International Journal of Approximate Reasoning 70 (2016) 123–136

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Modeling women's menstrual cycles using PICI gates in Bayesian network

Adam Zagorecki [a,*], Anna Łupińska-Dubicka [b], Mark Voortman [c], Marek J. Druzdzel [b,c]

[a] Operational and Decision Analysis Group, Cranfield University, Defence Academy of the United Kingdom, Shrivenham, SN6 8LA, United Kingdom
[b] Faculty of Computer Science, Białystok University of Technology, Wiejska 45A, 15-351 Białystok, Poland
[c] Decision Systems Laboratory, School of Information Sciences and Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA

ARTICLE INFO

Article history:
Received 1 April 2015
Received in revised form 27 November 2015
Accepted 2 December 2015
Available online 10 December 2015

Keywords:
Bayesian networks
PICI gates
Modeling
Parameter learning
Inference

ABSTRACT

A major difficulty in building Bayesian network (BN) 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 number of parameters that is linear in the number of parents. In this paper, we introduce a new class of parametric models, the Probabilistic Independence of Causal Influences (PICI) models, that aim at lowering the number of parameters required to specify local probability distributions, but are still capable of efficiently modeling a variety of interactions. A subset of PICI models is decomposable and this leads to significantly faster inference as compared to models that cannot be decomposed. We present an application of the proposed method to learning dynamic BNs for modeling a woman's menstrual cycle. We show that PICI models are especially useful for parameter learning from small data sets and lead to higher parameter accuracy than when learning CPTs.

1. Introduction

Bayesian networks (BNs) [1] have become a prominent tool for modeling 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

* Corresponding author.
E-mail addresses: a.zagorecki@cranfield.ac.uk (A. Zagorecki), a.lupinska@pb.edu.pl (A. Łupińska-Dubicka), mark@voortman.name (M. Voortman), marek@sis.pitt.edu (M.J. Druzdzel).

http://dx.doi.org/10.1016/j.ijar.2015.12.002
0888-613X/

prior probability tables for 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, under the assumption that there are substantial number of conditional independencies encoded in the graph structure. 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 $2^{10} = 1024$ probability distributions. Introducing just one additional parent increases this number to 2048. This may be overwhelming for an expert if the distributions are elicited. The same problem applies to learning from data: 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 [2].

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 [3] (another popular, although less precise name is *causal independence models*). In order to explain the new class of models, we should start with defining the concept of *combination function*. The combination function maps states of the parent variables into the states of the child variables, and is equivalent to the definition of the CPT of the child variable. For the ICI models, the combination function is assumed to be deterministic (and in practice corresponds to the truth-table of a corresponding logical operator, e.g., OR or AND, which gives the name for ICI models). However, it is possible to relax the assumption of deterministic dependency between the parents' states and the child states, allowing for non-deterministic relations. We will denote the newly proposed class *PICI* or *probabilistic ICI*. The most important property of the new class is that combination functions are potentially decomposable, which leads to substantial advantages in inference. 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 [4], because the decomposability property of the combination function results in smaller clique sizes in the junction tree algorithm [5]. 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 will be more faithful if the PICI assumptions are satisfied in representing the true underlying probability distribution if the PICI assumptions are satisfied, because a lower number of parameters will prevent the decompositions from over-fitting the data.

The remainder of this paper is structured as follows. Sections 2 and 3 discuss ICI models and explain the concept of decomposability of the combination function, respectively. Section 4 introduce the PICI models. Empirical evaluation is divided in two sections. Section 5 shows that inference in decomposed probabilistic ICI models is faster and that learning from small data sets is more accurate on benchmark BN models. Section 6 applies the ideas introduced in the paper to modeling a woman's menstrual cycle using dynamic BNs, and demonstrates that PICI gates lead to improved inference and prediction performance.

2. Dynamic Bayesian networks

A dynamic Bayesian network (DBN) is an acyclic directed graphical model of a stochastic process. The network is arranged into time-slices and each of the time-slice contains its own nodes, edges, and probabilities. A time-slice can be seen as an snapshot of the evolving temporal process. We can say that the DBN consists of a sequence of sub-models, each representing the system at a particular point of time. Time-slices are connected by temporal relations, which are represented by arcs joining individual variables from two consecutive time-slices. Arcs between slices flow forward in time.

In a DBN, the state of a system at time t is represented by a set of random variables $\mathbf{Z}^t$. The state at time t generally depends on the states at previous time steps. Typically, we assume that each state only directly depends on the immediately preceding state (i.e., the system is first-order Markov), and thus we represent the transition distribution as follows [6]:

$$P(\mathbf{Z}^t|\mathbf{Z}^{1:t-1}) = P(\mathbf{Z}^t|\mathbf{Z}^{t-1}) = \prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t)) ,$$

where Z_i^t is the i-th node at time t and $Pa(Z_i^t)$ denotes the parents of Z_i^t from the same or from the previous time-slice.

The joint probability distribution for a sequence of length T can be obtained by *unrolling* the network:

$$P(\mathbf{Z}^{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t)) .$$

In case of kth order model, we represent the transition distribution as follows:

$$P(\mathbf{Z}^t|\mathbf{Z}^{1:t-1}) = P(\mathbf{Z}^t|\mathbf{Z}^{t-1}, \mathbf{Z}^{t-2}, \ldots, \mathbf{Z}^{t-k}) = \prod_{i=1}^{n} P(Z_i^t|Pa(Z_i^t)) .$$

In this case, the set of parents $Pa(Z_i^t)$ can contain nodes not only from the previous time-slice, but also from the previous k time-slices. The joint probability distribution for a sequence of length T can be obtained by *unrolling* the network in the exactly same manner as previously:

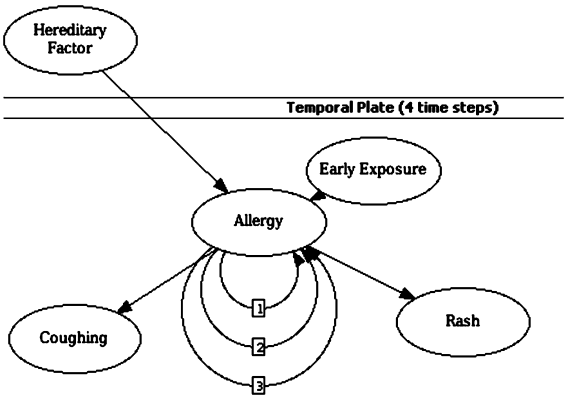

Fig. 1. A third order dynamic Bayesian network modeling causes and effects of allergy in children. *Number of slices* is the number of steps for which we perform the inference. In this example, one step means one year. Temporal plate is the part of dynamic network that contains temporal nodes. *Hereditary Factors* is time invariant; the values of remaining the nodes can change over time.

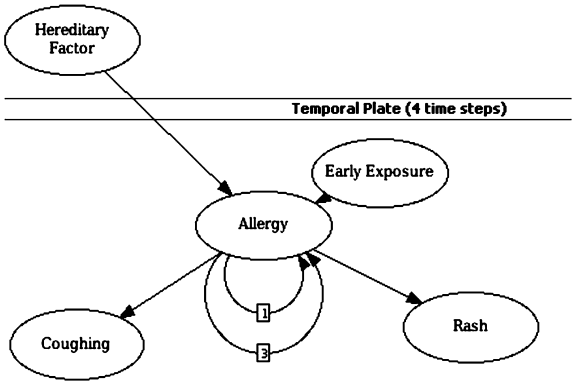

Fig. 2. An example of a 3rd order DBN model of woman's monthly cycle with only selected order influences.

$$P(\mathbf{Z}^{1:T}) = \prod_{t=1}^{T}\prod_{i=1}^{n} P(Z_i^t | Pa(Z_i^t)) \,.$$

As an example, consider the third order DBN shown in Fig. 1, illustrating various causes and effects of allergy in children. All variables in this example are Boolean. The tendency to develop allergies has a hereditary component: allergic parents are more likely to have allergic children, whose allergies are likely to be more severe than those from non-allergic parents. Exposure to allergens, especially in early life when the immune system is not yet fully developed, is also an important risk factor for allergy. When an allergen enters the body of an allergic child, the child can cough or develop a rash. The *Allergy* variable depends not only on *Hereditary factors* and *Early exposure* but also on its value from previous time steps.

The number of slices is the number of steps for which we perform the inference. The unit for the time step that is chosen varies on considered phenomenon. In this example, one step could mean one year. Temporal plate is the part of a DBN that contains nodes changing over time. *Hereditary Factors* do not change over time and, hence, the node is outside of the temporal plate. Such representation avoids copying time invariant nodes to every time-slice when unrolling the DBN for inference.

Furthermore a model of order k does not need to include influences of all orders between 1 and $k-1$. Fig. 2 shows a third order DBN, containing only selected temporal arcs.

Please note that models in Figs. 1 and 2 are both named third order DBN. However, they differ both in qualitative and quantitative part. The model pictured in Fig. 2 has a smaller number of temporal arcs, and consequently its CPT tables are also smaller. In this case, the transition distribution is represented as follows:

$$P(\mathbf{Z}^t | \mathbf{Z}^{1:t-1}) = P(\mathbf{Z}^t | \mathbf{Z}^{t-1}, \mathbf{Z}^{t-3}) = \prod_{i=1}^{n} P(Z_i^t | Pa(Z_i^t)) \,.$$

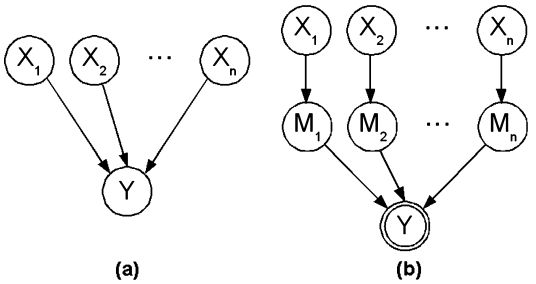

Fig. 3. (a) A Bayesian network. (b) The class of ICI models.

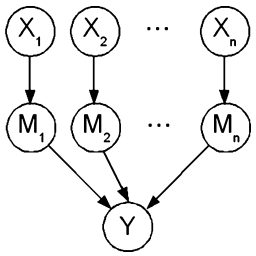

Fig. 4. Decomposition of ICI models.

3. 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 from one another 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. Fig. 3(a) shows a Bayesian network for multiple causes $X_1, \ldots, X_n$ and an effect Y. In Fig. 3(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.

An ICI model with n parent nodes can be defined as a set of n mechanism nodes M_i with each mechanism node having exactly one parent node X_i, and the combination node Y with n parent nodes $M_1, \ldots, M_n$ and a deterministic combination function.

The most popular ICI model is the noisy-OR gate [1,7], which reduces the number of parameters from exponential to linear in the number of parents. The CPTs for the i-th mechanism variables in the noisy-OR model are defined as follows:

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

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 necessary.

Let us define the *combination function* as the CPT of node Y as presented in Fig. 3(b). In the case of ICI models, the combination function 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 equivalent to 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 Fig. 4. In case of the noisy-OR gate, we can decompose the $OR(M_1, \ldots, M_n)$ function into $OR(M_1, OR(M_2, OR(\ldots OR(M_{n-1}, M_n)\ldots)))$. Heckerman and Breese [8] show empirically that this decomposition can significantly improve the efficiency of belief updating. The main reason for this improvement is a substantial reduction of the clique sizes in the junction tree algorithm [9].

4. 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

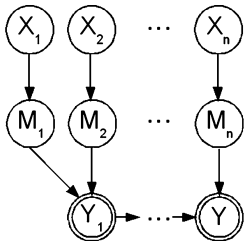

Fig. 5. The class of PICI models.

allow the values in the CPT of the Y node to take values from range $[0, 1]$. Because of this, we call the new class the *probabilistic ICI* (PICI) models.

A PICI model with n parent nodes can be defined as a set of n mechanism nodes M_i with each mechanism node having exactly one parent node X_i, and a non-deterministic combination node Y with n parent nodes $M_1, \ldots, M_n$ and a combination function (CPT of node Y) that allows for arbitrary probabilities.

We show the general model with an arbitrary combination function in Fig. 5. In the following section, we take a look at three models from the PICI class.

4.1. The Average Model

In the *Average Model* the PICI combination function takes the average of the outputs of the mechanisms. It is important to realize that each mechanism M_i has the same number of states as the Y node. To calculate the value of the first state in node Y, we count the number of mechanisms that are in the first state, and divide it by the number of parents. The resulting value will be the probability for the first state in Y. We 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.

Dagum and Galper [10] proposed a model that is similar to the Average model. However, there are a number of differences. Their models are strictly not ICI models nor PICI models, because they are not defined in form of mechanisms. Another difference is that their models do not assume parameter independence as their parameterization is in the form of marginal probabilities $P(Y|X_i)$ incorporates information about statistical dependencies between parent variables X, which makes learning of these models more complicated. More in-depth discussion can be found in [11], page 43.

4.2. Decomposable PICI models

From the practical perspective, the most interesting type of PICI models are those that are decomposable, similarly to the decomposable ICI models. The general form of a decomposable PICI model is displayed in Fig. 6(a).

The *Ladder Model* (LM) is a PICI model with the combination function:

$$f(M_1, \ldots, M_n)$$

that can be decomposed into a series of function

$$f_1(M_1, f_2(M_2, f_3(\ldots f_{n-1}(M_{n-1}, M_n)\ldots))).$$

The Average model that we introduced in the previous section is also decomposable. Formally, the decomposed form of the combination function is given by:

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

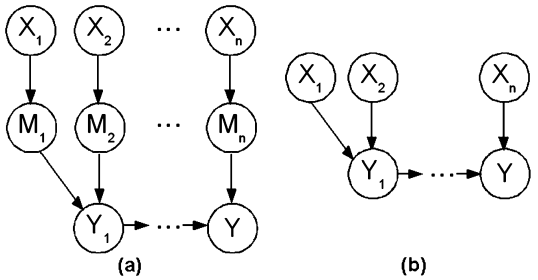

Fig. 6. (a) Decomposition of PICI models, (b) the Simple Ladder model.

Table 1
Number of parameters for the different decomposed models.

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	$m_y \sum_{i=1}^{n} (m_i - 1)$

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

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

Fig. 6(b) 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 [8] for the ICI model. The main differences between Heckerman's proposal and PICI models are: (1) lack of a distinguished state in the PICI models, and (2) the Y_i nodes are probabilistic rather than deterministic.

We show the number of parameters required to specify relations between parents and the child variable for each of the models 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 large numbers 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).

5. Empirical evaluation

5.1. Experiment 1: Inference

We compared empirically the speed of exact inference between CPTs and the PICI models, using the junction 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 randomly added arcs between each pair of parents with a probability of 0.1. We repeated the procedure of generating arcs between parents 100 times and took the average inference time for the 100 instances. The number of states in all variables was set to 2, 3, 4, and 5. Because of the computational complexity, not all experiments completed to the 20 parents. For each case when there was not enough memory available to perform belief updating in case of CPTs, we terminated the experiment. All experiments were executed on the same computer with 2.5 GHz Pentium 4 processor (with hyper-threading) and 786 MB RAM and using the Junction Tree algorithm implemented in SMILE Bayesian network library.

The results are presented in Figs. 7 and 8. 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. 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. The results are consistent with those of Heckerman and Breese [8], who show empirically that decompositions in general BNs can significantly speed-up inference.

5.2. 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

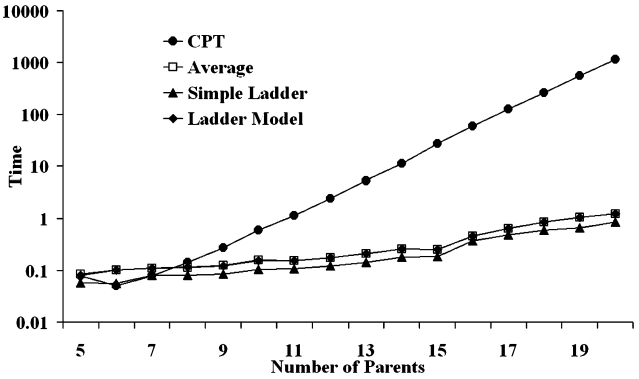

Fig. 7. Inference results for the network where all variables have two states.

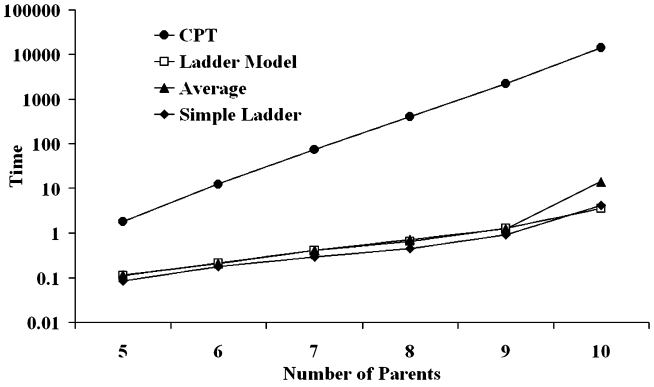

Fig. 8. Inference results for the network where all variables have five states.

(available at http://genie.sis.pitt.edu/): HAILFINDER [12], HEPAR II [2] and PATHFINDER [13]. We generated a data set with complete data from each of the selected families. Because the EM algorithm requires an initial set of parameters, we randomized the prior parameters. We then relearned the parameters of the CPTs and decomposed models from the same data using the EM algorithm [14], 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 CPTs, 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 lead to more accurate models than CPTs, especially for small data sets, i.e., when the number of cases is low. We compared the original CPTs with the relearned CPTs, decompositions and noisy-MAX using the Hellinger distance [15], which accounts for relative differences between probabilities. Hellinger distance is similar to Kullback–Leibler divergence [16], while being free from the disturbing property of the latter of being undefined for zero probabilities. 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.

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

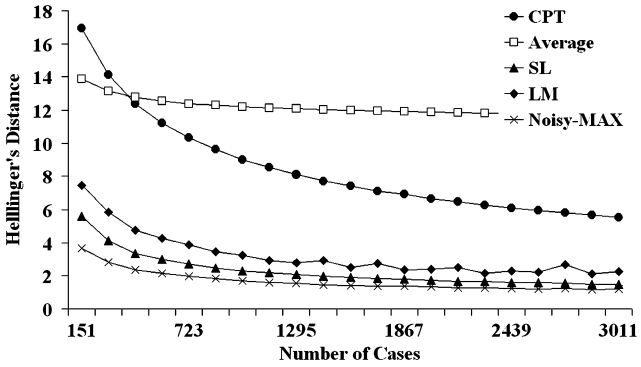

Fig. 9. Results for the F5 node in the Pathfinder network.

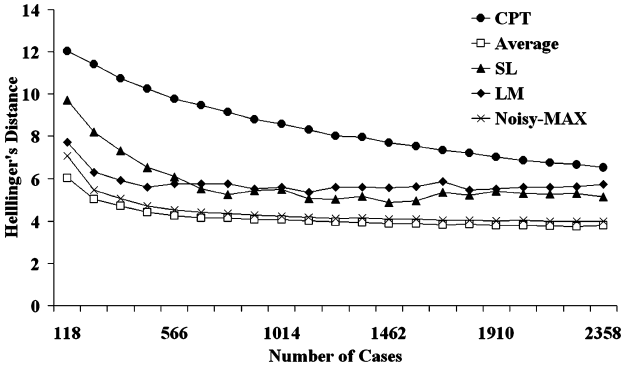

Fig. 10. Results for the PlainFcst node in the HAILFINDER network.

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' distinguished 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 models (SL, the noisy-MAX, etc.) than the CPT. The results were qualitatively comparable for each of the networks. We selected three representative examples and show the results in Figs. 9, 10, and 11. It is clear that the CPT network performs poorly when the number of cases is low, but when the number of cases increases, it's accuracy comes closer to the decompositions. In the limit (i.e., when the data set is infinitely large) it should fit data better, because the data are generated from CPTs. For node F5 of 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 performed poorly. 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 models for each network. The numbers indicate 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 were relatively small — for HEPAR II their sizes were 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 well to larger CPTs and we should expect even better results there.

There are 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.

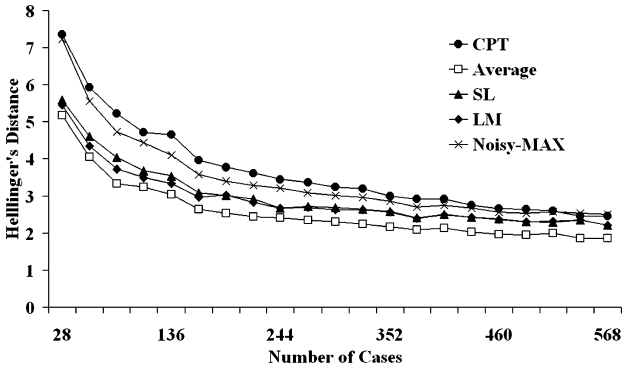

Fig. 11. Results for the Alt node in the HEPAR II network.

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.

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

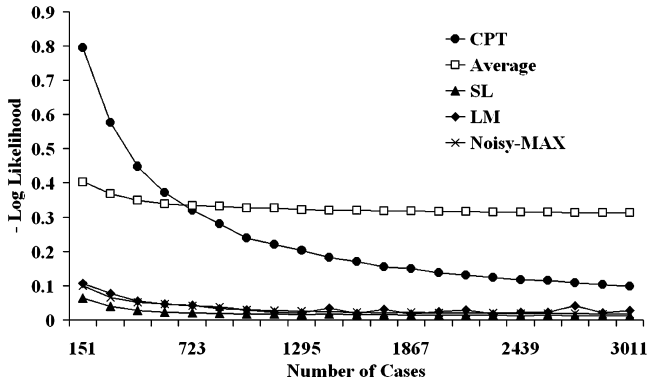

Fig. 12. Likelihood for node F5.

5.3. 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 function 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 a test set. We used the training set to learn the model and the test data set of the same size as the training set to calculate the likelihood function. Fig. 9 shows the Hellinger's distance for node F5, and Fig. 12 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.

6. Empirical evaluation 2

As the basis of our second empirical evaluation we used a dynamic Bayesian network (DBN) for monitoring a woman's menstrual cycle, described originally in [17]. The model uses a DBN learned from data to predict the time frame around ovulation when the probability of conception is high, given primary fertility signs such as body temperature, bleeding, and mucus constitution. One of the key modeling challenges for this problem is the substantial variation among women in terms of the cycle length (typically between 25 and 35 days), and the ovulation day (typically between 10th and 20th day),

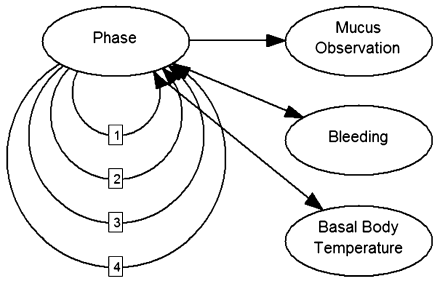

Fig. 13. An example of a 4th order CPT model of a woman's menstrual cycle.

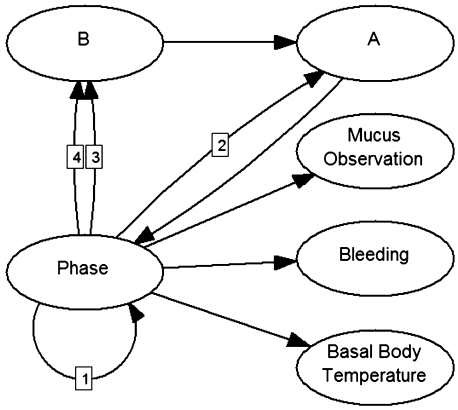

Fig. 14. An example of a 4th order SL model of a woman's menstrual cycle.

thereby making models based on population data insufficient to make accurate predictions. To make better predictions, one approach is to use data on a particular woman's past observed periods. However, those kind of data are always limited because of the time required to collect them. This problem makes it a suitable application of PICI because (1) the DBN models used in the original research suffered from large size CPTs and (2) the data are characterized by small sample sizes. For the study we selected the simple ladder model, as they offered the smallest number of parameters to learn among other models (excluding the noisy-OR).

The data used for learning came from a multinational European fecundability study [18]. From the original set of 7017 menstrual cycles for 881 women, in our experiments we used data on 3423 menstrual cycles of 236 women. We selected only those women for whom there were at least 7 cycles recorded. The BNs used in our experiments were based on the methodology and data described in [17] where the authors compared fertility awareness methods (FAMs). It is widely accepted that a woman needs at least six cycles to predict ovulations days reliably. We also excluded all cycles that did not have uniquely identified mucus peak or basal body temperature shift days. FAMs rely on at least one of these indicators and it would impossible to identify the fertile days for those cycles that do not have the peak day uniquely identified. Our goal was to predict fertile and infertile days.

6.1. Experiment 1

In the first experiment, we empirically compared the inference speed and time required for model learning between CPTs and the SL model. We were particularly interested in how the new models scale up with the growing number of time slices included in the model.

For each woman, we created seven models for both CPT and SL, each with different temporal order t, expressed in days, that ranged from 1 to 7. An example of a model with $t = 4$ based on CPT is shown in Fig. 13 and the corresponding model for SL is shown in Fig. 14. For the SL model, nodes A and B represent hidden nodes $Y_1, \ldots, Y_n$ from Fig. 6(b). The parent order was temporal with lower indices in Fig. 6(b) corresponding to lower temporal orders. One should note that in case of models with the first and second temporal order, the CPT and SL models are the same. Additionally, for each woman we created five models with selected temporal arcs: 1, 6, 10; 1, 6, 10, 15; 1, 3, 7, 10, 12; 1, 7–12; and 7–12, where for example 1, 6, 10 corresponded to a DBN of order 10 with three temporal arcs for $t = 1, 6$ and 10: $P(\mathbf{Z}^t|\mathbf{Z}^{t-1}, \mathbf{Z}^{t-6}, \mathbf{Z}^{t-10})$. These particular models were identified as well performing DBN models for that set of data.

The learning procedure was as follows: We learned initial model parameters based on the whole population using 5-fold cross validation. We used these population model parameters as the *a priori* parameters in all woman-specific models. In

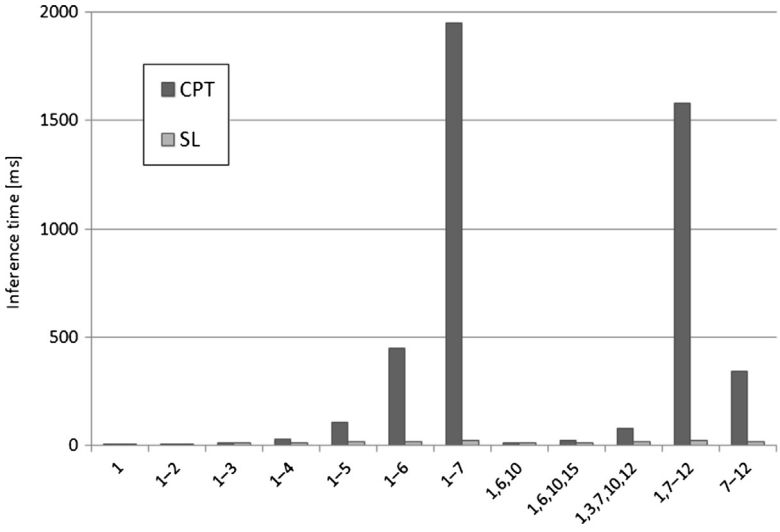

Fig. 15. Average time of inference.

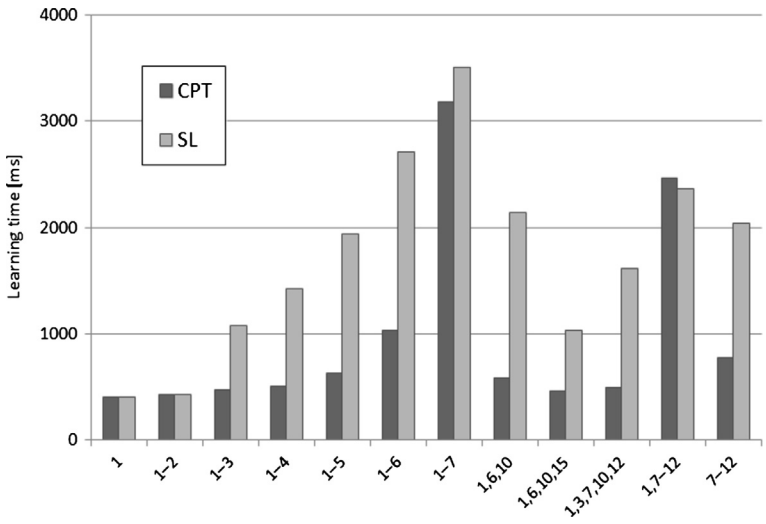

Fig. 16. Average time of learning.

order to learn a woman-specific model we updated model parameters using data cases specific for that woman. We used leave-one-out method at this step.

Fig. 15 shows that, as the temporal order increases, the average inference time for models with SL is significantly shorter that for the corresponding CPT models. The time for CPTs seems to grow exponentially with the number of parent nodes, while only linearly for SL. This trend is particularly evident for the temporal orders 5 and higher.

The average time of learning for SL models is longer in most cases. However this trend disappears with higher temporal orders (see Fig. 16). For the temporal order 8, the SL model was computed while the CPT model was not. This should be attributed to the use of the EM algorithm and the fact that our data did not have missing values — in case of SL, the EM algorithm must deal with missing values introduced by the PICI model, while for the CPT it does not.

From a practical perspective, the inference time is more critical than learning time, as the models could be updated only once a month, while queries can be expected much more often. This makes the application of PICI practically appealing for that problem.

6.2. Experiment 2

In the previous experiment, we have shown the value of PICI gates from the computational perspective. In the second experiment, we investigate the predictive accuracy of models. The goal of the experiment was to simulate the use of the model in a realistic application. The model of a woman's monthly cycle can be used for two quite opposite purposes: (1) to avoid an unwanted pregnancy, or (2) to increase chances of pregnancy by intensifying intercourse during the fertile period.

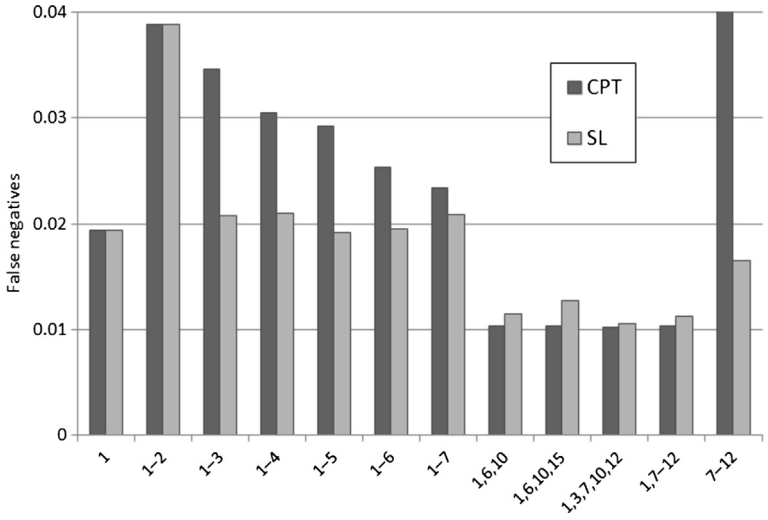

Fig. 17. Average percentage of false negatives.

For a model of menstrual cycle to be useful, it must correctly predict the ovulation day and, consequently, to determine the fertile window. We defined the days inside the fertile window that were classified as infertile are false negatives. The days that were identified as fertile and were outside the fertile window were defined as false positives.

To calculate the average percentage of false negatives and the average percentage of false positives we followed Wilcox et al. [19], who defines the fertile window as the period between the day of ovulation minus five days and the ovulation day plus one day. Days inside the fertile period that were classified as infertile are false negatives and days that were marked as fertile and were outside the fertile window are false positives. Please note that because of a possible application of a model like this in family planning, false negatives generally present a more serious type of error than false positives and, on one hand, may lead to unplanned pregnancy and, on the other hand, to decreasing the chance of conception in case of couples seeking pregnancy. Hence, we prefer to reduce the number of false negatives as much as possible. At the same time, the smaller the false positive rate, the closer the predicted day of ovulation is to the actual day of ovulation, which can be helpful for couples seeking pregnancy.

The fertile period of the menstrual cycle (fertile window) is defined as the time when an intercourse has a non-zero probability of resulting in conception. The number of fertile days during the monthly cycle is difficult to specify. However, it is useful and important to be able to predict beginning the fertile period as close to the ovulation as possible, assuming the minimal false negatives rate. In the experiment in Section 5, at every time step (i.e., every day of a cycle) each model computed the most probable day of the ovulation. If a time interval between the current and the day with the highest probability of the ovulation equaled at least six days (five days for life span of sperm and one more day to provide a safety margin against false negatives), we marked the current day as infertile. In all other cases the current day was the beginning of the fertile period.

We determined the number of fertile and infertile days marked by models in all cycles and divided this number by the total length of the cycle for each woman and for each cycle. Effectively, we obtained the percentage of all days that were misclassified as infertile (false negative rate) and percentage of all days that were misclassified as fertile (false positive rate).

The results are presented in Figs. 17 and 18. The SL models usually show a higher false positive rate. However, in most of the cases the SLs' false negative rates are lower than those for CPT models. This means that they more aligned with the user's objectives. The best results were achieved for models that combine different non-consecutive temporal orders (e.g., 1, 6, 10), with SL performing only slightly worse than CPTs, while showing considerable improvement in the inference performance.

7. Conclusions and discussion

We introduced a new class of parametric models, the PICI models, that relax some assumptions of causal independence models and allow for modeling a 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 focused 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. This results are in line with other related research — in particular Neil et al. [20] used the idea of decomposing local probability distributions to propose an efficient inference algorithm for hybrid Bayesian networks (combining discrete and continuous variables).

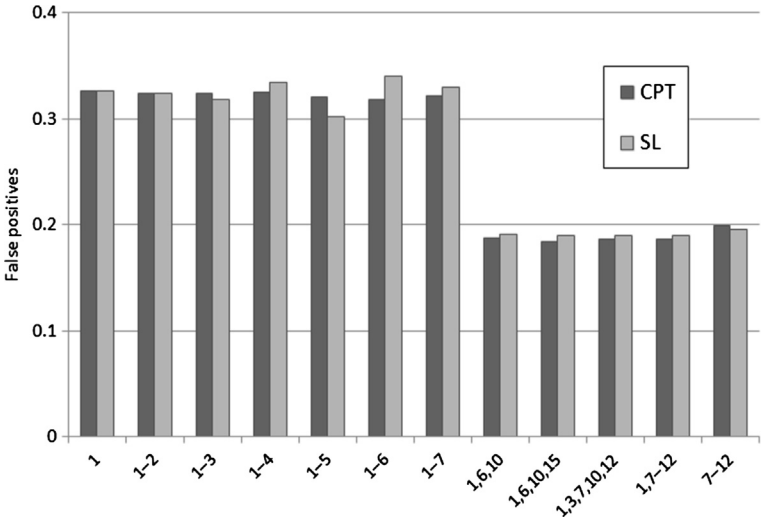

Fig. 18. Average percentage of false positives.

We used the EM algorithm to obtain parameters for the proposed models. The idea of using the EM algorithm has been exploited by other researchers [21] to estimate parameters of symmetric causal independence models. 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.

The proposed 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. To support this claim we provided an example from a domain where models with large CPTs (dynamic BNs) were learned from data and used to predict woman's menstrual cycle. We showed the benefits of applying PICI to that problem.

Acknowledgements

While we take full responsibility for any possible errors and inaccuracies, we would like to thank Changhe Yuan for his comments on an earlier draft of this paper. The authors would like to thank Bernardo Colombo, Guido Masarotto, Fausta Ongaro, Petra Frank-Herrmann, and other investigators of the European Study of Daily Fecundability for making the data used in our experiments available. The core of our implementation is based on the SMILE reasoning engine for graphical probabilistic model contributed to the community by the Decision Systems Laboratory, University of Pittsburgh and available at http://genie.sis.pitt.edu/. Some experiments were partially financed with the European Union funds as a part of the "Centre for Modern Education of the Bialystok University of Technology" project (Operational Programme Development of Eastern Poland) and the National Institute of Health under grant number U01HL101066-01. We would like to thank anonymous reviewers for the FLAIRS-2006 Conference, where we presented and published an earlier version of the first part of the paper [22], for their comments.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.
[2] A. Oniśko, M.J. Druzdzel, H. Wasyluk, Learning Bayesian network parameters from small data sets: application of Noisy-OR gates, Int. J. Approx. Reason. 27 (2) (2001) 165–182.
[3] D. Heckerman, J. Breese, Causal independence for probability assessment and inference using Bayesian networks, IEEE Transactions on Systems, Man, and Cybernetics 26 (6) (1996) 826–831.
[4] F.J. Díez, S.F. Galán, Efficient computation for the noisy MAX, Int. J. Intell. Syst. 18 (2) (2003) 165–177.
[5] N. Zhang, L. Yan, Independence of causal influence and clique tree propagation, in: Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), Morgan Kaufmann Publishers, San Francisco, CA, 1997, pp. 481–488.
[6] T. Dean, K. Kanazawa, A model for reasoning about persistence and causation, Comput. Intell. 5 (2) (1989) 142–150.
[7] M. Henrion, Some practical issues in constructing belief networks, in: L. Kanal, T. Levitt, J. Lemmer (Eds.), Uncertainty in Artificial Intelligence, vol. 3, Elsevier Science Publishing Company, Inc., New York, NY, 1989, pp. 161–173.
[8] D. Heckerman, J.S. Breese, A new look at causal independence, in: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94), Morgan Kaufmann Publishers, San Francisco, CA, 1994, pp. 286–292.
[9] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc. B 50 (2) (1988) 157–224.
[10] P. Dagum, A. Galper, Additive belief-network models, in: UAI, 1993, pp. 91–98.
[11] A. Zagorecki, Local probability distributions in Bayesian networks: knowledge elicitation and inference, Ph.D. thesis, University of Pittsburgh, 2010.
[12] W. Edwards, Hailfinder: tools for and experiences with Bayesian normative modeling, Am. Psychol. 53 (1998) 416–428.

[13] D.E. Heckerman, E.J. Horvitz, B.N. Nathwani, Toward normative expert systems: part I. The Pathfinder Project, Methods Inf. Med. 31 (1992) 90–105.
[14] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B 39 (1977) 1–38.
[15] G. Kokolakis, P. Nanopoulos, Bayesian multivariate micro-aggregation under the Hellinger's distance criterion, Res. Official Stat. 4 (1) (2001) 117–126.
[16] S. Kullback, R. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 79–86.
[17] A. Lupinska-Dubicka, M.J. Drużdżel, Modeling dynamic systems with memory: what is the right time-order?, in: The 8th Bayesian Modelling Applications Workshop, Barcelona, Spain, 2011, pp. 75–82.
[18] B. Colombo, G. Masarotto, Daily fecundability: first results from a new data base, Demogr. Res. 3 (5) (2000).
[19] A.J. Wilcox, C.R. Weinberg, D.D. Baird, Timing of sexual intercourse in relation to ovulation. Effects on the probability of conception, survival of the pregnancy, and sex of the baby, N. Engl. J. Med. 333 (23) (1995) 1517–1521.
[20] M. Neil, X. Chen, N. Fenton, Optimizing the calculation of conditional probability tables in hybrid Bayesian networks using binary factorization, IEEE Transactions on Knowledge and Data Engineering 24 (7) (2012) 1306–1312, http://dx.doi.org/10.1109/TKDE.2011.87.
[21] R. Jurgelenaite, T. Heskes, Learning symmetric causal independence models, Mach. Learn. 71 (2–3) (2008) 133–153, http://dx.doi.org/10.1007/s10994-007-5041-7.
[22] A. Zagorecki, M. Voortman, M.J. Druzdzel, Decomposing local probability distributions in Bayesian networks for improved inference and parameter learning, in: Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11–13, 2006, 2006, pp. 860–865.