Decomposing Local Probability Distributions in Bayesian Networks for Improved Inference and Parameter Learning

Adam Zagorecki, Mark Voortman and Marek J. Druzdzel
Decision Systems Laboratory
School of Information Sciences
and Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15213
{adamz,voortman,marek}@sis.pitt.edu

Abstract

A major difficulty in building Bayesian network models is the size of conditional probability tables, which grow exponentially in the number of parents. One way of dealing with this problem is through parametric conditional probability distributions that usually require only a linear number of parameters in the number of parents. In this paper we introduce a new class of parametric models, the pICI models, that aim at lowering the number of parameters required to specify local probability distributions, but are still capable of modeling a variety of interactions. A subset of the pICI models are decomposable and this leads to significantly faster inference as compared to models that cannot be decomposed. We also show that the pICI models are especially useful for parameter learning from small data sets and this leads to higher accuracy than learning CPTs.

Introduction

Bayesian networks (BNs) (Pearl 1988) have become a prominent modeling tool for problems involving uncertainty. Some examples from a wide range of their practical applications are medical diagnosis, hardware troubleshooting, user modeling, intrusion detection, and disease outbreak detection. BNs combine strong formal foundations of probability theory with an intuitive graphical representation of interactions among variables, providing a formalism that is theoretically sound, yet readily understandable for knowledge engineers and fairly easy to apply in practice.

Formally, a BN is a compact representation of a joint probability distribution (JPD). It reduces the number of parameters required to specify the JPD by exploiting independencies among domain variables. These independencies are typically encoded in the graphical structure, in which nodes represent random variables and lack of arcs represents probabilistic conditional independences. The parameters are specified by means of local probability distributions associated with variables. In case of discrete variables (the focus of this paper), the local probability distributions are encoded in the form of prior probabilities over nodes that have no parents in the graph and conditional probability tables (CPTs) for all other nodes. Specifying a series of CPTs instead of the JPD already heavily reduces the number of required parameters. However, the number of parameters required to specify a CPT for a node grows exponentially in the number of its parents. Effectively, the size of CPTs is a major bottleneck in building models and in reasoning with them. For example, assuming that all variables are binary, a CPT of a variable with 10 parents requires the specification of $2^{10} = 1,024$ probability distributions. If we introduce another parent, the number of required distributions will grow to $2,048$. This may be overwhelming for an expert if the distributions are elicited. If the distributions are learned from a small data set, there might not be enough cases to learn distributions for all the different parent configurations in a node (Oniśko, Druzdzel, & Wasyluk 2001).

In this paper, we introduce a new class of parametric models that require significantly fewer parameters to be specified than CPTs. The new models are a generalization of the class of *Independence of Causal Influence* (ICI) models (Heckerman & Breese 1996) (they call it the class of causal independence models), and their unique feature is that the combination function does not need to be deterministic. The combination function takes as input the values of parent variables and produces a value for the child variable. The most important property of the new class is that combination functions are potentially decomposable, which leads to substantial advantages in inference. We will denote the newly proposed class *pICI* or *probabilistic ICI*. Whenever confusion can be avoided, we will use the term *decompositions* to describe the new models. The pICI models, similarly to the existing class of ICI models with deterministic combination functions, have two main advantages. The first advantage is that inference may be faster (Díez & Galán 2003), because the decompositions result in smaller clique sizes in the joint tree algorithm (Zhang & Yan 1997). This becomes especially dramatic when the number of parents is large. The second advantage is that if we learn the decompositions instead of CPTs from a small data set, the resulting network is likely to be more faithful in representing the true underlying probability distribution, because a lower number of parameters will prevent the decompositions from overfitting the data.

The remainder of this paper is structured as follows. In the next section, we discuss ICI models and explain the decomposability of the combination function. Then we introduce the probabilistic ICI models. In the empirical evaluation sec-

tion, we show that inference in decomposed probabilistic ICI models is faster, and that learning from small data sets is more accurate.

Independence of Causal Influences (ICI) Models and Decompositions

The class of *Independence of Causal Influences* (ICI) models aims at reducing the number of parameters needed to specify conditional probability distributions. ICI models are based on the assumption that parent variables $X_1, \ldots, X_n$ act independently in producing the effect on a child variable Y. We can express the ICI assumption in a Bayesian network by explicitly representing the *mechanisms* that independently produce the effect on Y. The mechanisms are introduced to quantify the influence of each cause on the effect separately. So, if we assume this type of model, we only need to separately assess the probability distributions that describe mechanisms, and give a function for combining the results of the mechanisms. Figure 1(a) shows a Bayesian network for multiple causes $X_1, \ldots, X_n$ and an effect Y. In Figure 1(b) we see the same causes $X_1, \ldots, X_n$, but they produce their effect on Y indirectly through mechanism variables $M_1, \ldots, M_n$. The double circles indicate that the value of Y is generated by a deterministic function, which combines the outputs of the mechanisms. This is a fundamental assumption of the ICI models.

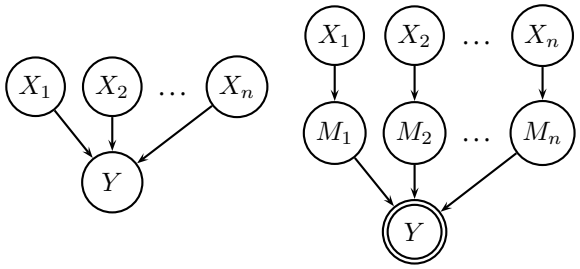

Figure 1: (a) A Bayesian network. (b) The class of ICI models.

An example of a well known ICI model is the noisy-OR gate (Pearl 1988, Henrion 1989), which reduces the number of parameters from exponential to linear in the number of parents. The CPTs for the mechanism variables in the noisy-OR model are defined as follows:

$$P(M_i = True|X_i) = \begin{cases} p_i \in [0, \ldots, 1], \text{ if } X_i = \text{True} ; \\ 0, \text{ if } X_i = \text{False} . \end{cases}$$

In the noisy-OR model, every variable has a *distinguished state*. Typically, this state indicates absence of a condition. If all the parents are in their distinguished states (i.e., are absent), then the child is also in its distinguished state. Note that the distinguished state is a property of the noisy-OR gate. Even though variables in most practical ICI models have distinguished states, it is not a strict requirement.

Node Y in Figure 1(b) is called the *combination function*, which in ICI is a deterministic function taking as input the values of the set of input variables and produces a value for the output variable. In case of noisy-OR, it is the deterministic OR function. If it is possible to decompose the combination function into a series of binary functions, the ICI model is said to be *decomposable*. An example of a decomposition is illustrated in Figure 2. In case of the noisy-OR gate, we can decompose the $OR(X_1, \ldots, X_n)$ function into $OR(X_1, OR(X_2, OR(\ldots OR(X_{n-1}, X_n)\ldots)))$. Heckerman and Breese (1994) showed empirically that this decomposition improves the efficiency of belief updating. The main reason for this improvement is a substantial reduction of the clique sizes in the joint tree algorithm (Lauritzen & Spiegelhalter 1988).

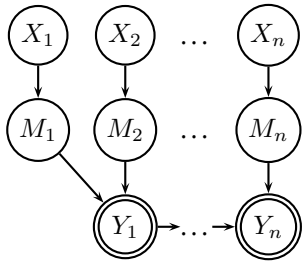

Figure 2: Decomposition of ICI models.

Probabilistic Independence of Causal Influences (pICI) Models

In this section, we propose a new class of models for modeling local probability distributions that is a generalization of the ICI models. The main difference is that we relax the assumption that the combination function is deterministic and allow the values in the CPT of the Y node to take values different from zero and one. Because of this, we call the new class the *probabilistic ICI* (pICI) models. We show the general model with an arbitrary combination function in Figure 3. In the following section, we take a look at a three models from the pICI class.

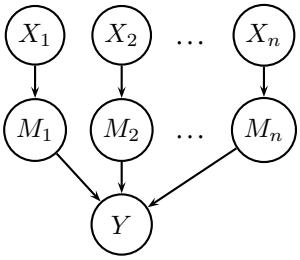

Figure 3: The class of pICI models.

The Average Model

An example of a pICI model that we propose in this paper is the *Average Model*. For this model, we chose a combination function that takes the average of the outputs of the mechanisms. It is important to realize that each mechanisn M_i has the same number of states as the Y node. For example, to

calculate the value of the first state in node Y, we count the number of mechanisms that are in state one, and divide it by the number of parents. The resulting value will be the probability for the first state in Y. We can repeat this process for all other states. Formally, the combination function for the Average model is given by:

$$P(Y = y|M_1, \ldots, M_n) = \frac{1}{n} \sum_{i=1}^{n} I(M_i = y) \ ,$$

where I is the indicator function that takes 1 when the condition in the brackets is true and 0 otherwise. Variables M_i and Y, as well as parent variables X_i, do not have to be binary.

The parameters of this model are expressed in terms of mechanisms — separate influences of a parent on the effect, and, therefore, they have meaning in the modeled domain, which is crucial for working with domain experts. The combination function is the average number of instantiations of mechanism variables. Such a setting has one important advantage over models like noisy-MAX (the multi-valued extension of noisy-OR) — it does not require additional semantic knowledge about the values (noisy-MAX assumes an ordering relation) and, therefore, can be easily applied to learning algorithms, as well as it is more flexible in terms of modeling.

Decomposable pICI models

The most important type of pICI models are those that are decomposable, similarly to the decomposable ICI models. The general decomposed form of the model is displayed in Figure 4. We call it the *Ladder Model* (LM).

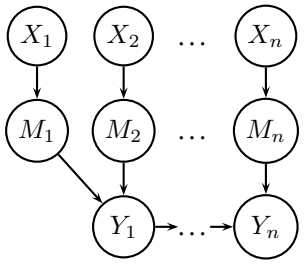

Figure 4: Decomposition of pICI models.

The Average model that we showed as an example of a pICI model is also decomposable. Formally, the decomposed form of the combination function is given by:

$$P(Y_i = y|Y_{i-1} = a, M_{i+1} = b)$$
$$= \frac{i}{i+1} I(y = a) + \frac{1}{i+1} I(y = b) \ ,$$

for $Y_2, \ldots, Y_{n-1}$ and I is again the indicator function. Y_1 is defined as:

$$P(Y_1 = y|M_1 = a, M_2 = b)$$
$$= \frac{1}{2} I(y = a) + \frac{1}{2} I(y = b) \ .$$

Decomposition	Number of parameters
CPT	$m_y \prod_{i=1}^{n} m_i$
LM	$(n-1)m_y^3 + m_y \sum_{i=1}^{n} m_i$
Average	$m_y \sum_{i=1}^{n} m_i$
SL	$m_1 m_2 m_y + m_y^2 \sum_{i=3}^{n} m_i$
Noisy-MAX	$m_y \sum_{i=1}^{n} (m_i - 1)$

Table 1: Number of parameters for the different decomposed models.

Figure 5 shows the *Simple Ladder* (SL) model which is basically a LM without the mechanism variables. This means that Y_i defines an interaction between the cumulative influence of the previous parents accumulated in Y_{i-1} and the current parent X_{i+1}. The SL model is similar to the decompositions proposed by Heckerman & Breese (1994) for the ICI model. The main differences are: (1) lack of a distinguished state in pICI models, and (2) the Y_i nodes are probabilistic rather than deterministic.

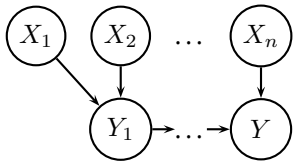

Figure 5: The Simple Ladder model.

The number of parameters required to specify relations between parents and the child variable for each of the models is shown in Table 1. Because m_y^3 is the dominating factor in case of the LM decomposition, LM is especially attractive in situations where the child variable has a small number of states and the parents have a large number of states. SL, on the other hand, should be attractive in situations where the parents have small numbers of states (the sum of the parents' states is multiplied by m_y^2).

Empirical Evaluation

Experiment 1: Inference

We compared empirically the speed of exact inference between CPTs and the new models, using the joint tree algorithm. We were especially interested in how the new models scale up when the number of parents and states is large compared to CPTs. We used models with one child node and a varying number of parents ranging from 5 to 20. We added arcs between each pair of parents with a probability of 0.1. Because the randomness of the arcs between the parents can influence the inference times, we repeated the procedure of generating arcs between parents 100 times and took the average inference time for the 100 instances. The last parameter to fix is the number of states in the variables and we subsequently used 2, 3, 4, and 5 states for all the variables. Because of the computational complexity, not all

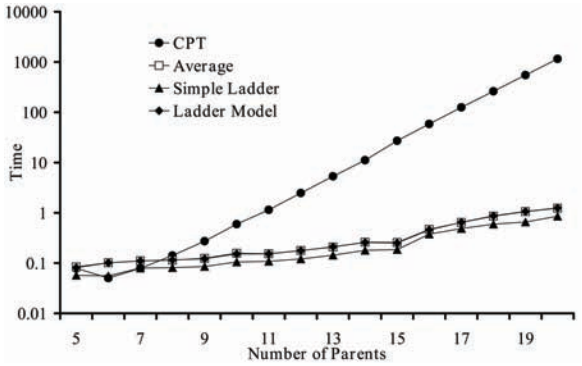

Figure 6: Inference results for the network where all variables have two states.

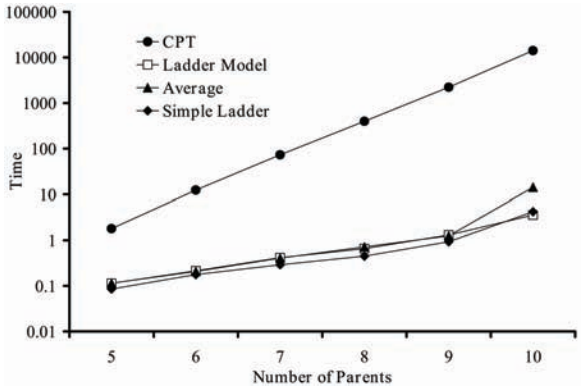

Figure 7: Inference results for the network where all variables have five states.

experiments completed to the 20 parents. When there was not enough memory available to perform belief updating in case of CPTs, we stopped the experiment.

The results are presented in Figures 6 and 7. We left out the results for 3 and 4 states, because these were qualitatively similar and only differed in the intersection with the y-axis. It is easy to notice that the decomposable models are significantly faster for a large number of parents, and the effect is even more dramatic when more states are used. The improvement in speed is substantial. Heckerman & Breese (1994) empirically showed that if decompositions are used in general BNs, it will speed-up inference.

Experiment 2: Learning

In this experiment, we investigated empirically how well we can learn the decompositions from small data sets. We selected 'gold standard' families (child plus parents) that had three or more parents from the following real-life networks (available at `http://genie.sis.pitt.edu/`): HAILFINDER (Edwards 1998), HEPAR II (Oniśko, Druzdzel, & Wasyluk 2001) and PATHFINDER (Heckerman, Horvitz, & Nathwani 1992). We generated a complete data set from each of the selected families. Because the EM algorithm requires an initial set of parameters, we scrambled randomly the prior parameters. We then relearned the parameters of the CPTs and decomposed models from the same data using the EM algorithm (Dempster, Laird, & Rubin 1977), repeating the procedure 50 times for different data sets. The number of cases in the data sets ranged from 10% of the parameters in the CPT, to 200%. For example, if a node has 10 parameters, the number of cases used for learning ranged from 1 to 20. In learning, we assumed that the models are decomposable, i.e., that they can be decomposed according to the LM, Average, and SL decompositions. The difference between the LM and Average model is that in the Average model the combination function is fixed, and in the LM we are learning the combination function. Note that the EM algorithm is especially useful here, because the decompositions will have hidden variables (e.g., the mechanism nodes). The EM algorithm is able to handle missing data. Our hypothesis is that the decompositions learn better than CPTs as long as the number of cases is low. We compared the original CPTs with the relearned CPTs, decompositions and noisy-MAX using the Hellinger's distance (Kokolakis & Nanopoulos 2001). The Hellinger distance between two probability distributions F and G is given by:

$$D_H(F,G) = \sqrt{\sum_i (\sqrt{f_i} - \sqrt{g_i})^2} \ .$$

To account for the fact that a CPT is really a set of distributions, we define a distance between two CPTs of node X as the sum of distances between corresponding probability distributions in the CPT weighted by the joint probability distribution over the parents of X. This approach is justified by the fact that in general it is desired to have the distributions closer to each other when the parent configuration is more likely. If this is the case, the model will perform well for the majority of cases.

We decided to use the Hellinger distance, because, unlike the Euclidean distance, it is more sensitive to differences in small probabilities, and it does not pose difficulties for zero probabilities, as is the case for Kullback-Leibler divergence (Kullback & Leibler 1951).

In order to do noisy-MAX learning, we had to identify the distinguished states. To find the distinguished states, we used a simple approximate algorithm to find both the distinguished states of the parents and the child. We based the selection of distinguished states on counting the occurrences of parent-child combinations N_{ij}, where i is the child state and j is the parent state. The next step was to normalize the child states for each parent: $N^*_{ij} = \frac{N_{ij}}{\sum_i N_{ij}}$. Child state i and parent state j are good distinguished state candidates if N^*_{ij} has a relatively high value. But we have to account for the fact that one child can have multiple parents, so we have to combine the results for each of the parents to determine the distinguished state of the child. For each parent, we select the maximum value of the state of a parent given the child state. We take the average of one of the child states over all the parents. The child state corresponding to the highest value of the average child states values is considered to be the child's distinguished state. Now that we have the child's distinguished state, it is possible to find the parents' distin-

Model	CPT	Average	SL	LM	MAX
Hepar	–	3	–	1	1
Hailfinder	–	1	4	1	–
Pathfinder	4	–	10	–	6

Table 2: Number of best fits for each of the networks for 2 cases per CPT parameter. For example, if the original CPT has 10 parameters, we used 20 cases to learn the models.

guished states in a similar way.

We ran the learning experiment for all families from the three networks in which the child node had a smaller number of parameters for all decomposition than the CPT. The results were qualitatively comparable for each of the networks. We selected three nodes, one from each network, and show the results in Figures 8 through 10. It is clear that the CPT network performs poorly when the number of cases is low, but when the number of cases increases, it comes closer to the decompositions. In the end (i.e., when the data set is infinitely large) it will fit better, because the cases are generated from CPTs. For node F5 from the PATHFINDER network, the Average model provided a significantly worse fit than the other models. This means that the Average model did not reflect the underlying distribution well. For other distributions, the Average model could provide a very good fit, while, for example, the noisy-MAX model performs poorly. Another interesting phenomenon is that in node F5 from the PATHFINDER network the parameters for the Average model were learned poorly. This is probably because the data comes from a distribution that cannot be accurately represented as the Average model. Again, it is important to emphasize that the pICI models performed better for almost all the decomposed nodes as is shown in the next paragraph.

Table 2 shows a summary of the best fitting model for each network. The number indicates for how many families a given model was the best fit for the situation when the number of cases was equal to two times the number of parameters in the CPT. We see that the selection of the best model is heavily dependent on the characteristics of the CPT — the distribution of the parameters and its dimensionality. However, in 27 of the 31 nodes, taken from the three networks, the decompositions (noisy-MAX included) performed better than CPTs. Also, the CPTs in our experiments relatively small — for HEPAR II it was roughly in the range of 100 to 400 parameters, for HAILFINDER 100 to 1200, and for PATHFINDER 500 to 8000. As we demonstrated in Experiment 1, our method scales to larger CPTs and we should expect more dramatic results there.

There is no general a priori criteria to decide which model is better. Rather these models should be treated as complementary and if one provides a poor fit, there is probably another model with different assumptions that fits better. We investigate how to address the problem of selecting an appropriate model in Experiment 3.

Experiment 3: Practical Application of Learning

One objection that could be made against our work is that in real-life we do not know the true underlying probability distribution. Hence, we have to use the available data for selecting the right ICI or pICI model. That is why we performed an experiment to test if it is possible to use the likelihood function of the data, to see which model fits the data best. The likelihood function is given by $l(\theta_{\text{Decomp}} : D) = P(D|\theta_{\text{Decomp}})$, where θ_{Decomp} denotes the parameters corresponding to a decomposition and D denotes the data.

We used cross-validation to verify if the likelihood func-

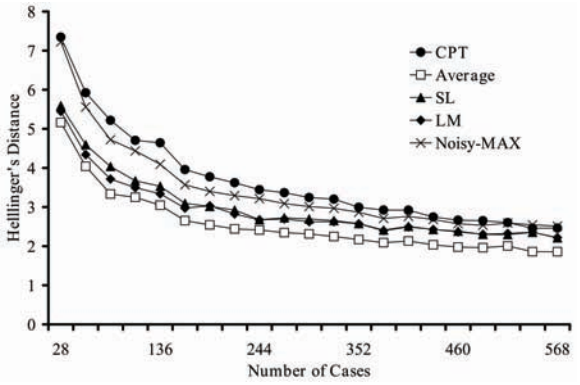

Figure 8: Results for the ALT node in the Hepar network.

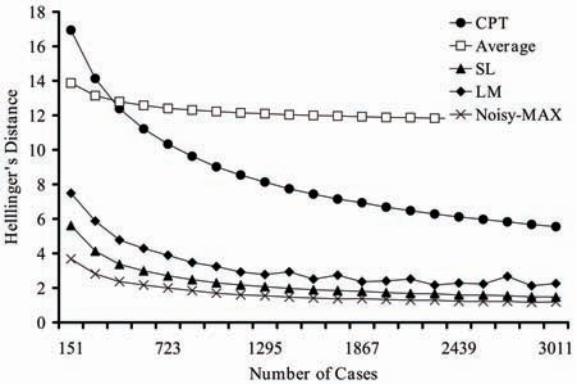

Figure 9: Results for the F5 node in the Pathfinder network.

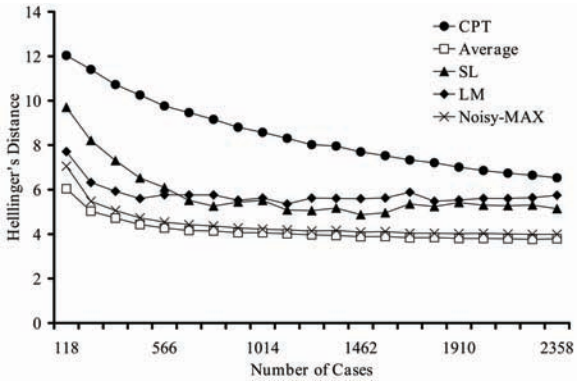

Figure 10: Results for the PlainFcst node in the HAILFINDER network.

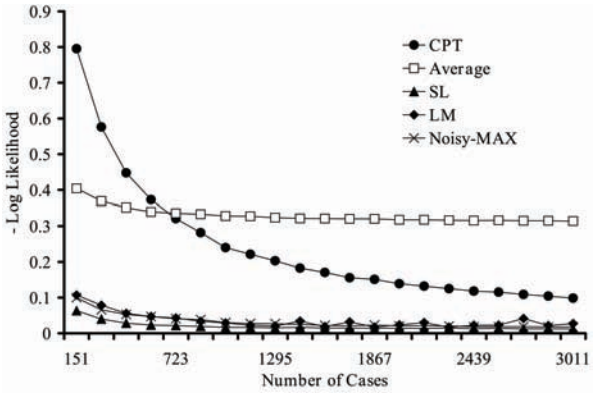

Figure 11: Likelihood for node F5.

tion is suitable to select the best decomposition. The experimental setup was the following. We used the same families as in experiment 1 and generated a data set from the gold standard model and split it into a training and test set. We used the training set to learn the model and a test data set of the same size as the training set to calculate the likelihood function. Figure 9 shows the Hellinger's distance for node F5, and Figure 11 shows the corresponding likelihood function. The shapes of the functions are essentially the same, showing that the likelihood function is a good predictor of model fit.

Conclusions

We introduced a new class of parametric models, the pICI models, that relax some assumptions of causal independence models and that allow for modeling wider variety of interactions. We proposed two pICI models, Ladder with Mechanisms and the Average model, and one derived model called Simple Ladder. The new models have a probabilistic combination function that takes the values of the input variables and produces a value for the output variable.

We focussed on a subset of the new class of models with decomposable combination functions. We showed the results of an empirical study that demonstrates that such decompositions lead to significantly faster inference. We also showed empirically that when we use these models for parameter learning with the EM algorithm from small data sets, the resulting networks will be closer to the true underlying distribution than what it would be with CPTs. Finally, we demonstrated that in real-life situations, we can use the likelihood function to select the decomposition that fits the model best.

Our models are intended for usage in real life models when a child node has a large number of parents and, therefore, the number of parameters in its CPTs is prohibitively large. In practice, this happens quite often, as is clear from the Bayesian networks that we used in our experiments.

Acknowledgments

This research was supported by the Air Force Office of Scientific Research under grant F49620-03-1-0187 and by Intel Research. All experiments described in this paper were performed using the SMILE library, available at `http://genie.sis.pitt.edu/`. While we take full responsibility for any errors in this paper, we would like to thank Changhe Yuan for his comments on an earlier draft and the FLAIRS reviewers for questions that prompted us for increasing the clarity of the paper.

References

Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* B(39):1–38.

Díez, F. J., and Galán, S. F. 2003. Efficient computation for the noisy MAX. *Int. J. Intell. Syst.* 18(2):165–177.

Edwards, W. 1998. Hailfinder: Tools for and experiences with Bayesian normative modeling. *American Psychologist* 53:416–428.

Heckerman, D., and Breese, J. S. 1994. A new look at causal independence. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, 286–292. San Francisco, CA: Morgan Kaufmann Publishers.

Heckerman, D., and Breese, J. 1996. Causal independence for probability assessment and inference using Bayesian networks. In *IEEE, Systems, Man, and Cybernetics*. 26:826–831.

Heckerman, D. E.; Horvitz, E. J.; and Nathwani, B. N. 1992. Toward normative expert systems: Part I. The Pathfinder Project. *Methods of Information in Medicine* 31:90–105.

Henrion, M. 1989. Some practical issues in constructing belief networks. In Kanal, L.; Levitt, T.; and Lemmer, J., eds., *Uncertainty in Artificial Intelligence 3*. New York, N. Y.: Elsevier Science Publishing Company, Inc. 161–173.

Kokolakis, G., and Nanopoulos, P. 2001. Bayesian multivariate micro-aggregation under the Hellinger's distance criterion. *Research in Official Statistics* 4(1):117–126.

Kullback, S., and Leibler, R. 1951. On information and sufficiency. *Ann. Math. Stat.* 22:79–86.

Lauritzen, S. L., and Spiegelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)* 50(2):157–224.

Oniśko, A.; Druzdzel, M. J.; and Wasyluk, H. 2001. Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates. *International Journal of Approximate Reasoning* 27(2):165–182.

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

Zhang, N., and Yan, L. 1997. Independence of causal influence and clique tree propagation. In *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, 481–488. San Francisco, CA: Morgan Kaufmann Publishers.