Dynamic importance sampling in Bayesian networks using factorisation of probability trees

Irene Martínez
Department of Languages and Computation
University of Almería
Almería, 04120, Spain

Carmelo Rodríguez and Antonio Salmerón
Department of Statistics and Applied Mathematics
University of Almería
Almería, 04120, Spain

Abstract

Factorisation of probability trees is a useful tool for inference in Bayesian networks. Probabilistic potentials some of whose parts are proportional can be decomposed as a product of smaller trees. Some algorithms, like lazy propagation, can take advantage of this fact. Also, the factorisation can be used as a tool for approximating inference, if the decomposition is carried out even if the proportionality is not completely reached. In this paper we propose the use of approximate factorisation as a means of controlling the approximation level in a dynamic importance sampling algorithm.

1 Introduction

In this paper we propose an algorithm for updating probabilities in Bayesian networks. This problem is known to be NP-hard even in the approximate case (Dagum and Luby, 1993). There exist several deterministic approximate algorithms (Cano et al., 2000; Cano et al., 2002; Cano et al., 2003; Kjærulff, 1994) as well as algorithms based on Monte Carlo simulation. The two main approaches are: Gibbs sampling (Jensen et al., 1995; Pearl, 1987) and importance sampling (Cheng and Druzdzel, 2000; Hernández et al., 1998; Moral and Salmerón, 2005; Salmerón et al., 2000; Shachter and Peot, 1990).

A class of these simulation procedures is composed by the importance sampling algorithms based on approximate pre-computation (Hernández et al., 1998; Moral and Salmerón, 2005; Salmerón et al., 2000). These methods perform first a fast but non exact propagation, consisting of a node removal process (Zhang and Poole, 1996). In this way, an approximate 'a posteriori' distribution is obtained. In the second stage a sample is drawn using the approximate distribution and the probabilities are estimated according to the importance sampling methodology (Rubinstein, 1981). In the most recent algorithm (Moral and Salmerón, 2005) a dynamic approach is adopted, in which the sampling distributions are updated according to the information collected during the simulation.

Recently, a new approach to construct approximate deterministic algorithms was proposed in (Martínez et al., 2005), based on the concept of approximate factorisation of the probability trees used to represent the probability functions.

The goal of this paper is to incorporate the ideas contained in (Martínez et al., 2005) to the dynamic algorithm introduced in (Moral and Salmerón, 2005), with the aim of showing that the approximation level can be controlled by the factorisation alternatively to pruning the trees.

The rest of the paper is organised as follows: in section 2 we analyse the main features of the dynamic importance sampling algorithm in (Moral and Salmerón, 2005). Section 3 explain the concept of approximate factorisation of probability trees. Then, in section 4 we explain how both ideas can be combined in a new algorithm. The performance of the new algorithm is tested using three real-world Bayesian networks in section 5 and the paper ends with the conclusions in section 6.

2 Dynamic importance sampling in Bayesian networks

Along this paper we will consider a Bayesian network with variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ where each X_i is a discrete variable taking values on a finite set Ω_{X_i}. By $\Omega_{\mathbf{X}}$ we denote the state space of the n-dimensional random variable $\mathbf{X}$.

Probabilistic reasoning in Bayesian networks requires the computation of the posterior probabilities $p(x_k|\mathbf{e})$, $x_k \in \Omega_{X_k}$ for each variable of interest X_k, given that some other variables $\mathbf{E}$ have been observed to take value $\mathbf{e}$.

The posterior probability mentioned above can be expressed as

$$p(x_k|\mathbf{e}) = \frac{p(x_k, \mathbf{e})}{p(\mathbf{e})} \quad \forall x_k \in \Omega_{X_k} \ ,$$

and, since $p(\mathbf{e})$ is a constant value, the problem of calculating the posterior probability of interest is equivalent to obtaining $p(x_k, \mathbf{e})$ and normalising afterwards. If we denote by $p(\mathbf{x})$ the joint distribution for variables $\mathbf{X}$, then it holds that

$$p(x_k, \mathbf{e}) = \sum_{\Omega_{\mathbf{X} \setminus (\{X_k\} \cup \mathbf{E})}} p(\mathbf{x}) \ ,$$

where we assume that the k-th coordinate of $\mathbf{x}$ is equal to x_k and the coordinates in $\mathbf{x}$ corresponding to observed variables are equal to $\mathbf{e}$. Therefore, the problem of probabilistic reasoning can be reduced to computing a sum, and here is where the *importance sampling* technique takes part.

Importance sampling is a well known method for computing integrals (Rubinstein, 1981) or sums over multivariate functions. A straightforward way to do that could be to draw a sample from $p(\mathbf{x})$ and then estimate $p(x_k, \mathbf{e})$ from it. However, $p(\mathbf{x})$ is often unmanageable, due to the large size of the state space of $\mathbf{X}$, $\Omega_{\mathbf{X}}$.

Importance sampling tries to overcome this problem by using a simplified probability function $p^*(\mathbf{x})$ to obtain a sample of $\Omega_{\mathbf{X}}$. The estimation is carried out according to the next procedure.

Importance Sampling

1. FOR $j := 1$ to m (sample size)
   (a) Generate a configuration $\mathbf{x}^{(j)} \in \Omega_{\mathbf{X}}$ using p^*.
   (b) Compute a weight for the generated configuration as:

$$w_j := \frac{p(\mathbf{x}^{(j)})}{p^*(\mathbf{x}^{(j)})} \ . \qquad (1)$$

2. For each $x_k \in \Omega_{X_k}$, estimate $p(x_k, \mathbf{e})$ as $\hat{p}(x_k, \mathbf{e})$ obtained as the sum of the weights in formula (1) corresponding to configurations containing x_k divided by m.

3. Normalise the values $\hat{p}(x_k, \mathbf{e})$ in order to obtain $\hat{p}(x_k|\mathbf{e})$, i.e., an estimation of $p(x_k|\mathbf{e})$.

In (Salmerón et al., 2000; Moral and Salmerón, 2005), a sampling distribution is computed for each variable, so that p^* is equal to the product of the sampling distributions for all the variables. The sampling distribution for each variable can be obtained through a process of variable elimination. Assume that the variables are eliminated following the order $X_1, \ldots, X_n$, and that, before eliminating the first variable, H is the set of conditional distributions in the network. Then, the next algorithm obtains a sampling distribution for he i-th variable.

Get Sampling Distribution(X_i,H)

1. $H_i := \{f \in H | f \text{ is defined for } X_i\}$.

2. $f_i := \prod_{f \in H_i} f$.

3. $f'_i := \sum_{x \in \Omega_{X_i}} f_i$.

4. $H := H \setminus H_i \cup \{f'_i\}$.

5. RETURN (f_i).

Simulation is carried out in an order contrary to the one in which variables are deleted. Each variable X_i is simulated from its sampling distribution f_i. This function is defined for variable X_i and other variables already sampled. The potential f_i is restricted to the already obtained values of the variables for which it is defined, except X_i, giving rise to a function which depends only on X_i. Finally, a value for this variable is obtained with probability proportional to the values of this potential. If all the computations are exact, it was proved in (Hernández et al., 1998) that, following this procedure, we are really sampling with the optimal probability $p^*(\mathbf{x}) = p(\mathbf{x}|\mathbf{e})$. However, the result of the combinations in the process of obtaining the sampling distributions may require a large amount of space to be stored, and therefore approximations are usually employed, either using probability tables (Hernández et al., 1998) or probability trees (Salmerón et al., 2000) to represent the distributions.

In (Moral and Salmerón, 2005) an alternative procedure to simulate each variable was used. Instead of restricting f_i to the values of the variables already sampled, all the functions in H_i are restricted, resulting in a set of functions depending only on X_i. The sampling distribution is then computed by multiplying all these functions. If the computations are exact, then both distributions are the same, as restriction and combination commute. This is the basis for the dynamic updating procedure proposed in (Moral and Salmerón, 2005). Probabilistic potentials are represented by probability trees, which are approximated by pruning some of its branches when computing f_i during the calculation of the sampling distributions. This approximation is somehow corrected according to the information collected during the simulation: the probability value of the configuration already simulated provided by the product of the potentials in H_i must be the same as the probability value for the same configuration provided by f'_i. Otherwise, potential f_i is re-computed in order to correct the detected discrepancy.

3 Factorisation of probability trees

As mentioned in section 2, the approximation of the probability trees used to represent the probabilistic potentials consists of pruning some of their branches, namely those that lead to similar leaves (similar probability values), that can be substituted by the average, so that the error of the approximation depends on the differences between the values of the leaves corresponding to the pruned branches. Another way of taking advantage of the use of probability trees is given by the possible presence of proportionality between different subtrees (Martínez et al., 2002). This fact is illustrated in figures 1 and 2. In this case, the tree in figure 1 can be represented, without loss of precision, by the product of two smaller trees, shown in figure 2.

Note that the second tree in figure 2 does not contain variable X. It means that, when computing the sampling distribution for variable X, the products necessary to obtain function f_i would be done over smaller trees.

3.1 Approximate Factorisation of Probability Trees

Factorisation of probability trees can be utilised as a tool for developing approximate inference algorithms, when the proportionality condition is relaxed. In this case, probability trees can be decomposed even if we find only *almost* proportional, rather than proportional, subtrees. Approximate factorisation and its application to Lazy propagation was proposed in (Martínez et al., 2005), and is stated as follows. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two sub-trees which are siblings for a given context (i.e. both sub-trees are children of the same node), such that both have the same size and their leaves contain only strictly positive numbers. The goal of the *approximate factorisation* is to find a tree $\mathcal{T}_2^*$ with the same structure than $\mathcal{T}_2$, such that $\mathcal{T}_2^*$ and $\mathcal{T}_1$ become proportional, under the restriction that the potential represented by $\mathcal{T}_2^*$ must be as close as possible to the one represented by $\mathcal{T}_2$. Then, $\mathcal{T}_2$ can be replaced by $\mathcal{T}_2^*$ and the resulting tree that

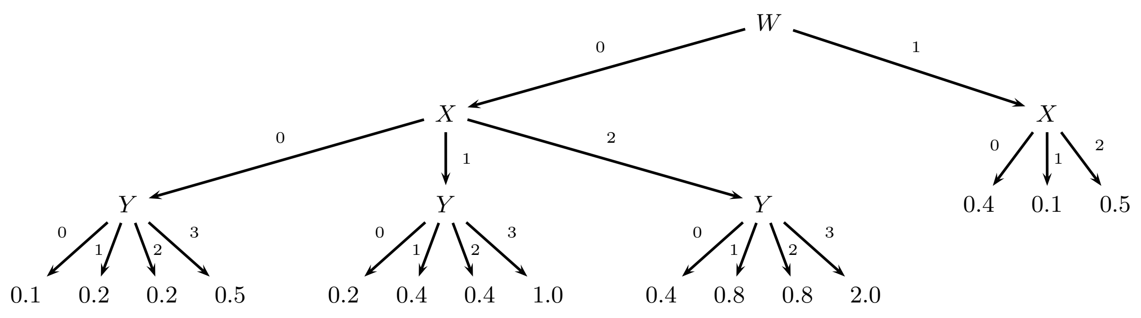

Figure 1: A probability tree proportional below X for context $(W = 0)$.

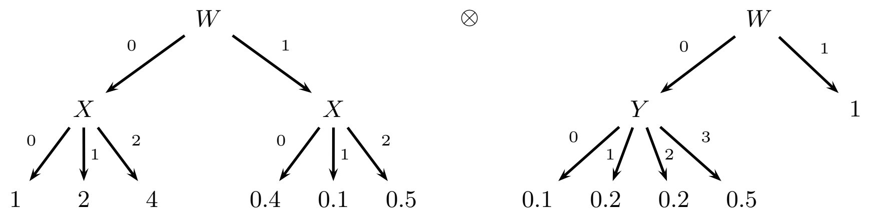

Figure 2: Decomposition of the tree in figure 1 with respect to variable X.

contain $\mathcal{T}_1$ and $\mathcal{T}_2$ can be decomposed by factorisation. Formally, approximate factorisation is defined through the concept of δ-factorisability.

Definition 1. A probability tree $\mathcal{T}$ is *δ-factorisable* within context $(\mathbf{X}_C = \mathbf{x}_C)$, with proportionality factors $\boldsymbol{\alpha}$ with respect to a divergence measure $\mathcal{D}$ if there is an $x_i \in \Omega_X$ and a set $L \subset \Omega_X \setminus \{x_i\}$ such that for every $x_j \in L$, $\exists \alpha_j > 0$ such that

$$\mathcal{D}(\mathcal{T}^{R(\mathbf{X}_C=\mathbf{x}_C, X=x_j)}, \alpha_j \cdot \mathcal{T}^{R(\mathbf{X}_C=\mathbf{x}_C, X=x_i)}) \leq \delta \ ,$$

where $\mathcal{T}^{R(\mathbf{X}_C=\mathbf{x}_C, X=x_i)}$ denotes the subtree of $\mathcal{T}$ which is reached by the branch where variables $\mathbf{X}_C$ take values $\mathbf{x}_C$ and X_i takes value x_i. Parameter $\delta > 0$ is called the *tolerance of the approximation.*

Observe that if $\delta = 0$, we have exact factorisation. In the definition above, $\mathcal{D}$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ are related in such a way that $\boldsymbol{\alpha}$ can be computed in order to minimise the value of the divergence measure $\mathcal{D}$.

In (Martínez et al., 2005) several divergence measures are considered, giving the optimal $\boldsymbol{\alpha}$ in each case. In this work we will consider only the measure that showed the best performance in (Martínez et al., 2005) that we will describe now. Consider a probability tree $\mathcal{T}$. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be sub-trees of $\mathcal{T}$ below a variable X, for a given context $(\mathbf{X}_C = \mathbf{x}_c)$ with leaves $\mathcal{P} = \{p_i : i = 1, \ldots, n \ ; \ p_i \neq 0\}$ and $\mathcal{Q} = \{q_i : i = 1, \ldots, n\}$ respectively. As we described before, approximate factorisation is achieved by replacing $\mathcal{T}_2$ by another tree $\mathcal{T}_2^*$ such that $\mathcal{T}_2^*$ is proportional to $\mathcal{T}_1$. It means that the leaves of $\mathcal{T}_2^*$ will be $\mathcal{Q}^* = \{\alpha p_i : i = 1, \ldots, n\}$, where α is the proportionality factor between $\mathcal{T}_1$ and $\mathcal{T}_2$. Let us denote by $\{\pi_i = q_i/p_i, i = 1, \ldots, n\}$ the ratios between the leaves of $\mathcal{T}_2$ and $\mathcal{T}_1$.

The χ^2 divergence, defined as

$$\mathcal{D}_\chi(\mathcal{T}_2, \mathcal{T}_2^*) = \sum_{i=1}^{n} \frac{(q_i - \alpha p_i)^2}{q_i} \ ,$$

is minimised for α equal to

$$\alpha_{\mathcal{X}} = \frac{\sum_{i=1}^{n} p_i}{\sum_{i=1}^{n} p_i/\pi_i} .$$

In this work we will use its normalised version

$$\mathcal{D}_{\mathcal{X}^*}(\mathcal{T}_2, \mathcal{T}_2^*) = \sqrt{\frac{\mathcal{D}_{\mathcal{X}}}{\mathcal{D}_{\mathcal{X}} + n}} , \qquad (2)$$

which takes values between 0 and 1 and is minimised for the same α.

4 Dynamic importance sampling combined with factorisation

As we mentioned in section 2, the complexity of the dynamic importance sampling algorithm relies on the computation of the sampling distribution. In the algorithm proposed in (Moral and Salmerón, 2005), this complexity is controlled by pruning the trees resulting from multiplications of other trees.

Here we propose to use approximate factorisation instead of tree pruning to control the complexity of the sampling distributions computation. Note that these two alternatives (approximate factorisation and tree pruning) are not exclusive. They can be used in a combined form. However, in order to have a clear idea of how approximate factorisation affects the accuracy of the results, we will not mix it with tree pruning in the algorithm proposed here.

The difference between the dynamic algorithm described in (Moral and Salmerón, 2005) and the method we propose in this paper is in the computation of the sampling distributions. The simulation phase and the update of the sampling distribution is carried out exactly in the same way, except that we have established a limit for the number of updates during the simulations equal to 1000 iterations. It means that, after iteration #1000 in the simulation, the sampling distributions are no longer updated. We have decided this with the aim of avoiding a high increase in space requirements during the simulation. Therefore, we only describe the part of the algorithm devoted to obtain the sampling distribution for a given variable X_i.

```
Get Sampling Distribution($X_i$,$H$,$\mathcal{D}$,$\delta$)

1. $H_i := \{f \in H | f$ is defined for $X_i\}$.
2. FOR each $f \in H_i$,
   (a) IF there is a context $\mathbf{X}_C$ for which
       the tree corresponding to $f$ is $\delta$-
       factorisable with respect to $\mathcal{D}$,
       • Decompose $f$ as $f_1 \times f_2$, where $f_1$
         is defined for $X_i$ and $f_2$ is not.
       • $H_i := (H_i \setminus \{f\}) \cup \{f_1\}$.
       • $H := (H \setminus \{f\}) \cup \{f_2\}$.
   (b) ELSE
       • $H := H \setminus \{f\}$.
3. $f_i := \prod_{f \in H_i} f$.
4. $f_i' := \sum_{x \in \Omega_{X_i}} f_i$.
5. $H := H \cup \{f_i'\}$.
6. RETURN $(f_i)$.
```

According to the algorithm above, the sampling distribution for a variable X_i is computed by taking all the functions which are defined for it, but unlike the method in (Moral and Salmerón, 2005), the product of all those functions is postponed until the possible factorisations of all the intervening functions are tested. Therefore, the product in step 3 is carried out over functions with smaller domains than the product in step 2 of the algorithm described in section 2.

The accuracy of the sampling distribution is controlled by parameter δ, so that higher values of it result in worse approximations.

5 Experimental evaluation

In this section we describe a set of experiments carried out to show how approximate factorisation can be used to control the level of approximation in dynamic importance sampling. We have implemented the algorithm in java, as a part of the Elvira system (Elvira Consortium, 2002). We have selected three real-world Bayesian networks borrowed from the Machine Intelligence group at Aalborg University (`www.cs.aau.dk/research/MI/`). The three networks are called Link (Jensen et al., 1995), Munin1 and Munin2 (Andreassen et al., 1989). Table 1 displays, for each network, the number of variables, number of observed variables (selected at random) and its *size*. By the size we mean the sum of the clique sizes that resulted when using HUGIN (Jensen et al., 1990) for computing the exact posterior distributions given the observed variables.

Table 1: Statistics about the networks used in the experiments.

	Vars.	Obs. vars.	Size
Link	441	166	23,983,962
Munin1	189	8	83,735,758
Munin2	1003	15	2,049,942

We have run the importance sampling algorithm with tolerance values $\delta = 0.1, 0.05, 0.01, 0.005$ and 0.001 and with different number of simulation iterations: 2000, 3000, 4000 and 5000. Given the random nature of this algorithm, we have run each experiment 20 times and the errors have been averaged. For one variable X_l, the error in the estimation in its posterior distribution is measured as (Fertig and Mann, 1980):

$$G(X_l) = \sqrt{\frac{1}{|\Omega_{X_l}|} \sum_{a_l \in \Omega_{X_l}} \frac{(p'(a_l|\mathbf{e}) - p(a_l|\mathbf{e}))^2}{p(a_l|\mathbf{e})(1 - p(a_l|\mathbf{e}))}} \tag{3}$$

where $p(a_l|\mathbf{e})$ is the true *a posteriori* probability, $p'(a_l|\mathbf{e})$ is the estimated value and $|\Omega_{X_l}|$ is the number of cases of variable X_l. For a set of variables $\{X_1, \ldots, X_n\}$, the error is:

$$G(\{X_1, \ldots, X_n\}) = \sqrt{\sum_{i=1}^{n} G(X_i)^2} \tag{4}$$

This error measure emphasises the differences in small probability values, which means that is more discriminant than other measures like the mean squared error.

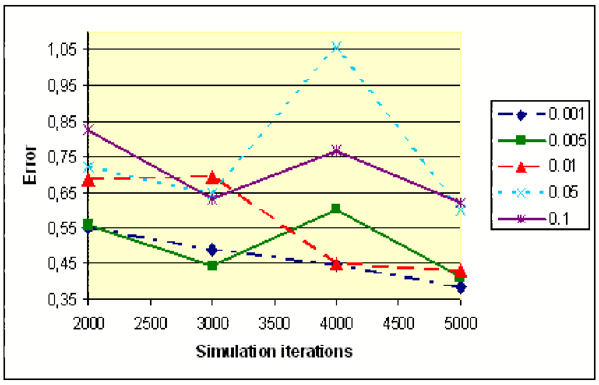

Figure 3: Results for Link with $\delta = 0.1, 0.05, 0.01, 0.005$ and 0.001.

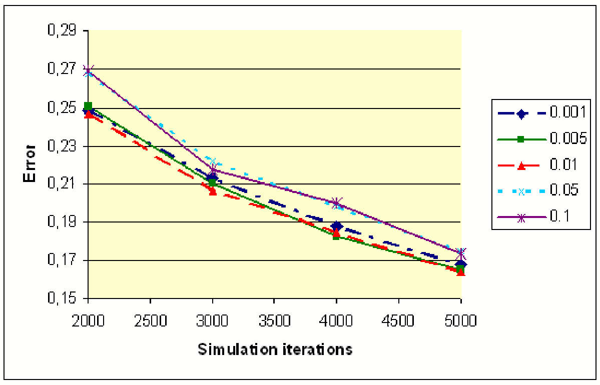

Figure 4: Results for Munin1 with $\delta = 0.1, 0.05, 0.01, 0.005$ and 0.001.

5.1 Results discussion

The results summarised in figures 3, 4 and 5, indicate that the error can be somehow controlled by means of the δ parameter. There are, however, irregularities in the graphs, probably due to the variance associated with the estimations, due to the stochastic nature of the importance sampling method.

The best estimations are achieved for the Munin1 network. This is due to the fact that few factorisations are actually carried out, and therefore the number of approximations is lower, what also decreases the variance of the estimations.

Factorisation or even approximate factorisation are difficult to carry out over very extreme distributions. In the approximate case, a high value for δ would be necessary. This difficulty

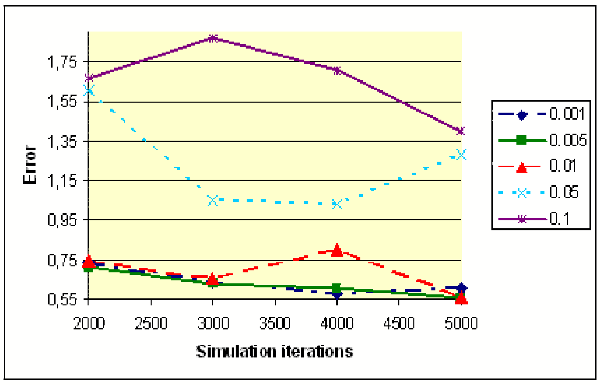

Figure 5: Results for Munin2 with $\delta = 0.1, 0.05, 0.01, 0.005$ and 0.001.

also arises when using tree pruning as approximation method, since this former method is good to capture uniform regions.

Regarding the high variance associated with the estimations suggested by the graphs in figures 3 and 5, it can be caused by multiple factorisations of the same tree: A tree can be factorised for some variable and afterwards, one of the resulting factors, can be factorised again when deleting another variable. In this case, even if the local error is controlled by δ, there is a global error due to the concatenations of factorisations that is difficult to handle. Perhaps the solution is to forbid that trees that result from a factorisations be factorised again.

6 Conclusions

In this paper we have introduced a new version of the dynamic importance sampling algorithm proposed in (Moral and Salmerón, 2005). The novelty consists of using the approximate factorisation of probability trees to control the approximation of the sampling distributions and, in consequence, the final approximations.

The experimental results suggest that the method is valid but perhaps more difficult to control than tree pruning.

Even though the goal of this paper was to analyse the performance of tree factorisation as a stand-alone approximation method, one immediate conclusion that can be drawn from the work presented here is that the joint use of approximate factorisation and tree pruning should be studied. We have carried out some experiment in this direction and the first results are very promising: the variance in the estimations and the computing time seems to be significantly reduced. The main problem of combining both methods is that the difficulty of controlling the final error of the estimation through the parameters of the algorithm increases. We leave for an extended version of this paper a detailed insight on this issue. The experiments we are currently carrying out suggest that the most sensible way of combining both approximation procedures is to use slight tree pruning and also low values for the tolerance of the factorisation. Otherwise, the error introduced by one of the approximation method can affect the other.

It seems difficult to establish a general rule for determining which approximation method is preferable. Depending on the network and the propagation algorithm, the performance of both techniques may vary. For instance, approximate factorisation is appropriate for algorithms that handle factorised potentials, as Lazy propagation or the dynamic importance sampling method used in this paper.

Finally, an efficiency comparison between the resulting algorithm and the dynamic importance sampling algorithm in (Moral and Salmerón, 2005) will be one of the issues that we plan to aim at.

Acknowledgments

This work has been supported by the Spanish Ministry of Education and Science through project TIN2004-06204-C03-01.

References

S. Andreassen, F.V. Jensen, S.K. Andersen, B. Falck, U. Kjærulff, M. Woldbye, A.R. Sorensen, A. Rosenfalck, and F. Jensen, 1989. *Computer-Aided Electromyography and Expert Systems*, chapter MUNIN - an expert EMG assistant, pages 255–277. Elsevier Science Publishers B.V., Amsterdam.

A. Cano, S. Moral, and A. Salmerón. 2000. Penniless propagation in join trees. *International Journal of Intelligent Systems*, 15:1027–1059.

A. Cano, S. Moral, and A. Salmerón. 2002. Lazy evaluation in Penniless propagation over join trees. *Networks*, 39:175–185.

A. Cano, S. Moral, and A. Salmerón. 2003. Novel strategies to approximate probability trees in Penniless propagation. *International Journal of Intelligent Systems*, 18:193–203.

J. Cheng and M.J. Druzdzel. 2000. AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188.

P. Dagum and M. Luby. 1993. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153.

Elvira Consortium. 2002. Elvira: An environment for creating and using probabilistic graphical models. In J.A. Gmez and A. Salmern, editors, *Proceedings of the First European Workshop on Probabilistic Graphical Models*, pages 222–230.

K.W. Fertig and N.R. Mann. 1980. An accurate approximation to the sampling distribution of the studentized extreme-valued statistic. *Technometrics*, 22:83–90.

L.D. Hernández, S. Moral, and A. Salmerón. 1998. A Monte Carlo algorithm for probabilistic propagation in belief networks based on importance sampling and stratified simulation techniques. *International Journal of Approximate Reasoning*, 18:53–91.

F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. 1990. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4:269–282.

C.S. Jensen, A. Kong, and U. Kjærulff. 1995. Blocking Gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies*, 42:647–666.

U. Kjærulff. 1994. Reduction of computational complexity in Bayesian networks through removal of weak dependencies. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 374–382. Morgan Kaufmann, San Francisco.

I. Martínez, S. Moral, C. Rodríguez, and A. Salmerón. 2002. Factorisation of probability trees and its application to inference in Bayesian networks. In J.A. Gámez and A. Salmerón, editors, *Proceedings of the First European Workshop on Probabilistic Graphical Models*, pages 127–134.

I. Martínez, S. Moral, C. Rodríguez, and A. Salmerón. 2005. Approximate factorisation of probability trees. *ECSQARU'05. Lecture Notes in Artificial Intelligence*, 3571:51–62.

S. Moral and A. Salmerón. 2005. Dynamic importance sampling in Bayesian networks based on probability trees. *International Journal of Approximate Reasoning*, 38(3):245–261.

J. Pearl. 1987. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32:247–257.

R.Y. Rubinstein. 1981. *Simulation and the Monte Carlo Method.* Wiley (New York).

A. Salmerón, A. Cano, and S. Moral. 2000. Importance sampling in Bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413.

R.D. Shachter and M.A. Peot. 1990. Simulation approaches to general probabilistic inference on belief networks. In M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, volume 5, pages 221–231. North Holland (Amsterdam).

N.L. Zhang and D. Poole. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328.