Chapter

Issues in the Probability Elicitation Process of Expert-Based Bayesian Networks

João Nunes, Mirko Barbosa, Luiz Silva, Kyller Gorgônio, Hyggo Almeida and Angelo Perkusich

Abstract

A major challenge in constructing a Bayesian network (BN) is defining the node probability tables (NPT), which can be learned from data or elicited from domain experts. In practice, it is common not to have enough data for learning, and elicitation from experts is the only option. However, the complexity of defining NPT grows exponentially, making their elicitation process costly and error-prone. In this research, we conducted an exploratory study through a literature review that identified the main issues related to the task of probability elicitation and solutions to construct large-scale NPT while reducing the exposure to these issues. In this chapter, we present in detail three semiautomatic methods that reduce the burden for experts. We discuss the benefits and drawbacks of these methods, and present directions on how to improve them.

Keywords: Bayesian networks, probability elicitation, node probability table, expert systems, artificial intelligence

1. Introduction

Bayesian network (BN) is a mathematical model that graphically and numerically represents the probabilistic relationships between random variables through the Bayes theorem. This technique is becoming popular to aid in decision-making in several domains due to the evolution of the computational capacity that makes possible the calculation of complex BN [1]. Some examples of BN application areas are: software development project management [2, 3]; large-scale engineering projects [4]; and the prediction of success in innovation projects [5].

On the other hand, there are open challenges related to the construction of BN. One of these challenges is to build the node probability tables (NPT). In cases where there are databases with enough information for the problem in question, it is possible to automate the process of constructing NPT through *batch learning* [6]. Unfortunately, in practice, in most cases, there is not enough data. That is, it is necessary to collect expert data and manually define the NPT [1].

Furthermore, experts can often understand and identify key relationships that data alone may fail to discover [7]. Therefore, the concept of smart data is defined by [7]: a method that supports data engineering and knowledge engineering

approaches with emphasis on applying causal knowledge and real-world facts to develop models.

In this context, it is necessary to manually elicit data from experts to define the NPT. However, given that the complexity of defining NPT increases exponentially, for large-scale BN, it becomes impracticable to manually define all the probability functions that compose each NPT [1]. In addition, experts often have time constraints and are rarely interested in manually defining NPT, partially because it is necessary to work with many probabilistic distributions for long periods [8].

In addition, other factors may compromise the process of probability elicitation to construct the NPT, such as commonly used heuristics. Some well know heuristics used to reduce the cognitive effort in probability assessment task may lead the expert towards biased judgment of probability, leading to systematic errors. Moreover, the experts are hardly able to keep mutually consistent distributions during the NPT definition [1]. In addition, factors such as boredom and fatigue are enough to make the criteria deviate during probability assessment [8], when in fact, it should be uniformly applied throughout the whole elicitation process.

A solution to solve this problem has been proposed by [1], which will be referenced herein as the ranked nodes method (RNM). Its goal is to define the NPT of the parent nodes and then generate the NPT of the child nodes. Ref. [1] introduces the concept of ranked nodes, ordinal random variables represented on a monotonically ordered continuous scale. A fundamental feature of this method is that mathematical expressions generate the child node's NPT. These expressions define the central tendency of the child node for each combination of states of the parent nodes and have as input a set of weights of the parent nodes, which quantifies the relative strengths of their influence on the child node, and a variance parameter.

Another approach was proposed by [8], which will be referenced here as the weighted sum algorithm (WSA). This method uses well know heuristics in its favor, more precisely, the availability [9] heuristic and the simulation [10] heuristic. The main focus of this method is to assemble part of the NPT from experts by asking questions that comprehend cases that are easy to recall by experts, which is likely to be associated to more realistic probabilities. In the WSA, the remainder of the NPT is generated using interpolation techniques.

A systematic approach to generate NPT of nodes with multiple parents is proposed in [11]. This approach is an adaptation of the analytic hierarchy process (AHP) method for the task of probability elicitation and semiautomatic generation of NPT, in which the expert needs only to make the assessment of probabilities conditioned on single parents. In this approach, the probability assessment is indirect by means of paired state judgments and the NPT is generated through the calculation of the product of the probabilities of the child node conditioned on single parents.

The three methods stated above reduce the burden for experts and allow the construction of complex BN in which manual elicitation of the NPT is unfeasible and, generally, there is not enough data to use batch learning. The reduced number of parameters to generate the NPT and consequently, reduced number of questions to ask the experts, makes it easier for the facilitator (e.g., BN expert) to deal with heuristics and possible biases during the NPT construction process. These methods can yet be extended with elaborate probability elicitation techniques (i.e., to improve its input).

Therefore, the objective of this research is to assess in detail three semiautomatic methods to generate NPT. We identified these methods in an exploratory study through a literature review. Additionally, we present heuristics that must be acknowledged during probability assessment for NPT construction and

discuss extensions to these methods. It is our understanding that these methods can yet benefit from elaborate probability elicitation techniques. Such techniques can add additional overhead when manually defining the NPT, but this overhead is hugely reduced with semiautomatic methods (i.e., given the reduced number of questions to ask the experts) making them a viable choice to improve the method's input.

This chapter is organized as follows. Section 2 presents an introduction to BN. Section 3 presents common heuristics which should be acknowledged and considered during the probability elicitation process. Section 4 presents a probability elicitation technique which can extend some of the semiautomatic methods. Section 5 presents three semiautomatic methods to generate NPT. Section 6 presents our conclusions and future works.

2. Background

Bayesian networks are graph models used to represent knowledge about an uncertain domain [12]. The Bayesian network, B, is a directed acyclic graph that represents a joint probability distribution over a set of random variables V [13]. The network is defined by the pair $B = \{G, \theta\}$, where $G = (V, E)$ is a directed acyclic graph with nodes V representing random variables and edges E representing the direct dependencies between these variables. θ is the set of probability functions (i.e., node probability table) which contains the parameter $\theta_{vi}|\pi_i = P_B(v_i|\pi_i)$ for each v_i in V_i conditioned by π_i, the set of parameters of V_i in G. Eq. (1) portraits the joint probability distribution defined by B over V. An example of a BN is depicted in **Figure 1**.

$$P_B(V_1, ..., V_n) = \prod_{i=1}^{n} P_B(V_i|\pi_i) = \prod_{i=1}^{n} \theta_{V_i}|\pi_i \tag{1}$$

In the above example, the probability of a person having cancer is calculated according to two variables: "Relatives had cancer" (Y_1) and "Smoke" (Y_2). The ellipses represent the nodes and the arrows represent the arcs. Even though the arcs represent the causal connection's direction between the variables, information can propagate in any direction [14]. Hence, the direction of the arrows indicates the dependency to define the probability functions. In this example, it is assumed that all the variables are Booleans. Since the node "Cancer" is pointed out by Y_1 and Y_2, the probability function is composed of probabilities for all possible combinations of states of Y_1 and Y_2.

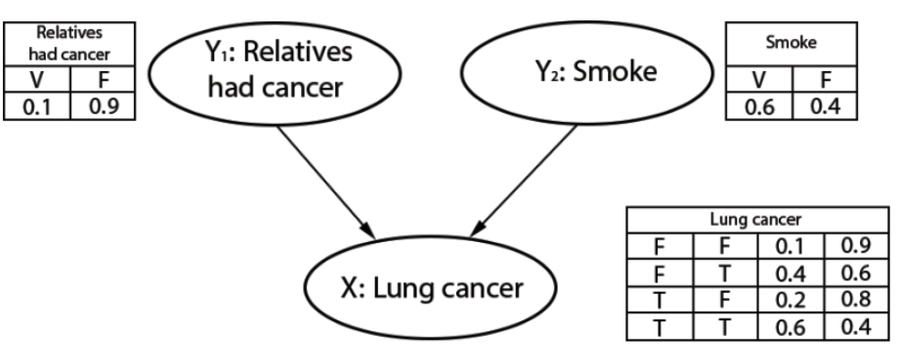

Figure 1.
BN example.

2.1 NPT's complexity

A challenge in constructing a BN is defining the NPT, which can be learned from data or elicited from domain experts. In practice, it is common not to have enough data for learning and elicitation from experts is the only option. However, the complexity of defining NPT grows exponentially, which makes the elicitation process costly and error-prone.

Let us consider the following example shown in **Figure 2**. In this BN, we want to assess *Teamwork* efficiency of a group of people that works collectively to achieve certain goals. *Teamwork* is directly influenced by *Autonomy* (i.e., self-management ability and shared leadership)*; Cohesion* (i.e., the capacity of being in close agreement and work well together)*;* and *Collaboration* (i.e., the ability to communicate and coordinate). This example will be used throughout this chapter.

To elicit all the probabilities needed to construct the NPT of the child node *Teamwork*, a facilitator (e.g., BN expert) has to ask 5^3 questions to the expert, a question for each $P(v_i|\pi_i)$. As we can see, the complexity of performing this task grows exponentially as the number of parents increases, making it quite expensive and error-prone.

Methods to address this problem were proposed. Noisy-OR and Noisy-MAX are two popular ones. However, the disadvantage of Noisy-OR is that it only applies to Boolean nodes. According to [1], the disadvantage of Noisy-MAX is that it does not model the extent of relationships required for large-scale BN. In this chapter, we present methods found in the literature that are applicable to a larger range of BN.

3. Heuristics in probability

The quantification process of a BN consists in converting expert knowledge, acquired through personal experiences, into probabilistic knowledge by eliciting a large number of subjective probabilities that reflect the expert's belief at a given moment about something. Probability assessment can be described as the task of quantifying the chances of an event occur, using percentages. However, as the degree of complexity increases, it becomes increasingly difficult to size the probability of occurrence of each of the possible events in a given scenario.

For instance, we may have a hunch as to who will be the winner of a particular tournament at a particular time, but we will never know for sure the exact probability since the number of factors that can influence the event goes beyond our reach. Apart from that, epistemic uncertainties (e.g., lack of knowledge about all

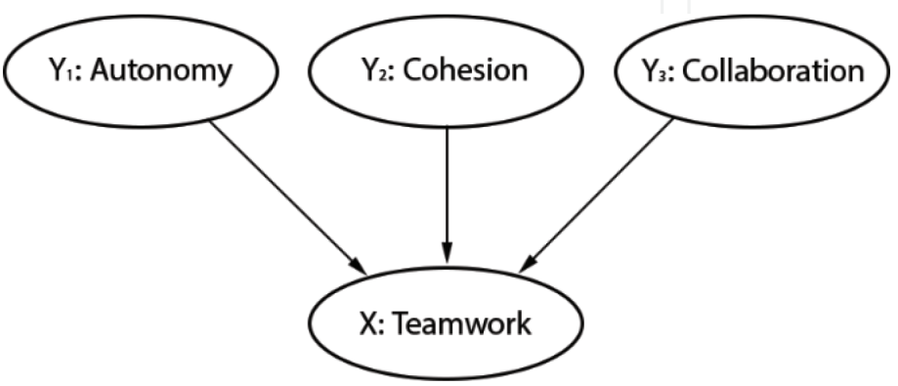

Figure 2.
BN example adapted from [15] where a child node Teamwork *is influenced by three parent nodes:* Autonomy *(Y_1),* Cohesion *(Y_2), and* Collaboration *(Y_3). Each node has five ordinal states: very low (VL), low (L), median (M), high (H), very high (VH).*

the participants in the tournament) and aleatory uncertainties (e.g., possibility of a team losing a player) play an important role in probability assessment. Nonetheless, if asked, one is capable of making an evaluation and give a quick answer. How do people manage to judge the probability of highly uncertain events?

According to [16], people make use of a limited number of heuristics, mental shortcuts, to reduce the complexity of judging the probability of an uncertain event. These mental shortcuts reduce the cognitive effort required to judge the probability of such events. However, they can lead to biases that result in systematic errors. In [16], three commonly used heuristics are presented: representativeness; availability; and anchoring.

The representative heuristic [16] describes the process by which people use the similarity of two events to estimate the degree to which one event is representative of another. It is used to answer questions such as: What is the probability that an event A originates from a process B? What is the probability of a process B generating event A? That is, if A is highly representative of B, the probability of A generating B is considered high. Conversely, if A is not representative of B, the probability of A originating from B is low.

Consider the following example adapted from [16]: "*Steve is very shy and withdrawn, has little interest in people, or in the real world. He has need for order and organization, and a passion for details".* Based on this description, what is Steven's most likely profession? Farmer or Librarian? You probably thought of a librarian. That happens because the probability of Steve's profession be a librarian is evaluated by the degree to which he is representative, or similar to, the stereotype of a librarian. However, several other factors that should have a significant effect on probability, like the prior probability, or base-rate frequency of the outcomes have no effect on representativeness. For example, the fact that there are many more farmers than librarians should be considered in this case, but it is neglected.

The availability heuristic [9, 16] is related to the judgment of probability of events occurring based on the ease with which we retrieve instances of these events in our mind. For example, to evaluate the likelihood that a person under the age of 30 years will suffer a heart attack, people usually do a quick search in their memory for cases they know of young people who have suffered a heart attack. This heuristic is useful because instances of larger classes are easier to remember than instances of smaller classes. However, the availability is affected by factors other than the frequency of events or probability. One may overestimate the probability of a young person getting cancer based on how recent an instance of such an event has occurred in his life, for example.

Anchoring and adjustment heuristic [16] occur when people judge probabilities based on an initial value, which is adjusted until the final response is reached. The problem with this heuristic is that the adjustments are usually insufficient. In other words, the expert assessment is likely to fluctuate around the initial anchor provided. It is important noting that, an anchor may be embedded in the formulation of a question to the domain expert (i.e., when a starting point is given), but it can also be the result of an incomplete computation.

In short, heuristics are mental shortcuts that reduce the cognitive effort in the task of reasoning about the probability of events with uncertainty. Although useful, it has its disadvantages that must be considered in the knowledge elicitation process. Therefore, it is imperative to acknowledge the possible biases derived from heuristics during the process of probability assessment, explicitly informing the experts of their existence and adopting appropriate methods to reduce their effects.

The number of probabilities to be elicited to construct an NPT may inevitably fall under some bias considering the effort needed from the experts. The semiautomatic methods reduce the number of questions to be asked to the expert or entirely

removes the need of direct evaluation of probabilities during the construction of the NPT, which makes it easier for the facilitator and the expert to deal with these heuristics during the elicitation process, seizing the benefits of the heuristics and reducing their possible negative effects.

4. Probability elicitation methods

The process of probability elicitation can be supported by a variety of techniques designed to aid experts when they find it hard to express their degrees of belief with numbers. These methods are based on setting-controlled situations in which probability assessments can be inferred from the expert's behaviors [17]. In this section, we describe the use of probability scales with visual aids to make probability assessment easier for experts. However, it is worth noting, visual aids like probability scales (i.e., which uses numbers) still tend to be biased.

It is our understanding that the use of visual elements such as probability scales can improve the input quality of semiautomatic methods (i.e., the ones which needs probability distributions as input), but indirect methods, which we do not discuss here, may improve the input quality as well. Several methods for indirect elicitation of probabilities have been developed. Some well know methods are: the odds method; the bid method; the lottery method; the probability-wheel method; among others [17, 18], these methods allow the extraction of probabilities without have to explicitly mention probabilities, so to speak.

Both direct and indirect methods can be incorporated at some degree into semiautomatic methods. The purpose of this section is to show one of these techniques which can extend semiautomatic methods, as an example. Also, different techniques may produce different results, so we encourage readers to check a comprehensive review of issues related to the probability elicitation task which has a section dedicated for direct and indirect methods [17].

4.1 Probability scale

A probability scale is composed of a line that can be arranged vertically or horizontally with discrete numerical anchors which denotes the probabilities. It is a direct probability assessment method. To assess a probability, the experts mark a position on the scale. The probability value is given by the marking distance to the zero point of the scale. An example of a numerical probability scale can be seen in **Figure 3**.

There is no standard scale. For instance, anchors may vary in distance and values according to the domain, and lines can be arranged in different positions. Moreover, during probability assessment, one can use both numerical and verbal anchors. In [19] it is proposed a double scale that combines numbers and textual descriptions of probability to aid in the communication of probabilities. According to [19], verbal descriptions commonly used by people to express probabilities are directly related to the numerical values of the probabilities itself. In **Figure 4**, we can see an example of a double scale arranged in the vertical position with numerical and verbal anchors.

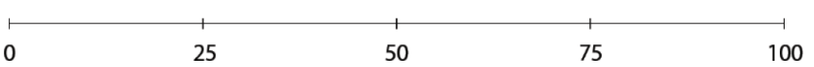

Figure 3.
Probability scale with numerical anchors.

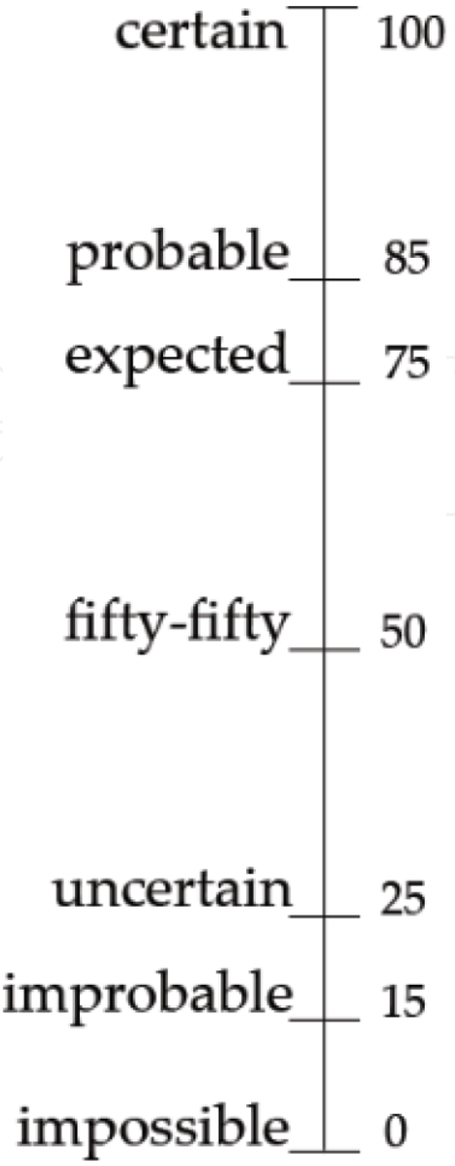

Figure 4.
Probability scale with numbers and words.

The advantage of using a scale is that it allows for the domain experts to think in terms of visual proportion rather than in terms of precise numbers. However, it is important to consider bias that may be introduced using probability scales. For example, let us say an expert is requested to indicate several assessments on a single line. In such a case, he is likely to introduce bias towards esthetically distributed marks. This bias is known as the spacing effect [17] and can be easily avoided by using a separate scale for each probability. Another bias that may be introduced by the use of probability scales is the tendency of people to use the middle of the scale. This bias is known as the centering effect [17].

Furthermore, scales can be used in combination with other components that may help in the task of probability assessment. In [20], a method is presented for elicitation of a large number of conditional probabilities in short time. This method was used to build a real-world BN for the diagnosis of esophageal cancer with more than 4000 conditional probabilities. This BN predicted the correct cancer stage for 85% of the patients [21]. The main idea of this method is to present to the expert a figure with a double scale and a text fragment for each conditional probability. An example of combining probability scales with other components can be seen in **Figure 5**.

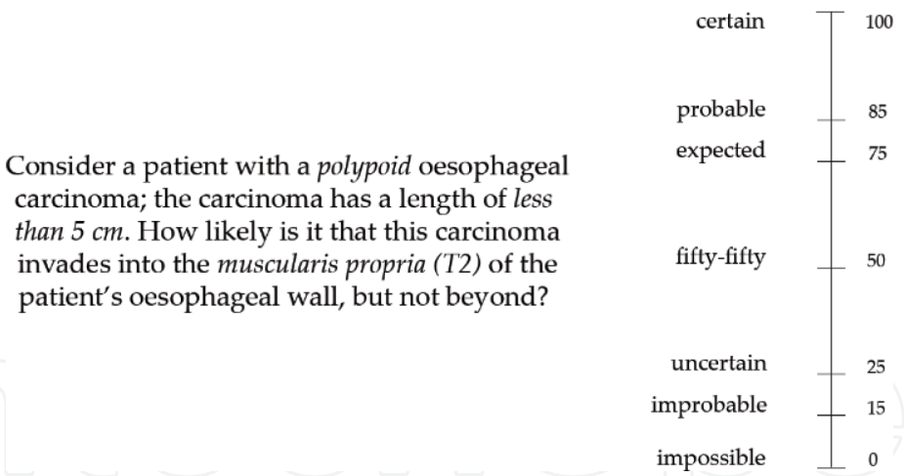

Figure 5.
Text fragment combined with a double scale for probability assessment.

On the left side is a text fragment describing the conditional probability to be assessed. On the right side, we have the double scale proposed in [19]. The text fragment is stated in terms of likelihood rather than frequency which circumvents the need for mathematical notation of the conditional probability. According to [21], the frequency format has been reported to be less liable to lead to biases and experts may experience considerable difficulty understanding conditional probabilities in mathematical notation. Conversely, such an approach may be less intuitive for domains in which it is difficult to imagine 100 occurrences of a rare event.

Nonetheless, in [20], the fragments of text and associated scales are grouped up accordingly to the conditional probability distribution. In so doing, domain experts can assess probabilities from the same conditional probability distribution simultaneously. In other words, the centering effect is avoided by presenting all the related probabilities (i.e., from the same probability distribution) at once for the expert to assess. This approach considerable reduces the number of mental changes during the probability elicitation process. In regards to the spacing effect, the proposed method avoids it by using a separated scale for each probability.

5. Semiautomatic methods

In this section, we present three methods to generate NPT that ease the burden for experts during the quantification process of a BN. These methods allow the construction of large-scale BN. The first is the RNM, which completely eliminates the need for direct probability assessment. The second is the WSA, which is based on two heuristics and needs only part of the NPT to be elicited from the expert. The third is an adaptation of the analytic hierarchy process (AHP) which reduces the cognitive effort, biases and inaccuracies from estimating probabilities to all combinations of states of multiple parents at a time. From now on, we will reference the latter as simply AHP. These three methods attack the magnitude problem of building NPT.

5.1 RNM

In [1], the ranked nodes method (RNM) is presented. In this chapter, it is introduced the concept of ranked nodes, ordinal random variables represented on a

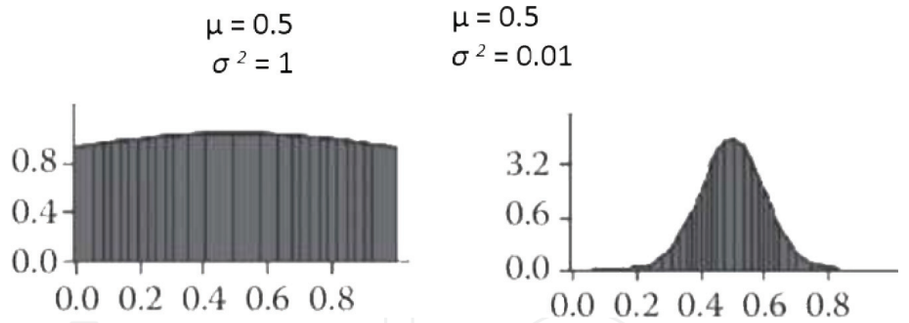

Figure 6.
Examples of TNormal.

continuous scale ordered monotonically in the interval [0, 1]. For example, for the ordinal scale ["Low", "Medium", "High"], "Low" is represented by the interval [0, 1/3], "Medium", by the interval [1/3, 2/3], and "High", by the interval [2/3, 1]. This concept is based on the doubly truncated Normal (TNormal) distribution.

A normal distribution is made of four parameters: μ, mean (i.e., central tendency); σ^2, variance (i.e., uncertainty about the central tendency); a, the lower bound (i.e., 0); and, b, upper bound (i.e., 1). With a normal distribution, it is possible to model a variety of curves (i.e., relationships) as a uniform distribution, achieved when $\sigma^2 = \infty$, and highly skewed distributions, achieved when $\sigma^2 = 0$. In **Figure 6**, we show an example of TNormal with same μ but different σ^2.

In this method, μ is defined by a weighted function of the parent nodes. There are four function types: weighted mean (*WMEAN*) Eq. (2), weighted minimum (*WMIN*) Eq. (3), weighted maximum (*WMAX*) Eq. (4) and a mix of both MIN and MAX functions (*MIXMINMAX*) Eq. (5). In practice, these functions define the central tendency of the child node for the combination of parent nodes states. The weight of each parent node, which quantifies the relative strengths of the influences of the parents on the child node, must be defined by a constant w in which $w \in \mathbb{N}$.

$$WMEAN(z_1, k, ..., z_n, k, w_1, ..., w_n) = \frac{\sum_{i=1}^{n} w_i * z_i, k}{\sum_{i=1}^{n} w_i} \tag{2}$$

$$WMIN(z_1, k, ..., z_n, k, w_1, ..., w_n) = \min_{i = 1, ..., n} \left\{ \frac{w_i * z_i, k + \sum_{j \neq 1}^{n} z_j, k}{w_i + n - 1} \right\} \tag{3}$$

$$WMAX(z_1, k, ..., z_n, k, w_1, ..., w_n) = \max_{i = 1, ..., n} \left\{ \frac{w_i * z_i, k + \sum_{j \neq 1}^{n} z_j, k}{w_i + n - 1} \right\} \tag{4}$$

$$MIXMINMAX(z_1, k, ..., z_n, k, wmin, wmax) = \frac{WMIN * \min_{i=1, ..., n} \{z_i, k\} + WMAX * \max_{i=1, ..., n} \{z_i, k\}}{WMIN + WMAX} \tag{5}$$

Fenton et al. [1] do not present the details to, in practice, implement the solution. Despite presenting the mixture functions, there is no information regarding the algorithms used to generate and mix TNormal, define samples size and define a conventional NPT given the calculated TNormals. The latter enables the integration of ranked nodes with other types of nodes such as Boolean and continuous, which brings more modeling flexibility.

In [22], it is proposed a probabilistic algorithm for this purpose, composed of two main steps: (i) generate samples for the parent nodes and

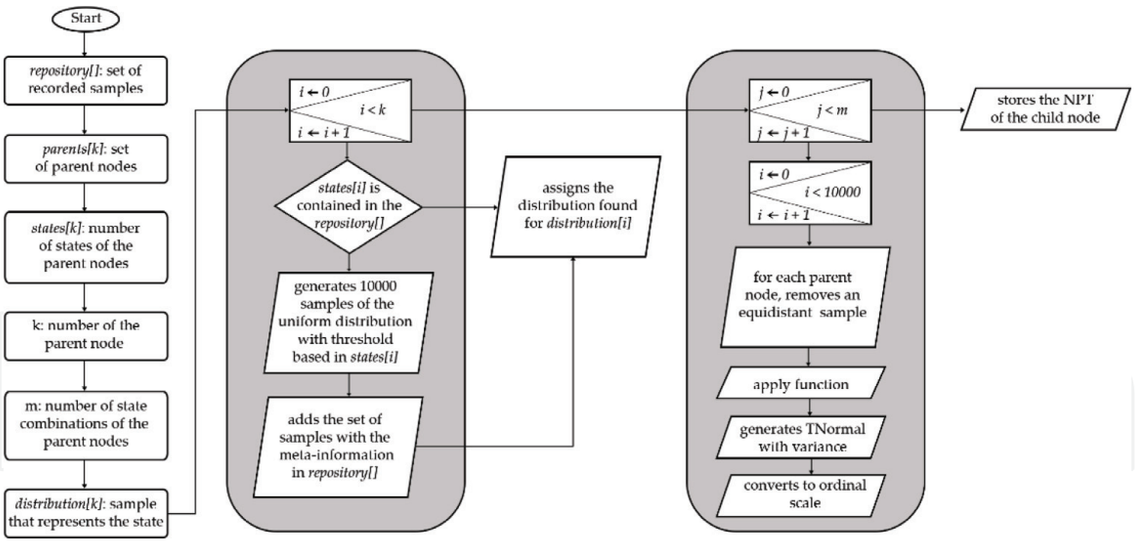

Figure 7.
Overview of the algorithm.

(ii) construct the NPT. In step (ii), for each possible combination of values for the parent nodes (i.e., each column of the NPT), the samples defined in the previous step are mixed given a function selected by the user and a TNormal is generated using the resulting mix and a variance defined by the user. An overview of the algorithm is shown in **Figure 7**.

As already mentioned, a ranked node is conceptually represented by an ordinal scale, which is mapped to the continuous interval [0, 1]. Thus, it is represented as a set of uniform distributions. For an ordinal scale with three values (e.g., "Bad", "Moderate" and "Good"): $U\ (0,1) = pbad * U\ (0,1/3) \cup pmoderate * U\ (1/3,2/3) \cup pgood * U\ (2/3,1)$, where p is the density of the distribution.

For the example shown in **Figure 8**, the set of uniform distributions is composed of the union of three uniform distributions: $U\ (0,1) = 54.7 * U\ (0,1/3) \cup 36.5 * U\ (1/3,2/3) \cup 8.80 * U\ (2/3,1)$. Numerically, this union is calculated using samples. Considering a sample size of 10,000, to represent the NPT of the example shown in **Figure 8**, it is necessary to collect 5700 random samples from $U\ (0,1/3)$, 3650 random samples from $U\ (1/3,2/3)$ and 880 random samples from $U\ (2/3,1)$.

Figure 7 shows that the algorithm is composed of four collections: *repository*[], a vector to store the samples of base states for the parent nodes; *parents*[k], a vector to store references to the parent nodes of each child node, in which k is the number of parents; *states*[m], a vector to store the states of each node, in which m is the number of possible values for the child node given the combination states of its parents; and *distribution*[m], a vector to store the resulting distribution for each possible combination of states of the parent nodes.

The repository strategy is used for optimization purposes. First, it is registered in memory (i.e., in *repository*[]) distributions that represent the base states, which are states with hard evidence (i.e., a node has 100% of chance for a given state). For instance, for a node composed of the states ["Bad", "Moderate", "Good"], are registered samples for: 100% "Bad", 100% "Moderate" and 100% "Good", which respectively has $\mu = 1/6$, $\mu = 1/2$ and $\mu = 5/6$. For this purpose, samples from a uniform distribution with the limits defined given the thresholds of the scale are collected.

For instance, for 100% "Good", it is collected samples of a uniform distribution limited in the interval [2/3, 1]. In [22] it is empirically defined that using a sample size of 10,000 is enough to guarantee a margin of error less than 0.1%. Each sample

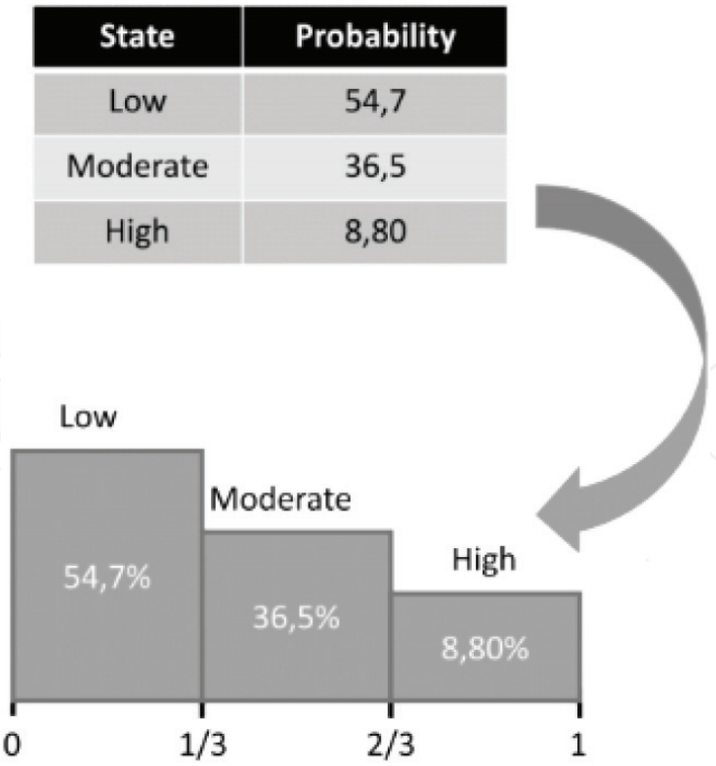

Figure 8.
Conversion from ordinal to continuous scale.

is registered with meta-data regarding its configuration (i.e., number of states and μ). The data in *repository*[] is used to generate samples for a node. Therefore, the samples for a base state are only generated once and reused later. The next step consists of, for each combination of the parent nodes, mix the TNormal using equidistant samples, randomly selected for each parent node. The samples are mixed using one of the given functions (e.g., *WMEAN*, *WMIN*, *WMAX* or *MIXMINMAX*) and the defined variance.

To mix the distributions, a random element from each sample of the parents is removed and used to calculate a resulting element using a given function. For instance, consider node A with two parents B and C. If we are calculating the probabilities of A for the combination "Low"-"High" and the selected function is *WMEAN* with equal weights, if the values removed in an iteration were 0.1 and 0.7, the resulting value would be 0.4. This step must be repeated until the collections of samples are empty.

Afterwards, the set of calculated elements and the given σ are used as input to generate a TNormal. The resulting distribution is converted to an ordinal scale and represents a column in the NPT of the child node (i.e., in the given example, the column for the combination "Low"-"High"). At the end of this step, all the possible combinations of states of the parent nodes are evaluated and the NPT for the child node is completed.

Accordingly, the inputs to generate the NPT of a child node are: a weighted expression capable of generating curves equivalent to distributions expected by the experts; a set of weights of the parent nodes; and a value for σ^2. To determine the weighted expression one can ask the experts to assess the mode of the child node for different combinations of the extreme states of the parent nodes [23]. For instance, let us consider the Bayesian network shown in **Figure 8** along with the mode assessments of the experts in **Table 1**.

Row	Parents			Teamwork				
	Autonomy	Cohesion	Collaboration	VL	L	M	H	VH
1	VL	VL	VL	X				
2	VL	VL	VH			X		
3	VL	VH	VH				X	
4	VH	VH	VH					X
5	VH	VH	VL				X	
6	VH	VL	VL		X			
7	VH	VL	VH				X	
8	VL	VH	VL				X	

Table 1.
Mode assessments for teamwork, checkmarks indicate the mode assessment of the expert.

First, let us consider the rows 1 and 4, where all the parent nodes are in the highest and lowest states respectively. As can be seen in **Table 1**, when o all the parent nodes are in the lowest or highest states, the mode of the child node is also the lowest or highest state. Such a probability distribution can be obtained by any of the weighted expressions.

Now, let us consider the row 1 as the initial state, rows 2, 6 and 8 indicate that when the state of a single parent node shifts from lowest to highest state the mode of the child node shifts towards the highest state. Similarly, consider row 4 as the initial state, rows 3, 5 and 7 indicate that when the state of a single parent node shifts from highest to lowest state, the mode of the child node also shifts towards the lowest state.

However, it is quite clear that the shift effect is stronger when it occurs from the lowest to highest state. Hence, **Table 1** reveals that the mode of the child node is inclined to go more towards the highest than lowest states which makes the *WMAX* function more suitable to express the distribution expected by the experts.

The process to determine the weights of the parent nodes and the variance parameter is not as straightforward as to determine the weighted expression. There is no guideline in the literature, as far as we know, to aid in this task. Nonetheless, one can use the mode assessments in **Table 1** as a starting point to define the weight of the parent nodes. For instance, considering *WMAX* is the most suitable function to express the probability distribution, let us examine the rows in which the states shift from lowest to highest in **Table 1**.

Finally, let us consider row 1 as the initial state, rows 2, 6 and 8 indicate that the parent nodes have different strengths of influence on the child node. That is, when the parent node *Autonomy* shifts from lowest to highest state the mode of *Teamwork* slightly shifts towards highest states, however, the shift is higher when the state changes in the parent node *Collaboration,* as can be seen, if one compares rows 2 and 6. A similar effect is observed when comparing rows 6 and 8. Hence, it is derived from **Table 1** the following constraint: *Autonomy weight* < *Collaboration weight* < *Cohesion weight*. Nevertheless, trial and error are yet necessary to discover suitable values for the weights and the variance parameter.

This method solves the magnitude problem of constructing NPT in complex Bayesian networks. On the other hand, a drawback to this method is that the domain context needs to fit a pattern that can be modeled by one of the weighted expressions. This solution has been validated through case studies in different real-world domains, such as human resources management in

software projects [24], software quality forecasting [25], air traffic control [26] and operational management [27].

5.2 WSA

In [8] the WSA method is proposed. This work introduced the concept of compatible parental configuration. The availability heuristic and the simulation heuristic are the base for this concept. As previously stated, the availability heuristic operates under the assumption that is easier to remember events that are more likely to occur. The simulation heuristic, in turn, operates according to which people determine the probability of an event based on how easy it is to simulate it mentally.

To formally define the concept of compatible parental configuration, we take as a basis the work of [28]. Superscript is used to represent the states of a node and subscript to differentiate the parent-nodes. Therefore, consider that for Y_i is assigned an arbitrary state y_i^v , that is, $Y_i = y_i^v$, since Y_j is another parent node, such that Y_j is considered compatible with $Y_i = y_i^v$ only when Y_j is in the state y_j^w which is most likely, according to the expert's knowledge, to coexist with $Y_i = y_i^v$. Hence, we use the notation $Comp\left[Y_i = y_i^v\right]$ to represent the set of states that are compatible with $Y_i = y_i^v$ for all parent nodes.

$$Comp\left[Y_i = y_i^v\right] = \left\{ y_j^w, \forall j \neq i \;\middle|\; \max_{w=1\ldots\left|Y_j\right|} P\left(y_j^w \middle| y_i^v\right) \right\} \tag{6}$$

The compatible parental configurations are captured during the elicitation process by asking the domain experts to choose off the top of their head a plausible combination of states for each $Comp\left[Y_i = y_i^v\right]$, which, theoretically, are easier to simulate and therefore, prone to more realistic probabilities. Hence, it is elicited from experts the probability distributions for all compatible parental configuration and relative weights. The NPT is calculated using a weighted sum algorithm [8] which takes these probability distributions and weights as input. The input data of the algorithm is obtained from the experts' knowledge, as follows:

1. relative weight (between zero and one) for each parent node, denoting its degree of influence on the child node $w_1, \ldots, w_n$;

2. $k_1 + \ldots + k_n$ probability distributions of X for compatible parental configurations.

$$p\left(x^l \middle| y_1^{v_1}, y_2^{v_2}, \ldots, y_n^{v_n}\right) = \sum_{i=1}^{n} w_j p\left(x^l \middle| Comp\left(Y_j = y_j^{v_j}\right)\right) \tag{7}$$

where w is the relative weight of the parent node, $l = 0, 1, \ldots, m$ and $v_j = 1, 2, \ldots, k_j$. A constraint must be observed: the sum of all the relative weights (i.e., of all parent nodes) must be exactly one. A weight equal to zero indicates that the parent node has no influence on the child node and therefore can be omitted from the relation. Conversely, a relative weight equal to one indicates that the parent node is the only determinant of the conditional probabilities on the child node.

For instance, let us consider the Bayesian network shown in **Figure 2** where we wish to assess teamwork. For the sake of simplicity let us say that all the parents have the states "Low", "Medium" and "High" instead of the five states from the original example. With WSA 3×3 distributions are needed to construct a complete NPT against 3^3 in case of manual elicitation. Starting with the parent Y_1, let us say

that the domain expert subjectively interprets the compatible parental configurations as an equivalence relation as follows:

$$\{Comp(Y_1 = s)\} \equiv \{Comp(Y_2 = s)\} \equiv \{Comp(Y_3 = s)\}, \text{for } s = l, m, h \tag{8}$$

When the domain expert provides 3 probability distributions over the node Y_1 then all 3×3 distributions for compatible parental configurations are obtained. To generate the NPT, the expert must assign relative weights to the parents to quantify the relative strengths of their influence on the child node. Let us say that the expert interprets *Autonomy* and *Cohesion* as having the same influence strength on the child node, and *Collaboration* as three times more important than *Cohesion* or *Autonomy*. Hence, assigning the following weights: $w_1 = .2$, $w_2 = .2$, $w_3 = .6$.

With the weights and 3 probability distributions over the node Y_1 as inputs, the weighted sum algorithm calculates all the 3^3 distributions required to populate the NPT. On the other hand, let us say that Eq. (8) is not satisfied, then all the 3×3 probability distributions must be elicited.

In such a case, the probability of *Teamwork (X) = "Low"* conditioned to *Autonomy (Y_1) = "Low", Cohesion (Y_2) = "Medium",* and *Collaboration (Y_3) = "High"* would be given by:

$$\begin{aligned} p(X = l|Y_1 = l, Y_2 = m, Y_3 = h) = {} & w_1 p(X = l|\{Comp(Y_1 = l)\} \\ & + w_2 p(X = l|\{Comp(Y_2 = m)\} \\ & + w_3 p(X = l|\{Comp(Y_3 = h)\} \end{aligned} \tag{9}$$

This summarizes the WSA method, for an in-depth description please check [8]. Unfortunately, [8] do not describe how to deal with situations where the expert cannot select a single compatible parental configuration. Hence, an extension to this method is proposed by [29] to deal with such situations by averaging the probabilities of valid compatible parental configurations that experts might select.

5.3 AHP

Although the direct assessment of probabilities in the construction of NPT is feasible for small Bayesian networks and relatively simple domains, for medium to large networks the complexity and burden for experts grows substantially. As the number of parents and states increase, the more difficult it becomes for experts to reason about conditional probabilities with multiple parents and multiple combinations of states at once, and the more susceptible it becomes to biases and inaccuracies [11].

In [11] it is proposed a systematic approach for generating conditional probabilities of nodes with multiple parents. It is an adaptation of the AHP method for the task of probability elicitation and semiautomatic generation of NPT where the expert only needs to provide probability assessments (i.e., indirect) conditioned on single parents. In this approach, the probability assessments are extracted from pairwise judgments of the states. The NPT is generated through the product of the probabilities of the child node conditioned on single parents.

Before using the proposed method [11] it is required to define an agreed upon scale to perform the pairwise judgments over the states of the node. Saaty's scale [30] can be used for this purpose or a custom one can be created. A good example of how to obtain a scale can be consulted in [19] in which four successive experiments were performed to generate a scale with numbers and words. The Saaty's scale has nine values as seen in **Table 2**.

Scale	Definition	Explanation
1	Equal likely	Event A and evet B are equal likely
2	Weak or slight	
3	Moderate more likely	Event A is moderate more likely than event B
4	Moderate plus	
5	Strong more likely	Event A is Strong more likely than event B
6	Strong plus	
7	Very strong more likely	Event A is very strong more likely than event B
8	Very, very strong	
9	Extremely more likely	Event A is extremely more likely than event B

Table 2.
Scale for the pairwise comparisons.

For a better understanding of the method, we substitute the original terminology used in the AHP for terms more appropriate to the probability context. Thus, the term *attribute* is replaced by *event* and the term *importance* is replaced by *likelihood*. To obtain prior probabilities pairwise comparisons of all states of the node are performed. Since each state is compared to every other state we can assemble a comparison matrix. In **Figure 9** we see an example of a comparison matrix used to define prior probabilities of a node.

In the above matrix, $a_{ij}(i = 1, 2, ..., n; j = 1, 2, ..., n)$ is specified by the question "comparing the state x^{s_i} with x^{s_j}, which is more likely and how more likely?". Once we have filled the values for a_{ij} we can find the values of a_{ji} by calculating the inverse of a_{ij}, i.e., $1/a_{ij}$. The final result is a reciprocal matrix with all elements in the diagonal equal to 1, that is, $a_{ii} = 1$ for all i.

The relative priority of x^{s_i} is obtained from the maximum eigenvector $\omega = (\omega_1, \omega_2, ..., \omega_n)^T$ of the matrix $(a_{ij})_{n\ x\ n}$ and the consistency of the matrix is the consistency ratio $CR = CI/RI$, where CI is the consistency index, defined by $(\lambda_{max} - n)/(n - 1)$ where λ_{max} is the maximum eigenvalue corresponding to ω, and RI is the random index given by **Table 3**. A comparison matrix with CR less than 0.10 is considered acceptable [11]. Although [31] has observed that this threshold may be inappropriate for the purpose of evaluating probabilities. Since the sum of

	x^{s_1}	x^{s_2}	x^{s_n}	ω
x^{s_1}	a_{11}	a_{12}	a_{1n}	ω_1
x^{s_2}	a_{21}	a_{22}	a_{2n}	ω_2
x^{s_n}	a_{n1}	a_{n2}	a_{nn}	ω_n

Figure 9.
Comparison matrix for prior probability elicitation of a node X.

n	1	2	3	4	5	6	7	8	9
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

Table 3.
Random consistency index where n is the number of states.

all elements in ω is 1 and the ith element ω_i represents the relative importance of the state x^{s_i}, ω_i is now interpreted as the prior probability of the state x^{s_i}, that is, $P(x^{s_i}) = \omega_i$.

Similarly, to obtain the probabilities of a node X with a single parent Y we estimate $P(x^{s_i}, |y^{s_j})$. In **Figure 10** we see the resulting matrix when node $Y = y^{s_j}$:

In the above matrix $a_{pq}(p = 1, 2, ..., n; q = 1, 2, ..., n)$ is specified by questions such as "if the node Y is in the y^{s_j} state, comparing the states x^{s_i} and x^{s_j} of the child node X, which one is more likely and how more likely? ". After obtaining $\omega_{ij}(i = 1, ..., n)$ we have $P(X = x^{s_i}|Y = y^{s_j}) = \omega_{ij}$. The number of matrices needed to obtain $\omega_{ij}(i = 1, 2, ..., n; j = 1, 2, ..., m)$ is equal to the number of states of Y. The obtained results compose the NPT of a child node X conditioned to the states of a parent Y, as shown in **Figure 11**.

The approach to generate the conditional probabilities for multi-parent nodes is based on [32], which states that when a node A in a Bayesian network has two parents B and C, its conditional probability in B and C can be approximated by $P(A|B, C) = \propto P(A|B)P(A|C)$ where $\propto$ is a normalizing factor that ensures that $\propto \sum_{a \in A} P(a|B, C) = 1$. Hence, to generate the complete NPT Eq. (10) is applied:

$$P(X = x^{s_i} \,|Y_1 = y_1^{s_i}, Y_2 = y_2^{s_i}, ..., Y_k = y_k^{s_i}) = \propto \prod_{j=1}^{k} P\left(X = x^{s_i}|Y_j = y_j^{s_i}\right) \qquad (10)$$

This approach focuses on easing the burden for experts by automatically generating probabilistic distributions of nodes with multiple parents, and consequently, the complete NPT through the calculation of the product of the probabilities conditioned on single parents. Thus, the expert assesses the probabilities of a particular child node conditioned to each of its parents, one at a time, and these probabilities are combined to get the node's conditional probability conditional on all its parents.

	x^{s_1}	x^{s_2}	x^{s_n}	ω_x
x^{s_1}	a_{11}	a_{12}	a_{1n}	ω_{1j}
x^{s_2}	a_{21}	a_{22}	a_{2n}	ω_{2j}
x^{s_n}	a_{n1}	a_{n2}	a_{nn}	ω_{nj}

Figure 10.
Comparison matrix of a node X conditioned on a single parent Y in the state y^{s_j}.

		States of Y		
		y^{s_1}	y^{s_2}	y^{s_m}
	x^{s_1}	ω_{11}	ω_{12}	ω_{1m}
States of X	x^{s_2}	ω_{21}	ω_{22}	ω_{1m}
	x^{s_n}	ω_{n1}	ω_{n2}	ω_{nm}

Figure 11.
Resulting NPT for a single parent node.

In [31] a similar method is proposed, also based on the AHP, which allow the quantitative evaluation of the inconsistency of experts in the task of probability assessment. The difference of the proposed methods is that in [11] the magnitude problem to construct NPT is reduced with a semiautomatic approach for the generation of the NPT and the cognitive effort is reduced because the experts only need to evaluate, indirectly, probabilistic distributions conditioned on a single parent at a time, whereas in [31] the effort is even greater than the direct elicitation of probabilities. Nonetheless, it is our understanding that the method proposed in [31] can somewhat extend other methods such as the WSA, without causing too much overhead. However, further studies are needed to confirm this.

6. Conclusion

Despite recent popularity, the construction of BN is still a challenging task. One of the main obstacles refers to defining the NPT for large-scale BN. It is possible to automate this process using batch learning, but it requires a database with enough information. In practice, this is not common. The other option is to elicit data from experts, which is unfeasible in most cases due to the number of probabilities required. A third option is to use semiautomatic methods that given an input (i.e., elicited from experts) generates the NPT.

In this chapter, we present three semiautomatic methods, found in an exploratory study through a literature review. These methods help, to a certain extent, to minimize the effects of human biases by reducing the parameters that are required to construct complete NPT. However, these methods are highly reliable on the input data elicited from experts. Therefore, flawed input necessarily produces nonsense output. For this reason, we present one of many probability elicitation techniques as an example, which can improve the input data needed by the semiautomatic methods and reduce the garbage in/garbage out effect.

The biggest problem with elaborated probability elicitation techniques is undoubtedly its cost, which is often greater than the direct elicitation of probabilities. Thus, these methods are not well suited for the construction of large-scale BN, despite been useful to deal with well know biases. However, it is our understanding that the cost to use elaborated probability elicitation techniques is drastically reduced when only is needed to elicit a small fraction of data of what would be necessary for manual definition of NPT. Therefore, the combination of semiautomatic methods and elaborated probability elicitation techniques might help building more reliable BN.

For example, let us consider the WSA method that uses a partial elicited NPT to generate a complete one using the concept of compatible parental configurations, weights of the parents and a weighted sum algorithm. Once the compatible parental configurations have been chosen, its probabilities can be elicited using a sophisticated probability elicitation technique with a rather small overhead. In one way, the probability elicitation technique becomes feasible and, theoretically, the input of the semiautomatic method is improved.

Nonetheless, it is evident that some methods may benefit more from elaborated probability elicitation techniques than others. However, it is still possible to use these techniques even in a method such as RNM. For example, the expert can inform the probabilities rather than the mode of each probabilistic distribution of the combination of extreme states (see **Table 1**). We believe that studies must be carried out to check if combining elaborated probability elicitation techniques with semiautomatic method can indeed improve the construction of large-scale BN.

Author details

João Nunes*, Mirko Barbosa, Luiz Silva, Kyller Gorgônio, Hyggo Almeida and Angelo Perkusich
Federal University of Campina Grande, Paraíba, Brazil

*Address all correspondence to: joaobatista@copin.ufcg.edu.br

IntechOpen

References

[1] Fenton NE, Neil M, Caballero JG. Using ranked nodes to model qualitative judgments in Bayesian networks. IEEE Transactions on Knowledge and Data Engineering. 2007;**19**(10):1420-1432

[2] Perkusich M et al. A procedure to detect problems of processes in software development projects using Bayesian networks. Expert Systems with Applications. 2015;**42**(1):437-450

[3] Perkusich M et al. Assisting the continuous improvement of scrum projects using metrics and bayesian networks. Journal of Software: Evolution and Process. 2017;**29**(6):e1835

[4] Lee E, Park Y, Shin JG. Large engineering project risk management using a Bayesian belief network. Expert Systems with Applications. 2009;**36**(3): 5880-5887

[5] De Melo ACV, Sanchez AJ. Software maintenance project delays prediction using Bayesian networks. Expert Systems with Applications. 2008;**34**(2):908-919

[6] Heckerman D. A tutorial on learning with Bayesian networks. In: Learning in Graphical Models. Dordrecht: Springer; 1998. pp. 301-354

[7] Constantinou A, Fenton N. Towards smart-data: Improving predictive accuracy in long-term football team performance. Knowledge-Based Systems. 2017;**124**:93-104

[8] Das B. Generating conditional probabilities for Bayesian networks: Easing the knowledge acquisition problem. arXiv preprint cs/0411034; 2004

[9] Tversky A, Kahneman D. Availability: A heuristic for judging frequency and probability. Cognitive Psychology. 1973;**5**(2):207-232

[10] Kahneman D, Tversky A. The Simulation Heuristic. No. TR-5. Stanford Univ CA Dept of Psychology; 1981

[11] Chin K-S et al. Assessing new product development project risk by Bayesian network with a systematic probability generation methodology. Expert Systems with Applications. 2009;**36**(6):9879-9890

[12] Ben-Gal I. Bayesian networks. Encyclopedia of statistics in quality and reliability. 2008;**1**

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

[14] Pearl J, Russell S. Bayesian networks. In: Handbook of Brain Theory and Neural Networks. Cambridge, MA, USA: MIT Press. 1998:149-153

[15] Freire A et al. A Bayesian networks-based approach to assess and improve the teamwork quality of agile teams. Information and Software Technology. 2018;**100**:119-132

[16] Tversky A, Kahneman D. Judgment under uncertainty: Heuristics and biases. Science. 1974;**185**(4157): 1124-1131

[17] Renooij S. Probability elicitation for belief networks: Issues to consider. The Knowledge Engineering Review. 2001; **16**(3):255-269

[18] Chesley GR. Subjective probability elicitation techniques: A performance comparison. Journal of Accounting Research. 1978;**16**(2):225-241

[19] Renooij S, Witteman C. Talking Probabilities: Communicating Probabilistic Information with Words

and Numbers. International Journal of Approximate Reasoning. 1999;**22**:169-194

[20] Van Der Gaag LC et al. How to elicit many probabilities. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 1999. pp. 647-654

[21] Van Der Gaag LC et al. Probabilities for a probabilistic network: A case study in oesophageal cancer. Artificial Intelligence in Medicine. 2002;**25**(2): 123-148

[22] Nunes Joao et al. An algorithm to define the node probability functions of Bayesian networks based on ranked nodes. International Journal of Engineering Trends and Technology (IJETT). 2017;**52**(3):151-157

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

[24] Fenton N et al. Making resource decisions for software projects. In: Proceedings of the 26th International Conference on Software Engineering. IEEE Computer Society; 2004. pp. 397-406

[25] Fenton N et al. Predicting software defects in varying development lifecycles using Bayesian nets. Information and Software Technology. 2007;**49**(1):32-43

[26] Neil M, Malcolm B, Shaw R. Modelling an air traffic control environment using Bayesian belief networks. In: 21st International System Safety Conference; Ottawa, Ontario, Canada. p. 2003

[27] Neil M, Fenton N, Tailor M. Using Bayesian networks to model expected and unexpected operational losses. Risk Analysis. 2005;**25**(4):963-972

[28] Mendes E et al. Towards improving decision making and estimating the value of decisions in value-based software engineering: The VALUE framework. Software Quality Journal. 2018;**26**(2):607-656

[29] Baker S, Mendes E. Assessing the weighted sum algorithm for automatic generation of probabilities in Bayesian networks. In: Information and Automation (ICIA), 2010 IEEE International Conference on. IEEE; 2010. pp. 867-873

[30] Saaty TL. How to make a decision: The analytic hierarchy process. Interfaces. 1994;**24**(6):19-43

[31] Monti S, Carenini G. Dealing with the expert inconsistency in probability elicitation. IEEE Transactions on Knowledge and Data Engineering. 2000;**12**(4):499-508

[32] Kim J, Pearl J. A computational model for causal and diagnostic reasoning in inference systems. In: International Joint Conference on Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 1983;1:190-193