Probabilistic Inference using Linear Gaussian Importance Sampling for Hybrid Bayesian Networks

Wei Sun and K.C. Chang
Dept. of Systems Engineering & Operations Research
School of Information Technology and Engineering
George Mason University
Fairfax, VA 22030, USA

ABSTRACT

Probabilistic inference for Bayesian networks is in general NP-hard using either exact algorithms or approximate methods. However, for very complex networks, only the approximate methods such as stochastic sampling could be used to provide a solution given any time constraint. There are several simulation methods currently available. They include logic sampling - the first proposed stochastic method for Bayesian networks, the likelihood weighting algorithm - the most commonly used simulation method because of its simplicity and efficiency, the Markov blanket scoring method, and the importance sampling algorithm. In this paper, we first briefly review and compare these available simulation methods, then we propose an improved importance sampling algorithm called linear Gaussian importance sampling algorithm for general hybrid model (LGIS). LGIS is aimed for hybrid Bayesian networks consisting of both discrete and continuous random variables with arbitrary distributions. It uses linear function and Gaussian additive noise to approximate the true conditional probability distribution for continuous variable given both its parents and evidence in a Bayesian network. One of the most important features of the newly developed method is that it can adaptively learn the optimal important function from the previous samples. We test the inference performance of LGIS using a 16-node linear Gaussian model and a 6-node general hybrid model. The performance comparison with other well-known methods such as Junction tree (JT) and likelihood weighting (LW) shows that LGIS-GHM is very promising.

Keywords: importance sampling, Bayesian networks, hybrid model, stochastic simulation

1. INTRODUCTION

Bayesian Network is a graphical probabilistic model to represent the dependence relationships between all random variables involved in intelligent systems. It has been applied in increasingly many areas such as target identification, data fusion, medical diagnosis, financial analysis, etc. In general, Bayesian network model consists of a set of nodes and arcs between the dependent nodes. With the structure and the prior probability distributions for all of the variables specified, the network contains the information needed to answer all probabilistic inference queries about these variables with or without evidence. Over the past two decades, many inference algorithms have been proposed to provide either exact or approximate solutions. Different inference algorithms have different tradeoffs in computational efficiency, accuracy, and applicable network topologies. For example, Pearl’s message propagation algorithm [1][14] is an exact inference algorithm only works on the single-connected Bayesnet with polynomial complexity in the number of nodes. In general, either exact or approximate inference algorithms are all stumbled on the theoretical NP-hardness with complex networks such as large, densely connected or networks containing mixed (discrete and continuous) variables. Unfortunately, these types of hybrid networks with non-Gaussian distributions or non-linear relationships are commonly encountered in practice. In such cases, the approximate method is the only available choice to give a solution with any time constraint.

There are mainly two classes of approximate inference. One is model simplification such as simplifying the network structure to make it singly connected, or discretizing the continuous variables, linearization of nonlinear relationship,
reversing arcs of evidence nodes, etc. Stochastic sampling methods, also called Monte Carlo simulation, is another class of approximate inference. They are the most popular and well-known approximate Bayesnet inference algorithms. Basically, they generate a set of random samples or instantiations of the network according to some pre-selected distribution, and then approximate the posterior distribution of the query nodes by the frequencies of appearances in the samples. The accuracy of resulting probability estimate depends on the sample size and the pre-selected distributions used for sampling, irrespective to the structure of the networks. Stochastic sampling methods can be further divided into two main categories: importance sampling algorithms and Markov Chain Monte Carlo (MCMC) methods. In this paper, we will focus on the importance sampling algorithms.

Logic sampling (LS) is the first and simplest forward sampling algorithm [3]. It repeatedly generates samples using the prior distributions of the Bayesian network, if the instantiation of evidence nodes are not consistent with the observation, then the sample is discarded. After enough trials, posterior distributions of query nodes are estimated by counting the frequencies with which relevant states occur in the samples. The biggest problem with logic sampling algorithm is that it wastes many inconsistent samples, especially when the observations are unlikely. Also the fraction of useful samples decreased exponentially with the number of evidence nodes. To overcome this problem, likelihood weighting (LW) method [4][12] considers every sample and weights each sample with likelihood of evidence conditioned on the sample. While LW performs significantly better than LS, it may still take a long time to converge with unlikely evidence in large networks. In the worst case, our experiments show that LW is not able to converge given extremely unlikely evidence even with huge amount of samples.

Both logic sampling and likelihood weighting algorithms use the prior distributions of the networks to generate samples. It is well known that the performance of sampling methods depends not only on the sample size, but more so on the sampling distributions. Instead of using the prior, another idea originated from the finite-dimensional integral, is to use the “importance” function for sampling. The importance distributions could be selected in many ways, but principally, the closer the importance distribution to the true posterior distribution, the more efficient and accurate the algorithm will be. Shachter and Peot introduced two variants of importance sampling algorithm: self-importance sampling (SIS) and heuristic importance sampling (HIS) [4]. SIS updates its importance function occasionally using the scores generated in the samples. HIS first simplifies the network as a polytree, then use a modified version of Pearl’s polytree algorithm [1] to compute its importance function and never update it during sampling. Performance of SIS and HIS are not stable as the chosen importance functions could lead to bad approximation of the actual posterior distributions [13].

Recently, a very efficient importance sampling algorithm called the adaptive importance sampling for Bayesian networks (AIS-BN) [13] by Jian Cheng and M. J. Druzdzel was proposed. AIS-BN updates the importance function iteratively by learning from the samples generated by the previous importance function. It takes advantages of the structure of Bayesian network by decomposing the importance function into a product of conditional probability distributions (CPD). Instead of learning the optimal importance distribution of the entire network, it learns the optimal importance function for each unobserved node given evidence and its parents in every iteration. AIS-BN also introduces different weights for samples generated at different learning stage. Experimental results show AIS-BN performs very well in large networks, even with extremely unlikely evidence. However, AIS-BN only works for purely discrete networks where all variables involved are discrete. Unfortunately, most practical models involve hybrid modes where both discrete and continuous variables are needed. In this paper, we extend the basic ideas of AIS-BN to hybrid models by learning importance conditional probability distribution (ICPD) for continuous nodes. The formal definition of ICPD will be presented in the next section.

Most of the stochastic sampling algorithms can be considered as a special case of importance sampling. The main differences between them are how they choose and update the importance functions and how they generate and weight the samples. In this paper, we develop an approach to systematically and adaptively approximate the optimal importance function by linear function and additive Gaussian noise for continuous variables. This iterative procedure, called linear Gaussian importance sampling algorithm (LGIS), learn the importance function from the previously generated samples, then recursively generate new samples according to the latest learned importance function. In the next section, we will introduce LGIS algorithm in details. Section 3 describes our experimental models and the simulation results. A summary section with additional discussion will be presented in Section 4.
2. LINEAR GAUSSIAN IMPORTANCE SAMPLING FOR GENERAL HYBRID MODEL

The purpose of Bayesian inference is to find the best estimate of the posterior distributions for the query variables given evidence. Let us denote the set of all the unobserved nodes in the network as $X$, the set of all evidence nodes as $E$, and the node of our interest as $T$. The corresponding lower case letters $x$, $e$, $t$, stand for their specific instantiations respectively. Without loss of generality, we denote the posterior probability distribution to be estimated as $Pr(T|E=e)$. If $T$ is a discrete random variable, then $Pr(T|E=e)$ is a probability mass function; if $T$ is a continuous random variable, then $Pr(T|E=e)$ is a probability distribution function. Using importance sampling (for more details of the mathematic foundation of importance sampling, see [15]), we could compute the probabilities (for discrete nodes) or likelihoods (for continuous nodes) in Bayesian networks. For example, in order to compute $Pr(E=e)$, we need to integral $Pr(X, E=e)$ over all $X$:

$$Pr(E = e) = \int_X Pr(X, E = e)$$

Assuming an importance function $f(x)$ was used to generate samples of $X$, with $n$ independently sample points $x_1, x_2, \ldots, x_n$, it can be easily shown that the average value of $Pr(X, E=e)$ over $f(x)$ is an estimate of $Pr(E=e)$:

$$Pr(E = e) \approx \frac{1}{n} \sum_{i=1}^{n} \frac{Pr(X = x_i, E = e)}{f(x_i)}$$

(1)

The posterior distribution $Pr(T|E)$ could then be estimated by

$$Pr(T \mid E) = \frac{Pr(T, E)}{Pr(E)}$$

Theoretically, we should use different optimal importance functions for computing $Pr(T,E)$ and $Pr(E)$ to increase the accuracy of the estimate of the posterior distribution. However in practice, we usually use the same importance function for computing $Pr(E=e)$ and $Pr(T,E=e)$. When the sample size is large enough and the chosen importance function is close enough to the optimal importance function, all biases in estimation will decrease to lead to a reasonable estimate.

It is easily proven that the optimal importance function for computing $Pr(E=e)$ using importance sampling is the true posterior distribution denoted by $\rho(X) = Pr(X \mid E = e)$. However, it is often difficult to obtain this function directly and exactly due to its potential high dimensionality. Nonetheless, the joint distribution can be decomposed into a product of conditional probabilities in Bayesian networks. We can use the following function to approximate the true posterior distribution:

$$\rho(X) = \hat{\rho}(X) = \prod_{i=1}^{n} Pr(X_i \mid Pa(X_i), E = e)$$

(2)

where $X_1, X_2, \ldots X_n$ are the $n$ nodes in the network except evidence nodes, $Pa(X_i)$ are the parent nodes of node $X_i$. This function integrates the effect of all the evidence on every node except evidence nodes in the network during sampling process. If the structure of the network remains the same after absorbing the evidence, this function is the optimal importance function. However, if the posterior structure of the model changes drastically, this importance function may perform poorly.

In equation (2), $Pr(X_i \mid Pa(X_i), E = e)$ is the conditional probability distribution of $X_i$ given all of its parent nodes and evidence. We define this distribution the importance conditional probability distribution (ICPD) of $X_i$. The key question is how to find ICPD for every unobserved node. The idea is to estimate the ICPD for each node from the current samples, and then generate new samples according to the learned ICPD for the next iteration. For discrete variables, estimating ICPDs is straightforward as they can be represented in a table form [13]. But for continuous variables, because of their infinite possible states and their arbitrary functional relationships with their parents, it is much harder to develop a systematic procedure for learning ICPD. In this paper, we propose to approximate the relationships between continuous nodes and their parents as a linear function with an additive Gaussian noise. Specifically, we use.
the least square method to learn the linear function to fit the sample data, and then model the difference between linear function approximation and the raw sample data as a Gaussian mixture [9]. It is understood that higher order approximation may yield better performance than linear approximation, but with more parameters to be learned and higher computational costs. Our experimental results show that the linear approximation with an additive Gaussian mixture noise performs fairly well.

To illustrate the concepts, let us look at a simple Bayesian network containing a continuous node $Y$ and its parents $P_1, \ldots, P_n$, as given in Figure 1,

![Figure 1. A Simple BN Example](image)

Without evidence, we know the conditional distribution of $Y$ given its parents is the prior CPD. However, given evidence $E$, the ICPD of $Y$ is an unknown arbitrary function of its parents $P_1, \ldots, P_n$, denoted as $Y = f(P_1, P_2, \ldots, P_n)$. LGIS approximates this function by a linear function with an additive Gaussian mixture $G$.

\[
ICPD (Y) = Pr(X_1 | Pa(X_i), E = e) = f (P_1, P_2, \ldots, P_n) = a_0 + \sum_{i=1}^{n} a_i P_i + G \tag{3}
\]

In equation (3), $a_0$, $a_1$, $\ldots$, $a_n$ are the linear coefficients which could be easily learned by least square method from samples and $G$ is the Gaussian noise term to model the difference between the linear approximation and the true unknown ICPD. Started from any importance function, such as the prior, each sample generated will be assigned a score based on the degree of consistence to the given evidence, which could be computed using equation (1). All samples with associated scores represent the joint posterior distribution. They will be used to learn the ICPDs for all of the unobserved nodes. The importance function will then be updated as the product of all the ICPDs.

The basic procedure of LGIS is presented as the following:

1. Initialize importance function $Pr^0(X)$ with some heuristic methods, or just using the prior, set the desired number of samples $n$ at each learning step and the number of the total iterations $k$;

2. for $i = 1: k$
   
   for $j = 1: n$
   
   generate a sample according to $Pr^{k-1}(X)$
   
   calculate the score for the current sample
   
   end
   
   Resampling based on score
   
   from the re-sampled samples, learn the ICPD for each node to approximate $Pr^k(X) = \prod_{j=1}^{n} Pr(X_j | Pa(X_j), E = e)$,

   update the importance function as $Pr^k(X) = a \times Pr^{k-1}(X) + (1-a) \times Pr^r(X)$, where “$a$” is the learning rate.
3. After the learning stage, final samples are generated by the latest importance function to be used for estimating the posterior distribution of the query nodes of our interest.

There are some important issues and adjustable parameters in LGIS. First of all, initialization of importance function is very critical to determine if it will converge or not. We usually use the prior as the initialization with some constraints on effective sample size before we proceed. This is because we need at least a minimum number of effective samples in order to start learning. Number of total learning iterations, sample size at different stage and learning rate are highly model dependent and they could affect inference performance.

### 3. EXPERIMENTAL MODEL AND PERFORMANCE ANALYSIS

LGIS algorithm is aimed for the hybrid nonlinear non-Gaussian model. It is well known that no exact inference exist for such type of Bayesian networks. For testing purposes, we designed two experimental models. One is a 16-node hybrid linear Gaussian model called LG-16 consisting of two discrete nodes and other linear Gaussian continuous variables. For LG-16, Junction tree can give us an exact inference solution. We use that as the gold standard for performance comparison. Another model is a general hybrid 8-node Bayesian network called GHM-8 with highly non-linear conditional relationships. GHM-8 has one discrete node with 8 states, one uniform continuous node distributed between 0 and $\pi$, four continuous nodes with Gaussian conditional distributions, and two evidence nodes with Gaussian conditional distributions. The conditional relationships for the continuous variables in GHM-8 are highly nonlinear. Model structures and parameters of LG-16 and GHM-8 are shown in Figures 2 and 3 respectively.

Because of its nonlinearity and non-Gaussianity, there is no exact inference solution for GHM-8. In this case, LW algorithm is used with a large number of samples to produce an approximate a posteriori distribution. The results are used as the benchmark for comparison. Without loss of generality, we choose the discrete node TD in LG-16 and the discrete node T in GHM-8 as the target nodes of interest respectively. Given randomly chosen evidence sets, we compare the inference performance between LGIS and LW. A simple performance metric based on root mean square error (RMSE) is used to evaluate the algorithm, i.e.,

$$RMSE = \sqrt{\frac{1}{n_s} \sum_{i=1}^{n_s} (P'(x_i) - P(x_i))^2}$$

where $n_s$ is the number of states of the compared node, $P'(x_i)$ is the exact marginal probabilities of state $i$ and $P(x_i)$ is the corresponding estimated probabilities. RMSE has been commonly used in performance comparison because of its simplicity and ability to represent the estimation error. In addition, convergence rate which can be used to judge the quality of sampling algorithm can also be observed in terms of RMSE.

In all experiments, we initialize the importance function using the prior distributions. Notice that for discrete variables, impoverishment might occur during sampling resulted from bad importance function. It means only one or two of the discrete states dominate the distribution while the remaining states have negligible probabilities. In that case, the true state may be missed and unrecoverable if it is not one of the dominate state. To avoid this problem, we define a probability threshold for the state of the discrete variable. If the probability of a state is less than the threshold, we artificially replace it with the threshold and subtract the corresponding probability from other states accordingly. This artificial step is needed only in the first few learning steps where the impoverishment problem is more likely to happen.
3.1 Parameters Selection

We have several adjustable parameters for LGIS. To avoid oscillation in importance function learning, one could use an adaptive learning rate strategy where the rate is reduced iteratively. However, we use a fixed learning rate 0.7 for LGIS in all of our experiments. In initialization, we generated more samples by setting a constraint for the minimum number of effective samples. This is because the initial importance function is usually poor. Empirical results show that 8,000 initial samples with a minimum of 3 effective samples are acceptable for LGIS with both experiment models. We generated 2,000 samples in each subsequent learning step and the number of total learning steps is 10 in all of the experiments. After learning step, we finally generated 10,000 samples by the latest importance function and use them to estimate the posterior distribution.

3.2 Simulation Results for Model LG-16

We generated a total of 140 test cases on LG-16. In cases 1 to 100, evidence sets are generated by forward sampling. They are further divided into two groups consisting of 40 evidence sets with likelihood greater than 1e-5 and 60 evidence sets with likelihood between 1e-5 to 1e-10. The last 40 test cases are artificially generated to create extremely low likely evidence (between 1e-10 to 1e-30). We compared the performance by LGIS and LW in all test cases using RMSE based on the exact solution provided by Junction tree algorithm. To make the comparison fair, we set the same inference time for LGIS and LW. Because LGIS takes some time to learn and update the importance function, LW is able to generate more samples than LGIS with the same time constraint. Table 1 shows the performance results.
With higher likely evidence, LGIS performs similarly to LW. But with unlikely evidence, LGIS provided significantly better performance. In fact, with extremely low likely evidence, LW can not converge in most cases using the prior as the importance function.

<table>
<thead>
<tr>
<th></th>
<th>40 cases evidence likelihood</th>
<th>60 cases evidence likelihood</th>
<th>40 cases evidence likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt; 1e-5</td>
<td>1e-5 – 1e-10</td>
<td>1e-10 – 1e-30</td>
</tr>
<tr>
<td>LGIS</td>
<td>0.003635</td>
<td>0.0027678</td>
<td>0.0051364</td>
</tr>
<tr>
<td>LW</td>
<td>0.0041231</td>
<td>0.013931</td>
<td>0.12834</td>
</tr>
</tbody>
</table>

Table 1: average RMSE comparison

3.3 Simulation Results for Model GHM-8

Experiment with GHM-8 is to show the robustness of LGIS. 100 randomly generated evidence sets by forward sampling are chosen for testing. Average RMSE are computed for LGIS and LW over those evidence sets. Simulation results are listed as the below:

- LGIS (~20k total): 0.0016255
- LW 50k: 0.0060784
- LW 500k: 0.0021141
- LW 1m: 0.0015145
- LW 5m: 0.00063227

From the results, we can see that LW needs roughly 50 times more samples than LGIS to achieve a similar performance.

4. SUMMARY

For complex or hybrid networks, the approximate methods such as stochastic sampling are the only alternative to provide a solution for probabilistic inference. In this paper we propose an improved importance sampling algorithm called LGIS using linear function to approximate the true conditional function for continuous nodes in hybrid Bayesian networks. The key feature of the method is that it can update the important function by learning from the previous samples for continuous random variables. We use the linear functions and Gaussian mixture model to approximate the importance conditional distributions (ICPD) for the continuous variables in the network. We test the performance of LGIS using two hybrid models and compare that with the well known LW method. Experimental results show that LGIS performs significantly better than LW both in computational requirement and estimation accuracy. We believe this is a good initial step and one potential future research direction is to use higher order approximation for the relationships between continuous variables. Other areas that deserve some attentions are the dynamic learning rate and sample weight assignment at the each learning stage. Initialization for importance function is critical in determining the convergence rate and deserves a closer look. Backward simulation may be helpful in this critical step [16].

5. REFERENCES


