (*WMAX*) and the mixture of the classic minimum and maximum functions (*MIXMINMAX*).

Hence, to properly use the RNM it is necessary to understand the mixture process to select the appropriate parameters: the weighted expression; a set of weights of the parent nodes; and the variance (σ). However, even when RNM experts are available, the application of the RNM method still presents challenges.

The means to identify a suitable expression to mix the TNormals of the parent nodes based on mode assessments of the domain experts is straightforward, as described in Laitila and Virtanen [11]. Conversely, the discovery of the weights and variance parameters are far more complex. Such tasks are usually performed by the RNM experts using a trial and error strategy. This strategy comes down to a cycle of generating, verifying and adjusting the parameters to regenerate the NPTs, which is repeated until a satisfying result is discovered [7, 11].

In this paper, we present a novel approach to improve the applicability of RNM. Our main goal is to encapsulate its complexity, allowing for its use by domain experts with no prior knowledge about ranked nodes. We use "what if" analysis (i.e., truth tables) to elicit knowledge from experts using visual aids. Given the information collected, we use the expert system proposed in Silva et al. [5] to obtain the weighted expression to mixture the TNormals and an algorithm named "Simulated Expert" to estimate the optimal variance and set of weights of the parent nodes.

We evaluated our solution with an experiment in which multiple domain experts applied our approach to RNM and two other methods to quantify a BN model related to the evaluation of cohesion of agile software development teams. The NPTs generated with the three methods were compared in terms of accuracy using manually defined NPTs as benchmark. The results showed that our solution is promising as it has achieved greater accuracy compared to the other methods.

2 Background

BNs are probabilistic graph models used to represent knowledge about uncertain domains. A BN, B, is a directed acyclic graph that represents a joint probability distribution over a set of random variables V [9]. The network is defined by the pair $B = \{G, \Theta\}$. G is the directed acyclic graph in which the nodes $X_1, \ldots, X_n$ represent random variables and the arcs represent the direct dependencies between these variables. Θ represents the set of probability functions. This set contains the parameter $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$ for each x_i in X_i conditioned by π_i, the set of parents of X_i in G. Equation 1 presents the joint distribution defined by B over V.

Table 1. Example of a truth table.

Parent A	Parent B	Parent C	Child D
Very low	Very high	Very low	Low
Very high	Very low	Very low	Low
Very low	Very low	Very high	Low
Very low	Very high	Very high	High
Very high	Very low	Very high	Low
Very high	Very high	Very low	Medium

$$P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} P_B(x_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi i} \qquad (1)$$

We present an example of a BN in Fig. 1, in which ellipses represent the nodes and arrows represent the arcs. The probability functions are usually represented by NPTs. Even though the arcs represent the causal connection's direction between the variables, information can propagate in any direction [13].

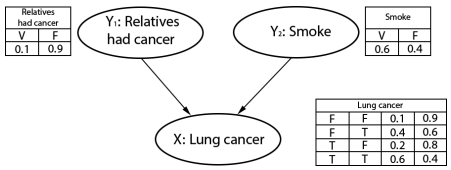

Figure 1. A Bayesian network example.

According to Fenton et al. [7], to define the NPT, the RNM user should define the resulting TNormal parameters constructing "truth tables" using example scenarios, which they define as "what if" analysis. An example is shown in Table 1. By analysing it, we can conclude that defining the parameters is not straightforward and there is a need to understand the TNormal mixture process to apply the RNM.

In Perkusich et al. [15], a simplified approach to use the RNM whenever there is a need to collect data from multiple experts was presented. Instead of using "what if" analysis, it asks the experts to order the relationships between the child and parents nodes given their relative magnitude. The collected data is analysed statistically and used to define the weights for the function of μ, having the function type defined to *WMEAN* and a fixed variance of $5.0E^{-4}$. Therefore, although it encapsulates the complexity of the RNM approach, it has limited modelling capabilities and, as discussed in Perkusich et al. [16], it might produce incorrect NPTs.

In da Silva et al. [5], an approach based on production rules was proposed to encapsulate the complexity of calibrating the NPTs. Given a set of input values, the developed expert system automatically calibrates the NPTs. In this work, the modelling capabilities of the approach presented in Fenton et al. [7] is combined with ranked nodes specific knowledge encapsulation of the approach presented in Perkusich et al. [16]. However, the proposed approach also fixed the variance in $5.0E^{-4}$, which is a limiting factor.

3 Solution

SBNE is an approach to elicit expert knowledge and apply the RNM method without relying on the assistance of RNM experts. SBNE stands for Simulated Bayesian Network Expert. This approach can be divided into three steps: (i) direct or indirect probability assessment from domain experts; (ii) use of production rules[1] to define the weighted expression; and (iii) use of the "Simulated Expert" algorithm to estimate the input parameters required to apply the RNM method. In (i) domain experts use a GUI (still a prototype) that allows them to evaluate probability distributions directly (i.e., using numbers), or indirectly (i.e., using a visual tool).

3.1 Knowledge Elicitation Process

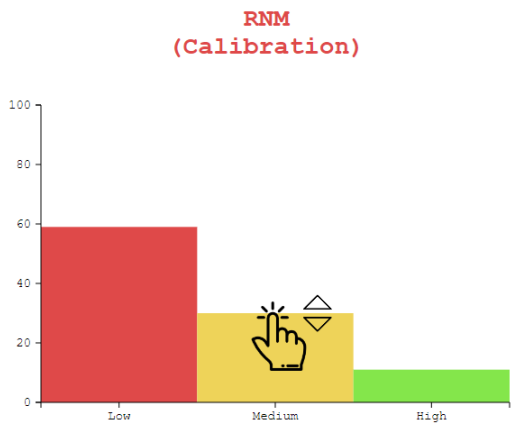

Figure 2. Component Prototype (A).

The prototype used in the probability elicitation process is here decomposed into two separate figures (for presentation purposes only). Hence, for each combination of extreme cases of the parent nodes (i.e., each row in the truth table) domain experts provides the expected probability distribution using an interactive bar chart (see Fig. 2) or sliders horizontally arranged (see Fig. 3). During

[1]Files available at https://github.com/SEKE2019/SBNE

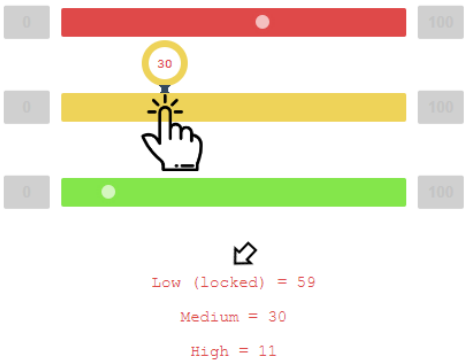

Figure 3. Component Prototype (B).

the elicitation process the domain experts directly interacts with the vertical bars so that by raising or lowering one of the bars the others automatically adjust itself. Strictly speaking, the natural order, which would be to inform the probability distributions with numbers and update the bar chart, is subverted so that users can interact and evaluate probability distributions by reasoning in terms of proportion rather than numerical terms, if they so wish. The sliders shown in Fig. 3 behave the same way.

A commonly employed strategy by domain experts is to first set up the bar relative to the state that they have greater confidence in estimating, and adjust the other states accordingly. To apply this strategy, the users can lock or unlock states by clicking over it, as shown with an arrow in Fig. 2, so that changes in other states do not modify the locked states. In short, the domain experts are able to provide the input data to the "Simulated Expert", reasoning in terms of proportion, in which case they ignore the numerical information of the prototype, or reasoning directly in numerical terms, in which they use the elements presented in Fig. 3.

That been established, let us consider a simple case in which we have a child node C with two parent nodes, A and B, all having three states each (e.g., "low", "medium" and "high"). To generate the child node's NPT, the domain expert needs to assess four probability distributions as shown in Table 2. The Table 2 is basically a truth table composed by all the combinations of extreme states of the parent nodes.

In this case, each row in Table 2 is filled with data from the interaction of the domain experts with the components presented in Fig. 2 and 3. In other words, the domain experts inform four probability distributions, which constitutes a subset of the child node's NPT. This subset is them used as input for the "Simulated Expert" to estimate the optimal parameters for the RNM algorithm. The weighted expression is defined using production rules as proposed in [5].

Table 2. Truth table for a node with two parents.

Rows	Parents		Child		
	A	B	C		
			Low	Medium	High
1	Low	Low	1	0	0
2	Low	High	0	0.3	0.7
3	High	Low	0	0.3	0.7
4	High	High	0	0	1

Table 3. Truth table approximation from the Simulated Expert.

Rows	Parents		Child		
	A	B	C		
			Low	Medium	High
1	Low	Low	0.9965	0.0035	0
2	Low	High	0	0.2995	0.7005
3	High	Low	0	0.2995	0.7005
4	High	High	0	0.0035	0.9965

Result obtained with the Simulated Expert for the child node C using the weighted expression *WMAX* with the following parameters: $\sigma^2 = 0.062$; weight 4 for all parent nodes; $BS = 3.94E^{-06}$.

3.2 Simulated Expert

The Simulated Expert is an algorithm that receives as input the target probability distributions and a weighted expression to estimate the most suitable parameters (i.e., variance and set of weights of the parents) for the RNM algorithm to generate the NPTs.

The algorithm can be divided into three steps: (i) search for the most likely range of the optimal variance; (ii) identify the combination of weights of the parent nodes; and (iii) estimate the optimal variance parameter, as detailed below.

Step (i):

1. generate a variance vector V in range $5.0E^{-4}$ to 0.2 with step $5.0E^{-4}$;
2. set the weight of all parent nodes to 5;
3. define a "resolution" constant $\delta = 10$ in which the desired accuracy is inversely proportional to its value;
4. calculate the step $s = \frac{|V|}{\delta}$ to perform the search for the most probable interval of the optimal variance;
5. traverse V and at each s, define the variance of the child node using the current variance (e.g., shaded boxes in Fig. 4), calculate the NPT and its relative score. Do that until there is no room for improvement (e.g., row 5 in Fig. 4) or until it reaches the end of V;
6. when the score starts decreasing return to the previous used variance index and calculate $a = index - \frac{|V|}{\delta}$ and $b = index + \frac{|V|}{\delta}$, the most likely interval where the optimal variance must be (e.g., row 7 in Fig. 4).

Step (ii):

1. set the variance of the child node using the median variance in $v \in V$ (i.e., $v = V[a : b]$), the interval at which the optimal variance is more likely to be (e.g., output of (i) illustrated in Fig. 4 row 7 in which it would be the variance located in index 5);
2. runs all combinations of weights of the parent nodes and stores the optimal set of weights.

Step (iii):

1. using the optimal set of weights obtained in previous step, traverse the subset v (e.g., row 7 in Fig. 4) performing the same actions as in item 5 from step (i).

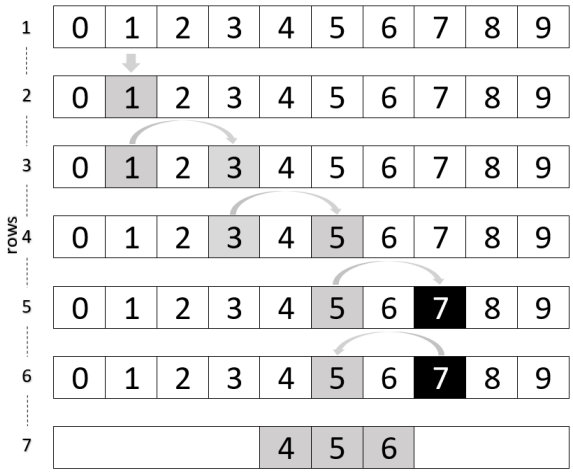

Figure 4. Illustration of the execution of the algorithm from step (i).

For each execution of the RNM algorithm, a score of the estimated probabilities relative to the target probabilities is computed using the Brier Score, but any other similarity measure can be used (e.g., euclidean distance, Kullback-Leibler divergence).

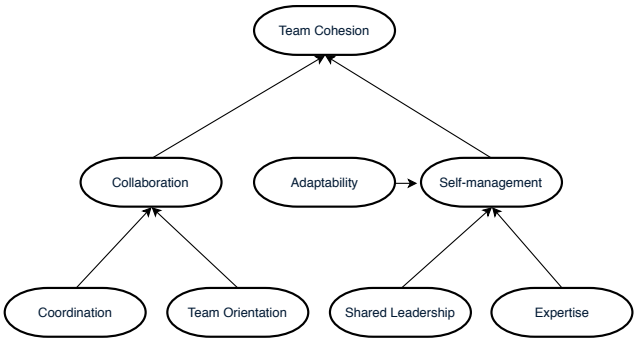

Figure 5. BN used in the experiment.

4 Empirical Validation

An experiment was conducted with undergraduate students that work as junior software developers at the Embedded Lab – a software development lab located at the Federal University of Campina Grande, Brazil. The purpose of the experiment was to validate the proposed approach.

In the experiment, 10 developers quantified a BN (adapted from the model proposed in Freire et al. [8]) using the proposed approach and two others methods: the Weighted Sum Algorithm (WSA) and a variation of the WSA, here named WSAAHP, that employs indirect probability elicitation by means of pairwise comparisons between states in which the domain experts compare the likelihood of states using a verbal and numerical scale.

Each domain expert also manually defined NPTs that served as benchmark for comparing the methods. A Randomised Complete Block Design (RCBD) was adopted in the experiment. The methods were compared in terms of accuracy. The accuracy is defined here as how well the NPTs generated represent the mode of probability distributions in the benchmark NPTs.

We focused on the following research question and null hypothesis:

> **RQ1**: Does the use of the SBNE approach to the RNM method maintain, improve or degrades expert-driven BNs accuracy?
>
> $\mathbf{H}_0$: The proposed approach to RNM is less accurate than the other methods.

The BN used in the experiment is presented in Fig. 5. The domain experts built 40 NPTs. Nevertheless, only the three parent nodes NPT was considered in the analysis (i.e., the one associated with the child node self-management), since it is the most complex. All the NPTs are available in an online repository[2].

[2]https://github.com/SEKE2019/SBNE

Table 4. Tukey simultaneous tests for differences of means

Difference of Method Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
WSA-RNM	-0.1267	0.0414	(-0.2323; -0.0211)	-3.06	0.018
WSAAHP-RNM	-0.1602	0.0414	(-0.2658; -0.0546)	-3.87	0.003
WSAAHP-WSA	-0.0335	0.0414	(-0.1391;0.0721)	-0.81	0.702

Individual confidence level = 98%.

An analyse of variance (ANOVA) was performed, which indicated that there is statistically significant difference (p-value = 0.003) between the accuracy of the methods with a significance level of 0.05. Tukeys HSD post hoc test was performed to determine which methods are in fact different in regards to accuracy level.

The Table 4 summaries the Tukey simultaneous test for differences of means. As can be seen in the Table 4, the confidence interval for the difference between the means of WSA-RNM and WSAAHP-RNM do not include zero, which indicates that the difference is statistically significant between these methods. The results show that RNM is 13% and 16% (i.e., rounded values) more accurate than WSA and WSAAHP, respectively. Therefore, $\mathbf{H}_0$ was rejected.

5 Threats to Validity

Despite the results obtained with the proposed approach, the comparison with different methods poses as a threat to internal validity. Nevertheless, the method WSA can be considered as a good benchmark because it has been mathematically and empirically validated in the literature [6, 2]. Moreover, the external validity may be limited, considering that the participants of the experiment were undergraduate students who work as software developers and that the experiment is bound to a specific context.

6 Conclusions

Despite recent popularity, the construction of BNs is still challenging. One of the challenges refers to defining the NPTs for large-scale BN. It is possible to automate this process using batch learning when there is a database with enough information. In practice, this is not common. The other option is to elicit data from experts, which becomes unfeasible for large scale BN. Fenton et al. [7] presented a solution based on ranked nodes. However, to apply this solution a BN expert is usually necessary.

In this paper, we complement the work of Fenton et al. [7] and Silva et al. [5] by presenting a novel approach to apply the RNM without relying on BN experts. The solution is named Simulated Bayesian Network Expert (SBNE) and it consists of eliciting a subset of the NPT from the domain

experts, which is used as input to estimate the optimal parameters (without relying on RNM experts) for the RNM to generate the NPTs.

Nonetheless, this approach can be used by RNM experts, reducing their effort to identify the optimal parameter for the RNM algorithm. We compared the proposed approach to the RNM with two other methods and the RNM outperformed them, reaching a mean accuracy of 75.78% against 63.12% of WSA and 59.77% of WSAAHP. These results are promising and validate our approach, which makes RNM accessible to a wider range of users.

Notwithstanding, it is not our goal to state which method is the best. For such a purpose, more experiments would be needed to investigate the matter. That said, it is our belief that future works in this area should concentrate on examining the proposed approach against BNs derived from RNM experts and comparing multiple methods using well known BN models from the literature such as ALARM [3] and Hailfinder [1].

References

[1] B. Abramson, J. Brown, W. Edwards, A. Murphy, and R. L. Winkler. Hailfinder: A bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

[2] S. Baker and E. Mendes. Evaluating the weighted sum algorithm for estimating conditional probabilities in bayesian networks. In *SEKE*, volume 2010, pages 319–324, 2010.

[3] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89*, pages 247–256. Springer, 1989.

[4] A. C. Constantinou, N. E. Fenton, and M. Neil. Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using bayesian networks. *Knowledge-Based Systems*, 50:60 – 86, 2013.

[5] R. M. da Silva, M. Perkusich, R. M. Saraiva, A. S. Freire, H. O. Almeida, and A. Perkusich. Improving the applicability of bayesian networks through production rules. In *SEKE*, pages 8–13, 2016.

[6] B. Das. Generating conditional probabilities for bayesian networks: Easing the knowledge acquisition problem. *arXiv preprint cs/0411034*, 2004.

[7] N. E. Fenton, M. Neil, and J. G. Caballero. Using ranked nodes to model qualitative judgments in bayesian networks. *IEEE Trans. on Knowl. and Data Eng.*, 19(10):1420–1432, Oct. 2007.

[8] A. Freire, M. Perkusich, R. Saraiva, H. Almeida, and A. Perkusich. A bayesian networks-based approach to assess and improve the teamwork quality of agile teams. *Information and Software Technology*, 100:119–132, 2018.

[9] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[10] D. Heckerman. Learning in graphical models. chapter A Tutorial on Learning with Bayesian Networks, pages 301–354. MIT Press, Cambridge, MA, USA, 1999.

[11] P. Laitila and K. Virtanen. Improving construction of conditional probability tables for ranked nodes in bayesian networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1691–1705, 2016.

[12] E. Lee, Y. Park, and J. G. Shin. Large engineering project risk management using a bayesian belief network. *Expert Syst. Appl.*, 36(3):5880–5887, Apr. 2009.

[13] J. Pearl and S. Russell. Bayesian networks. *Handbook of brain theory and neural networks*, 1995.

[14] M. Perkusich, K. Gorgonio, H. Almeida, and A. Perkusich. Assisting the continuous improvement of scrum projects using metrics and bayesian networks. *Journal of Software: Evolution and Process*, 2016. Article in Press.

[15] M. Perkusich, A. Perkusich, and H. O. de Almeida. Using survey and weighted functions to generate node probability tables for bayesian networks. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pages 183–188. IEEE, 2013.

[16] M. Perkusich, G. Soares, H. Almeida, and A. Perkusich. A procedure to detect problems of processes in software development projects using bayesian networks. *Expert Systems with Applications*, 42(1):437 – 450, 2015.