Implementing and Improving a Method for Non-Invasive Elicitation of Probabilities for Bayesian Networks

Martinus de Jongh, Marek Druzdzel, Leon Rothkrantz

Abstract: *Knowledge elicitation is difficult for expert systems that are based on probability theory. The elicitation of probabilities for a probabilistic model requires a lot of time and interaction between the knowledge engineer and the expert. Druzdzel and van der Gaag [5] have proposed a theoretical framework that would allow the use of multiple types of probabilistic information for elicitation of Bayesian networks. This framework has been used as a starting point for the implementation of a non-invasive elication method.*
Keywords: *Artificial Intelligence, Bayesian networks, conflict detection, Dirichlet distribution, GeNIe, knowledge elicitation, linear programming, probability theory, SMILE.*

INTRODUCTION

Bayesian networks (BNs) [7] can be used to store the knowledge of human experts in probabilistic models. BNs are graphical models that very efficiently represent joint probability distributions. This efficiency is the result of conditional independence assumptions that are added to the model. To create a Bayesian network, its graph needs to be defined and its conditional probability tables (CPTs) for every node need to be filled with the necessary probabilities. Basically three approaches exist:

- Elicit the necessary information from a domain expert.
- Learn the necessary information from data.
- The knowledge engineer creating the BN estimates the probabilities himself, using relevant literature.

Both the graph and the CPTs can be created using all of the methods. Depending on the application one method may be more appropriate than the other. Generally, in situations where there is not enough data or no data at all, elicitation of knowledge from experts is the only viable approach to get the necessary probabilities for a model.

Knowledge elicitation for systems based on probability theory is difficult. To describe this difficulty in a nutshell: the expert has to assign probabilities, which are numerical values, to all the possible events the application is modelling. Beside the fact that the number of probabilities increases exponentially with the size of the model, experts may find it difficult to assign an exact value to an event. It is generally better to use an indirect approach that uses a sort of betting game to determine the desired probabilities [2][4].

Until now it is assumed that the expert actually somehow, maybe by using betting games, can give an estimate of the desired probabilities. However, it is possible that an expert cannot give an estimate for a certain probability directly, but only implicitly by estimating other probabilities first and calculating the desired probability using, for instance, Bayes' rule. Also an expert may have information that is not quantitative of nature, i.e. not in the form of numerical probabilities. Other types of probabilistic information exist that are qualitative of nature and cannot be directly interpreted as probabilities for a Bayesian network. If the expert can provide these types of information and it is relevant for the model to be developed it would be very inefficient if it would be impossible to use this information for the model.

If it were possible to use probabilities and probabilistic information other than the necessary conditional probabilities directly this would make the elicitation process easier for the expert. Here with easier it is meant that any information relevant to the model can now be used for filling the CPT entries of the Bayesian network, and it would no longer be necessary to let the expert transform the information he or she has into conditional

probabilities that can be directly put in the BN. Such an elicitation method can be considered to be non-invasive. Non-invasive can be defined as [5]:

"Allowing any type of probabilistic information, quantitative or qualitative of nature, the expert is willing to state to be interpreted directly for the elicitation of probabilities for a Bayesian network."

Druzdzel and van der Gaag [5] have proposed a theoretical framework for an elicitation method that lets an expert specify various statements of qualitative and quantitative nature and interprets these statements as constraints used to guide the determination of the CPT entries for a Bayesian network.

The outline of this paper is as following: first related work is discussed, followed by a discussion of the theory of the performed research. Next, one of the performed experiments is discussed, ending with conclusions and future work.

RELATED WORK

A lot of work has been done in knowledge elicitation in general [3], many different types of interviewing and observation techniques exist. Specifically for eliciting probabilities usually indirect elicitation techniques are used to elicit probabilities from experts. One of the most popular techniques is using betting games [2].

Betting games can be helpful when an expert can give an estimation of the desired probability. When the expert cannot produce an estimate for the desired probability, but has other relevant and useful knowledge, it should be possible to use the information for the model. There exist several methodologies that are not based on probability theory and that can incorporate other types of knowledge into a model. Two examples are Possibility Theory [9], that uses non-crisp logic allows for dealing with uncertain and imprecise information, and Dempster-Shafer theory [8], which is a generalization of the Bayesian theory of subjective probability and focuses more on belief functions and the degree of belief one has in a statement.

The approach proposed by Druzdzel and van der Gaag [5] is to consider the distribution hyperspace of all possible joint probability distributions over the variables. A point somewhere in this hyperspace will be the true probability distribution, Pr, over the variables. If there is no information available, qualitative or quantitative, then Pr can be any point in the hyperspace. Once more information is known about Pr, some of the probability distributions in the hyperspace will become incompatible with this information.

Probability elicitation can be looked upon as constraining the distribution hyperspace as much as possible to find the true distribution Pr. All information for the distribution Pr is expressed as constraints for the hyperspace, and, assuming that all compatible distributions are equally likely, 2^{nd} order probability distributions over the probabilities of the distribution Pr are derived. These distributions are used for determining the probabilities of the joint probability distribution. Since the method allows the use of various types of probabilistic information it is possible to use any information the expert is willing to state. This allows for the process of eliciting probabilities to be non-invasive. The basic idea of the approach is to have a canonical form for the interpretation of probabilistic information.

The form builds on the property that any joint probability distribution on a set of variables is uniquely defined by the probabilities for all possible combinations of values for all variables. With all these values known, any probability over the set can be computed by using marginalization and conditioning. Any information about the true distribution Pr can now be represented as a system of (in)equalities with the constituent probabilities as unknowns. Any solution of this system will be a joint probability distribution that is compatible with all the specified probabilistic information. If there are no solutions, then the

provided information is inconsistent.

THEORY

The framework proposed by Druzdzel and van der Gaag [5] was used as a starting point for an implementation of a non-invasive elicitation method. The framework was implemented and improved, which was necessary to make it feasible.

Decomposing a Bayesian Network

The BN is decomposed into smaller sub networks to make the joint probability distributions (J-PDFs) that the method has to work with have a more manageable size. A method for decomposing a BN has been designed. The idea is to break up the BN into families; sub networks that consist out of a node and its parents. Using family networks (FNs), the expert can focus on providing information for a only a small part of the network without having to worry about other nodes.

Translation of Expert Statements into Constraints

When the BN is decomposed, the expert can provide probabilistic statements per family for each family. Once the expert is done, the statements will have to be translated into constraints for the probability hyperspace. The equations/inequalities are represented by binary expression trees. The reason for this choice was that binary expression trees are most commonly used for similar problems.

The parsing of the different types of probabilistic statements was designed in such a way that the more complex probability statements make use of the simpler ones or creates extra constraints when necessary. One example is when dealing with conditional probability statements. An extra constraint must be added to ensure that the conditional part of the probability has a probability larger than 0.

Identification of Probability Bounds

When the system has acquired the constraints from the expert, it could start the sampling process, but this would be quite inefficient [5]. First a pre-processing step is performed to reduce the size of the sample space. Linear programming (LP) is used to tighten the bounds for every constituent.

It is important to notice that only the linear constraints provided by the expert can be used for these calculations. It was researched how it would be possible to automatically determine if a constraint was linear and to extract the necessary information from the constraints to create the matrices necessary for the LP procedure. Using some knowledge of how the different constraints typically look like and by implementing some symbolic mathematical operations for the expression trees, a method was developed that manipulates the trees into a standard form that makes it very easy to decide if a constraint is linear or nonlinear. There may exist situations where this method will fail, but in these cases the method will mistake a linear equation for an nonlinear. In this situation the constraint will be excluded from the LP process, which is not as bad as trying to include nonlinear constraints in the LP process.

Using the LP procedure does indeed decrease the size of the sample space and thus will improve the efficiency of the sampling process, but because the nonlinear constraints are not considered in this process, there should still be more to gain in sampling efficiency.

Derivation of the 2nd order Distributions

To generate samples for evaluation, a truncated Dirichlet distribution (TDD) has been used. This is a variant of the Dirichlet distribution where the entries of the sample vector

can be constrained by lower and upper bounds. Samples from a TDD are generated by sequentially generating samples from the marginal distributions of the TDD, which according to Fang et al.[6] are truncated beta distributions, which are beta distribution that are sampled between specified upper and lower bounds. An algorithm to generate samples for TDDs has been created using [6].

Conflict Detection

All the constraints have to be satisfied for a sample to be valid, any sample that does not satisfy all constraints must be discarded. There is one big problem with this approach: conflicting constraints will prohibit any sample to be valid. A set of constraints is conflicting with each other when it is impossible for all the constraints to be satisfied at the same time. Detecting conflicts is hard, one can never be sure that the absence of valid samples is because of conflicting constraints or that the sampling procedure just has not hit inside the feasible area. Pin-pointing the conflicting constraints is even harder, and there does not yet exist a good procedure for finding conflicting constraints for the nonlinear, non convex case, which is the worst case situation encountered when applying the method.

Two heuristics are being used to aid the expert when deciding which constraint(s) need to be changed or removed. One heuristic (majority heuristic) was devised by the authors and the other, the constraint effectiveness heuristic, was created by [1].

The majority heuristic is based on the idea that when the majority of the constraints evaluates a sample as true that there might be something wrong with the minority of constraints that has evaluated the sample as false. In this situation 1 is added to the minority counter for each constraint that belonged to the minority. In the situation that the majority evaluated the sample as false, the minority counters are left unchanged. The reason for this is that there are no conflicting constraints only when all constraints have simultaneously evaluated a sample as true at least once.

When all constraints simultaneously evaluated a sample as false there is no guarantee that the set of constraints does not contain any conflicting constraints. Thus when a constraint evaluates a sample as true, this can be considered as stronger evidence then when it evaluates the sample as false.

After all samples have been processed by the heuristic, some steps are performed to normalize the results so that each statement provided by the expert gets a heuristic value between 0 and 1. The closer the value is to 1.0 the "worse" the constraint is.

Merging Family Networks

Merging the family networks and the generated samples back into the original BN is not an easy process. The sample process has been performed on each of the family networks independently and possibly the results for the CPT entries may vary per FN. Since different FNs contain different variables it is likely that different FNs do not have the same set of constraints. The difference in constraints has influenced the sampling process and through the collected samples, the shapes of the histograms for the different CPT entries of the nodes. Another problem that contributes to the difficulty of merging FNs is that when the BN is decomposed in the FNs, some nodes will be a child node in one FN and a parent in possibly multiple other FNs. When this happens the node will have different CPTs for the different FNs. When the node is a child, its values will be conditioned on all its parents. When the node is a parent, it may not have any parents itself in this FN and in this case it will have a prior distribution and not a conditional one.

A merging method for merging the FNs into the BN that is mathematical sound has not been found. Currently, the FNs with their samples and histograms are shown to the user of program (most likely a domain expert and/or a knowledge engineer), and the user

must decide how he or she wants to fill in the BN. The idea always was to give the user more control over the end result, but with just one histogram for each entry of the nodes.

EXPERIMENT

An experiment was done to evaluate the performance of the implementation created for the method.

Goal: The goal for the experiment is to compare the influence of two different probability statement sets on the resulting histograms.

Design: The Bayesian network used in the experiment is the alarm system network by [7]. Two sets of constraints were created for this network. The first set of constraints can be considered as a sort of control group. For half of the CPT entries of the BN probability statements were created that would be in the form $0.9x \leq P(\cdot) \leq 1.1x,$ where P(.) is the probability of a CPT entry and x is the real value of the entry. These statements create an interval around the real value, where the upper and lower bound differ 10% from the real value. Some example statements are shown in Table 1:

Table 1: Probability statement

0.0009 ≤ P(b) ≤ 0.0011
0.0018 ≤ P(e) ≤ 0.0022
0.855 ≤ P(a|b,e) ≤ 1.0
0.81 ≤ P(j|a) ≤ 0.99
0.63 ≤ P(m|a) ≤ 0.77

The second set of probability statements was created in a manner that is more likely to be encountered in real life. Normally an expert would be necessary to create the statements and he or she would probably be able to translate rules of thumb into probability statements. For the experiment probability statements have been created that describe the CPT entries of the BN loosely. The statements are still based on the true values of the CPTs, but no intervals have been specified and values in the statements have been chosen to be less restrictive.

Results Both the probability statement sets have been inputted in the method together with the alarm system BN. Three different family networks were identified.

For both sets of probability statements the method has generated constraints that were used for the sampling process. The probability statements are divided over the FNs. The generated FNs are shown in Figure 1. Only statements that are relevant for a FN are translated into constraints for the sampling process. After these constraints were generated the Linear Programming process was run using the linear constraints. All the constraints were linear, so all could be used to determine probability bounds for all the family networks.

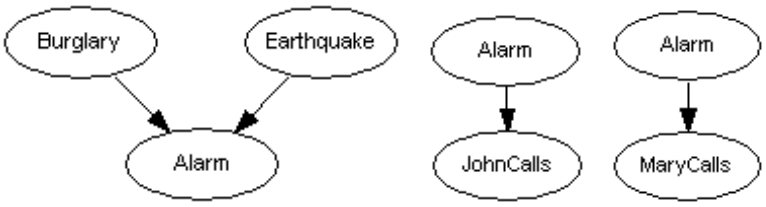

Figure 1: Generated Family Networks

The probability bounds improve the efficiency of the method enormously. Only a very small sub volume of the whole hypercube needs to be sampled now. In the case of the experiment the probability bounds create a new hypercube that has a volume of only

$6.6*10^{-27}$ times the whole hypercube volume. And only the intersection of the 7-simplex with this hypercube will be searched by the sample generator.

After the generation of the constraints and the calculation of the probability bounds the sampling process is started. After the method has found 1000 samples that satisfy all the constraints, the samples are used to calculate samples for the CPT entries of the FNs. The results of these calculations are histograms for the different CPT entries.

Conclusion

Some general observations can be made about a large number of the generated histograms. The shape of most of the histograms is similar. The histograms resemble a uniform distribution. One example is Figure 2(a), a histogram generated from the "strict" probability statement set for the Alarm node.

A likely cause for this shape to be present in so many histograms is the use of uniform sampling on the simplex. The generated samples of the joint probability distributions are all as likely to occur, so when calculating samples for the CPTs these samples will also show the same characteristics.

Not all the histograms of the two experiments have a uniform shape. In the smaller family networks consisting out of the nodes Alarm and JohnCalls or Alarm and MaryCalls, the histograms created for the Alarm node did not have a uniform shape. An example is shown in figure 2(b).

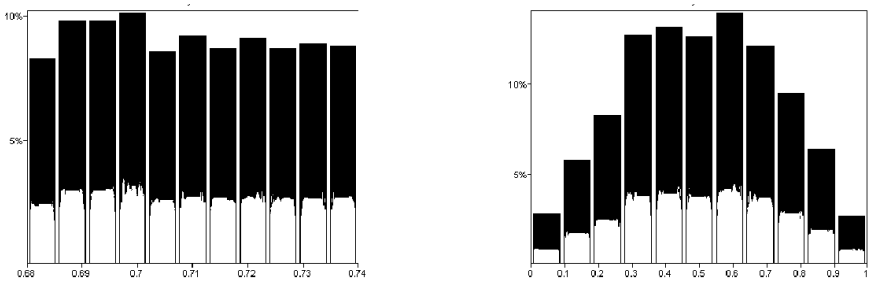

Figure 2: Histograms: (a) P(¬a|¬be), (b) P(a)

CONCLUSIONS AND FUTURE WORK

Work on the method is not completely done yet. Some research questions still need to be solved and the created implementation is not completely ready for integration with GeNIe, the software for developing Bayesian Networks created by the Decision Systems Laboratory. What has been accomplished is an implementation that proves the concept presented in [5]. It is indeed possible to let an expert state probabilistic information and to use this information to derive 2nd order distributions over all the CPT entries. But experimental results show that the method needs further fine-tuning, considering the results it currently generates. Even with very strictly specified constraints the generated histograms currently do not give very clear advice on what value to choose for a CPT entry. Furthermore the method has its limitations. A limiting factor for the method is the maximum number of parents a node has in a Bayesian network. A node with a large number of parents has a very large number of CPT entries, which in turn results in a sample space with a very high number of dimensions. One possible problem connected to sampling in a space with a high number of dimensions is a possible limitation of the accuracy of the implementation of the truncated Dirichlet distribution. Inaccuracies could possibly lead to a higher rejection ratio of samples. Another problem, more likely to occur, is that sets of probability statements provided by the expert can result in very few linear constraints. This can cause the linear programming process to be not very successful in reducing the size of the sample space. If the constrained volume is very small and the

remaining sample space is much larger, this will again cause a higher ratio of rejected samples, causing the method to become extremely slow.

The computation time the method needs to generate results completely depends on the input of the user. Factors that are present independent of user input, for example the requirement that every sample must be a valid probability distribution, have been dealt with as efficient as possible. Every generated sample by default satisfies the axioms of probability, and the ability to sample between probability bounds without the use of rejection sampling has increased the efficiency of generating samples by many orders of magnitude. When the proposed importance sampling scheme is implemented, the efficiency of generating samples may increase even more. It is even likely that, when using importance sampling, the method will perform better in the situation the generated probability bounds have not adequately reduced the sample space. Once valid samples are found the method will increasingly focus on the area where the valid samples were found.

REFERENCES

[1] CHINNECK, J. W. Discovering the Characteristics of Mathematical Programs via Sampling. *Optimization Methods and Software 17(2)* (2002), 319–352.

[2] CLEMEN, R. T., AND REILLY, T. *Making Hard Decisions with DecisionTools®*, 2 ed. Duxbury, Pacific Grove, CA, 2003.

[3] COOKE, N. J. Varieties of Knowledge Elicitation Techniques. *Int. J. Hum.-Comput. Stud. 41*, 6 (1994), 801–849.

[4] COOKE, R. *Experts in Uncertainty: Opinion and Subjective Probability in Science*. Oxford University Press, 1991.

[5] DRUZDZEL, M., AND VAN DER GAAG, L. C. Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)* (San Francisco, CA, 1995), Morgan Kaufmann Publishers, pp. 141–148.

[6] FANG, K.-T., GENG, Z., AND TIAN, G.-L. Statistical Inference for Truncated Dirichlet Distribution and its Application in Misclassification. *Biometrical Journal* (2000), 1053–1068.

[7] PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[8] SHAFER, G. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

[9] ZADEH, L. Fuzzy Sets as a Basis for a Theory of Possibility. *Fuzzy Sets and Systems 1* (1978), 3–28.

ABOUT THE AUTHORS

ir. ing. Martinus de Jongh, Man-Machine Interaction Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, E-mail: m.a.dejongh@gmail.com

dr. Marek Druzdzel, Decision Systems Laboratory, School of Information Sciences & Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA, USA, E-mail: marek@sis.pitt.edu

dr. drs. Leon Rothkrantz, Man-Machine Interaction Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, E-mail: L.J.M.Rothkrantz@ewi.tudelft.nl