Parameterising Bayesian Networks

Owen Woodberry,[1] Ann E. Nicholson,[1] Kevin B. Korb,[1] and Carmel Pollino[2]

[1]School of Computer Science and Software Engineering
[2]Water Studies Centre, School of Chemistry
Monash University, Clayton, Victoria 3800, Australia
{owenw,annn,korb}@csse.monash.edu.au
carmel.pollino@sci.monash.edu.au

Abstract. Most documented Bayesian network (BN) applications have been built through knowledge elicitation from domain experts (DEs). The difficulties involved have led to growing interest in machine learning of BNs from data. There is a further need for combining what can be learned from the data with what can be elicited from DEs. In this paper, we propose a detailed methodology for this combination, specifically for the parameters of a BN.

1 Introduction

Bayesian networks (BNs) are graphical models for probabilistic reasoning, which are now widely accepted in the AI community as intuitively appealing and practical representations for reasoning under uncertainty. A BN is a representation of a joint probability distribution over a set of statistical variables. It has both a qualitative aspect, the graph structure, and a quantitative aspect, marginal and conditional probabilities. The structure is a directed acyclic graph and formally represents the structural assumptions of the domain, i.e., the variables comprising the domain and their direct probabilistic dependencies, which are typically given a causal interpretation. The quantitative aspect associates with each node a conditional probability table (CPT), which describes the probability of each value of the child node, conditioned on every possible combination of values of its parents. Given both the qualitative and the quantitative parts, probabilities of any query variables posterior to any evidence can be calculated [10].

Most reported BN applications to date (including medical and other diagnosis, planning, monitoring and information retrieval - see [6, Ch.5] for a recent survey) have been built through knowledge elicitation from domain experts (DEs). In general, this is difficult and time consuming [4], with problems involving incomplete knowledge of the domain, common human difficulties in specifying and combining probabilities, and DEs being unable to identify the causal direction of influences between variables. Hence, there has been increasing interest in automated methods for constructing BNs from data (e.g., [11, 5]). Thus far, a methodology and associated support tools for Knowledge Engineering Bayesian Networks (KEBN) are not well developed. Spiral, prototype-based approaches to KEBN have been proposed (e.g., [7, 6]), based on successful software development processes (e.g. [2]). However, these provide little guidance on how to integrate the knowledge engineering of the qualitative and quantitative components or again on how to combine knowledge elicitation from DEs and automated knowledge discovery methods. While there have been attempts at the latter, they remain rudimentary (e.g., [9, 8]). Here we present a more detailed methodology, based on the spiral prototype model, for knowledge engineering the quantitative component of a BN. Our methodology explicitly integrates KE processes using both DEs and machine learning, in both the parameter estimation and the evaluation phases. The methodology was developed during the knowledge engineering of an ecological risk assessment domain, described in [12].

2 Quantitative Knowledge Engineering Methodology

A possible methodology for quantitative KEBN is outlined in Figure 1. This method illustrates possible flows (indicated by arrows) through the different KE processes (rectangular boxes), which will be executed either by humans (the DE and the knowledge engineer, represented by clear boxes) or computer programs (shaded boxes). Major choice points are indicated by hexagons.

The initial stage in the development spiral is **Structural Development and Evaluation**, which on the first iteration will produce an unparameterized causal network; a network structure must exist prior to parameterization and may need to be reconsidered after evaluation We do not describe this process in any detail, however it should also proceed in an iterative fashion. Once a BN structure has been established, the next step is **parameter estimation**, involving specifying the CPTs for each node. Figure 1 shows that the parameter estimates can be elicited from DEs (1),[1] or learned from data (2) or, as proposed here, generated from a combination of both sources (an example is shown in path 3). In early prototypes the parameter estimates need not be exact, and uniform distributions can be used if neither domain knowledge nor data are readily available. A detailed description of the parameter estimation process is provided in Section 3 below.

The second major aspect of quantitative knowledge engineering is **quantitative evaluation**. Evaluative feedback can be generated using either DEs or data or both, as we have done here. When data is available, several measures can be used to evaluate BNs, including predictive accuracy, expected value computations and information reward. DE evaluation techniques include elicitation reviews and model walkthroughs (see Figure 1). Another kind of evaluation is *sensitivity analysis*. This involves analysing how sensitive the network is, in terms of changes in updated probabilities of some query nodes to changes in parameters and inputs. Measures for these can be computed automatically using BN tools (shown as **Sensitivity to Parameters** and **Sensitivity to Findings** processes, in Figure 1), but these need to be evaluated by the DE in conjunction with the KE. A detailed description of sensitivity analysis is given in Section 4.

3 Parameter Estimation

During **expert elicitation** the DEs provide or refine estimates of the BN parameters. Direct elicitation employs such questions as *"What is the probability that variable **A** takes this state given these parent values?"* Alternatives are to use frequencies, odds, or qualitative elicitation, using terms such as 'high' or

[1] This can also include the domain literature as a source of parameter estimates.

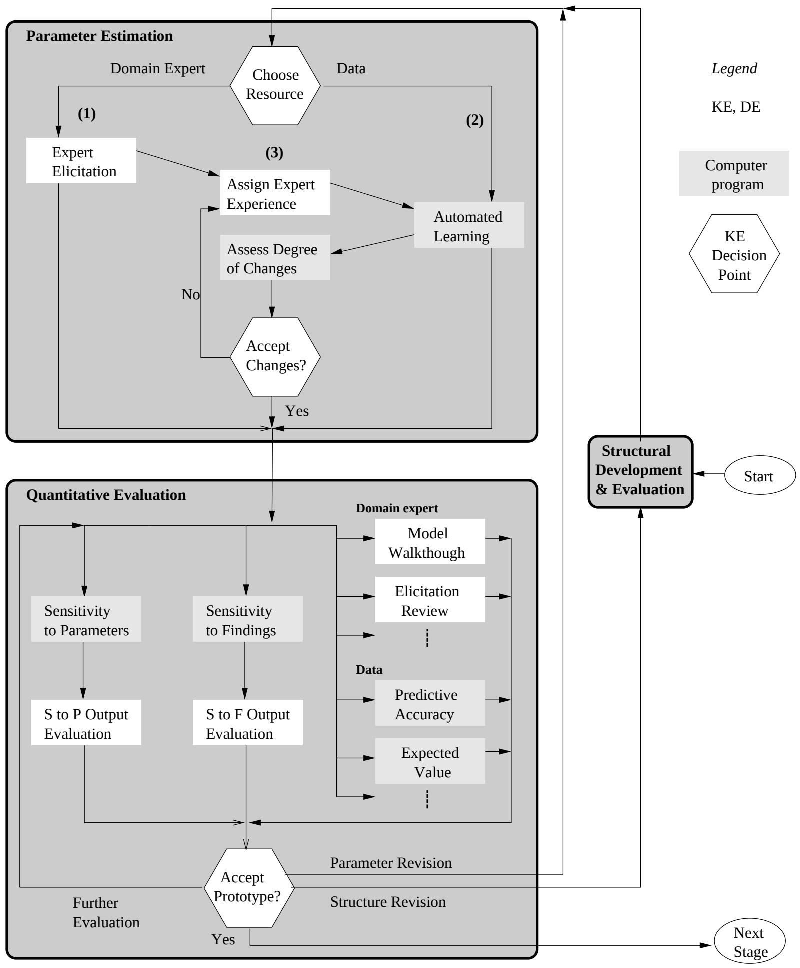

Fig. 1. Quantitative Knowledge Engineering Methodology

'unlikely', with the mapping to actual probabilities calibrated separately. In addition to eliciting precise parameters, it can also be useful to elicit an acceptable range for the parameter. As many are familiar with 95% confidence intervals from statistics, DEs might be comfortable reporting intervals having a 95% chance of capturing the desired parameter, although other ways of specifying a range of values are equally legitimate. Such intervals can be used during later evaluation to identify parameters needing further attention, as we shall see.

When data is of good quality and voluminous, estimating parameters from the data is clearly preferable. Many techniques are available for this (see [6, Ch.7]). Problems with incomplete data can be ameliorated also by incorporating other sources of information for parameters, such as expert knowledge, before automated learning. The combination of elicitation and data-based parameterization requires the elicited information to be weighted relative to the data available. In Figure 1 this is done in the **Assign Expert Experience** process, where an experience weighting is assigned to the expert parameter estimates, based on the confidence in the estimates obtained during expert elicitation. These are then treated as equivalent to the size of a hypothetical initial data sample.

After incorporating the data in parameter estimation, the next step is to compare the new with the original parameterization. In Figure 1 we consider this to be an automated process, **Assess Degree of Changes**. As mentioned above, during parameter elicitation an acceptable range of values can also be elicited. Any parameters estimated from the data to be outside this range should be flagged for attention. An alternative method for comparing the parameterizations looks at the Bhattacharyya distance [1] between the two probability distributions. This distance is computed for each possible combination of parent values; higher distances between conditional distributions trigger further attention. The DE must then assess whether these flagged parameter refinements obtained after automated learning are acceptable (in the **Accept Changes** decision point in Figure 1). If not, an iterative investigation of different mappings of the expert experience into equivalent sample sizes can be undertaken.

4 Quantitative Evaluation

After parameterization, the second major aspect of quantitative knowledge engineering is evaluation, which guides further iterations of BN development. When data is available, it can be used for evaluation. Where the data is also being used to learn the structure or the CPTs, it is necessary to divide it into training data and test data, so that evaluation is not done with the very same data used for learning. The most common method of evaluation is to determine the *predictive accuracy* of the BN, which measures the frequency with which the modal node state (that with the highest probability) is observed to be the actual value.

Even when adequate data is available, it is important to involve the DE in evaluation. If expert elicitation has been performed, a structured review of the probability elicitation is important. This procedure could involve: comparing elicited values with available statistics; comparing values across different DEs and seeking explanation for discrepancies; double-checking cases where probabilities are extreme (i.e., at or close to 0 or 1), or where the DEs have indicated a low confidence in the probabilities when originally elicited.

We now review two different types of sensitivity analysis and discuss how we adapted them into algorithms suitable for our purposes. One type of sensitivity study looks at how the BN's posterior distribution changes under different observed conditions, in a "sensitivity to findings" study. The other looks at how the model's distribution changes when particular parameters are altered. Curiously, researchers thus far appear to have employed one or the other of these, but not both in any one study (e.g., [3, 7]). Both are needed for a careful investigation of the properties of a network.

Sensitivity to Findings Analysis The properties of d-separation can be used to determine whether evidence about one variable may influence belief in a query variable. It is possible to measure this influence and rank evidence nodes by how much of an effect they have. This information can be used to provide guidance for collecting the most informative evidence or as a check on whether the model reflects the DE's intuitions.

Sensitivity to findings can be quantified using two types of measures, entropy and mutual information. **Entropy**, $H(X)$, is commonly used to evaluate the uncertainty, or randomness, of a probability distribution $H(X) = -\sum_{x \in X} P(x) \log P(x)$. We can measure the effect of one variable on another using **mutual information (MI)** $I(X|Y) = H(X) - H(X|Y)$. We have implemented this type of sensitivity to findings (see [12]). Our algorithm computes and displays both the entropy of a specified query node and the ranked mutual information values for a specified set of interest nodes, given a set of evidence for some other observed nodes. The user can subsequently investigate how changes to the evidence will affect the entropy and MI measures. This process allows the DE to identify whether a variable is either too sensitive or insensitive to other variables in particular contexts, which in turn may help identify errors in either the network structure or the CPTs.

Sensitivity to Parameters Analysis Identifying sensitive parameters in a BN is important for focusing the knowledge engineering effort, for it will focus effort in refining parameterization on those values which have the biggest impact on the target variables. How best to identify these sensitivities remains a current research topic. Sensitivity analysis could be done using an empirical approach, by altering each of the parameters of the query node and observing the related changes in the posterior probabilities of the target node. However, this can be extremely time consuming, especially on large networks. Coupé and Van der Gaag [3] address this difficulty by first identifying a "sensitivity set" of variables given some evidence. These are those variables which can potentially change, meaning the remaining variables can be eliminated from further analysis. The sensitivity set can be found using an adapted d-separation algorithm (see [12]). Coupé and Van der Gaag also demonstrated that the posterior probability of a state given evidence under systematic changes to a parameter value can be given a functional representation, either linear or hyperbolic.

We have implemented this type of sensitivity to parameters (see [12]). When a particular evidence instantiation is set, our algorithm identifies the type of sensitivity function for the parameters by checking whether the query node has any observed descendant nodes. Once the sensitivity function is determined for a parameter, its coefficients can be computed. If the plotted sensitivity function does not behave as the DE expects (its slope, direction or range is unexpected), then this could indicate errors in the network structure or CPTs.

KE Decision: Accept Prototype Quantitative evaluation can be used to identify problems with the BN structure and parameters. After the model has been evaluated using a particular technique, the KE and DE must determine whether the prototype is to be accepted for the next stage of development. This decision is not intended to be the end of the knowledge engineering, or even prototyping, process. If the prototype is not sufficiently validated for prototype acceptance, **Further evaluation** is one option for the KE and DE. It will often be necessary to use multiple evaluation techniques to validate the model: for example, sensitivity to findings and parameter analyses evaluate different aspects of the model with little overlap, and hence don't substitute for each other. If problems with either the structure or the parameters have been identified, it will be necessary to re-visit the relevant KE processes, **Structural Development & Evaluation** or **Parameter Estimation** respectively, via the main spiral iteration in Figure 1.

5 Conclusion

This study presents a practical approach to the knowledge engineering of Bayesian networks, specifically focusing on their parameterisation. In many real-world applications neither human expertise nor statistical data will suffice to generate parameters reliably. Our methodology incorporates both sources in an iterative prototyping approach, which is guided by quantitative evaluation techniques. We have employed this method successfully in our ecological risk assessment model, which has been accepted for use [12]. In future work we will continue to develop the methodology in application to our ERA model and in other domains.

References

1. A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematics Society*, 35:99–110, 1943.
2. F. Brooks. *The Mythical Man-Month: Essays on Software Engineering.* Addison-Wesley, Reading, MA, second edition, 1995.
3. V. M. H. Coupe and L. C. van der Gaag. Properties of sensitivity analysis of Bayesian belief networks. *Annals of Mathematics and Artificial Intelligence*, 36:323–356, 2002.
4. M.J. Druzdzel and L.C. van der Gaag. Building probabilistic networks: Where do the numbers come from? *IEEE Trans. on Knowledge and Data Engineering*, 12(4):481–486, 2001.
5. D. Heckerman and D. Geiger. Learning Bayesian networks. In P. Besnard and S. Hanks, editors, *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI95)*, pages 274–284, San Francisco, 1995.
6. K. B. Korb and A. E. Nicholson. *Bayesian Artificial Intelligence.* Computer Science and Data Analysis. CRC, Boca Raton, 2004.
7. K.B. Laskey and S.M. Mahoney. Network engineering for agile belief network models. *IEEE: Transactions on Knowledge and Data Engineering*, 12:487–98, 2000.
8. A. Nicholson, T. Boneh, T. Wilkin, K. Stacey, L. Sonenberg, and V. Steinle. A case study in knowledge discovery and elicitation in an intelligent tutoring application. In Breese and Koller, editors, *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI01)*, pages 386–394, 2001.
9. A. Onisko, M.J. Druzdzel, and H. Wasyluk. Learning Bayesian network parameters from small data sets: application of Noisy-OR gates. In *Working Notes of the Workshop on "Bayesian and Causal networks: from inference to data mining." 12th European Conference on Artificial intelligence (ECAI-2000)*, 2000.
10. J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, CA, 1988.
11. C.S. Wallace and K.B. Korb. Learning linear causal models by MML sampling. In A. Gammerman, editor, *Causal Models and Intelligent Data Management.* Springer-Verlag, 1999.
12. O. Woodberry, A. Nicholson, K. Korb, and C. Pollino. Parameterising Bayesian networks: A case study in ecological risk assessment. Technical Report 2004/159, School of Computer Science and Software Engineering, Monash University, 2004.