Improving Importance Sampling by Adaptive Split-Rejection Control in Bayesian Networks

Changhe Yuan[1] and Marek J. Druzdzel[2]

[1] Mississippi State University, Mississippi State, MS 39762, USA
cyuan@cse.msstate.edu
[2] School of Information Sciences and Intelligent Systems Program,
University of Pittsburgh, Pittsburgh, PA 15260, USA
marek@sis.pitt.edu

Abstract. Importance sampling-based algorithms are a popular alternative when Bayesian network models are too large or too complex for exact algorithms. However, importance sampling is sensitive to the quality of the importance function. A bad importance function often leads to much oscillation in the sample weights, and, hence, poor estimation of the posterior probability distribution. To address this problem, we propose the *adaptive split-rejection control* technique to adjust the samples with extremely large or extremely small weights, which contribute most to the variance of an importance sampling estimator. Our results show that when we adopt this technique in the EPIS-BN algorithm [14], adaptive split-rejection control helps to achieve significantly better results.

1 Introduction

Bayesian networks [11] have become core tools for knowledge representation in Artificial Intelligence. One obstacle to their application is their computational complexity, which is NP-hard [3]. Importance sampling based algorithms are a popular alternative when Bayesian network models are too large or too complex for exact algorithms. The state of the art importance sampling algorithm is the EPIS-BN algorithm [14], whose main idea is to use several steps of *loopy belief propagation* (LBP) [10] to estimate an importance function for importance sampling. The algorithm is shown to reach the precision limit of sampling algorithms on some networks. However, due to the potential instability of LBP and, hence, possibly poor importance functions, EPIS-BN can still perform sub-optimally. This is manifested in the oscillation in the sample weights. Samples with extremely large weights and extremely small weights contribute most to the variance of the estimator. To address this problem, we propose an *adaptive split-rejection control* technique to adjust these samples. The technique consists of two parts: *adaptive split control*, which is to adaptively split samples with extremely large weights, and *adaptive rejection control*, which is to stochastically reject samples with extremely small weights. Although the adaptive split-rejection control may introduce correlation into the samples, the correlation is minimal in comparison to the reduction in the overall variance for sample sets with much oscillation. Our results show that when we apply adaptive split-rejection control in EPIS-BN, it helps the algorithm to achieve significantly better results. Two closely related techniques are

Z. Kobti and D. Wu (Eds.): Canadian AI 2007, LNAI 4509, pp. 332–343, 2007.

resampling [9] and *pruned-enriched Rosenbluth method* (PERM) [5,7,12]. The major advantage of our technique is that it exploits the fact that importance sampling in Bayesian networks is a high dimensional problem such that it generates useful samples more efficiently.

2 Importance Sampling

We start with the theoretical roots of importance sampling. We will use capital letters for variables and lowercase letters for their states. Bold letters denote sets of variables or states. Let $f(\mathbf{X})$ be a function of n variables $\mathbf{X} = \{X_1, ..., X_n\}$ over the domain $\Omega \subset R^n$. Consider the problem of estimating the multiple integral

$$V = \int_{\Omega} f(\mathbf{x}) d\mathbf{x} \ . \tag{1}$$

We assume that the domain of integration of $f(\mathbf{X})$ is bounded, i.e., that V exists. Importance sampling approaches this problem by estimating

$$V = \int_{\Omega} \frac{f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x}) d\mathbf{x} \ , \tag{2}$$

where $g(\mathbf{X})$, called the *importance function*, is a probability density function such that $g(\mathbf{X}) > 0$ across the entire domain Ω. One practical requirement of $g(\mathbf{X})$ is that it should be easy to sample from. In order to estimate the integral, we generate samples $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ from $g(\mathbf{X})$ and use the generated values in the sample-mean formula

$$\hat{V} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\mathbf{x}_i)}{g(\mathbf{x}_i)} \ . \tag{3}$$

The estimator almost surely converges to V under certain weak assumptions [4].

The performance of the estimator in Equation 3 can be measured by its variance

$$var_g[\frac{f(\mathbf{X})}{g(\mathbf{X})}] = \int_{\Omega} \frac{f^2(\mathbf{x})}{g(\mathbf{x})} d\mathbf{x} - V^2. \tag{4}$$

Rubinstein [13] shows that if $f(\mathbf{X}) > 0$, the optimal importance function is

$$g(\mathbf{X}) = \frac{f(\mathbf{X})}{V} \ . \tag{5}$$

In this case, the variance of the estimator is zero. However, the concept of the optimal importance function is of rather theoretical significance because finding V is equivalent to finding the posterior distribution, which is the problem that we are facing. Nevertheless, it suggests that if we find instead a function that is close enough to the optimal importance function, we can still expect good convergence rates.

3 Improving Importance Sampling by Adaptive Split-Rejection Control

Since we normally have no access to the optimal importance functions, we have to resort to approximation techniques or even guess work to estimate them. Importance functions thus obtained may yield far from optimal performance, which is directly reflected in the much oscillation in the sample weights. In such a sample set, extremely large weights and extremely small weights contribute most to the variance of the estimator. Furthermore, if the ratio f/g is unbounded, the variance of the estimator may be infinite, in which case importance sampling is unreliable. To remedy this problem, we propose the *adaptive Split-Rejection control* technique to adjust these samples. The technique includes two parts: *adaptive split control* and adaptive rejection control.

3.1 Adaptive Split Control

Samples with extremely large weights have a large impact on the variance of an importance sampling estimator, because they dominate the estimator and make other samples less useful. If some of these large-weight samples are in the unimportant parts of f, the estimator will wrongly put too much emphasis on these parts, and its performance will inevitably be poor. To deal with this problem, we propose the *adaptive split control* technique. The basic idea is to split a sample with a large weight into several samples with smaller weights. This technique is related to the enrichment method in [12,5,7]. In Bayesian networks, since we need to go over all the nodes in the topological order to draw a single sample, we can further exploit the power of adaptive split control by appointing several rejection nodes, say every 50th nodes. More formally, adaptive split control works as follows.

Adaptive Split Control:

1. For $i = 1, ..., M$, draw $\mathbf{x}_i$ from $g(\mathbf{X})$, and keep track of the cumulative weights for every rejection node.
2. For all the rejection nodes, sort the weights ascendingly, and let split threshold $c_s = w_{\lfloor \alpha_s \times M \rfloor}$ using the choose *split percentile* α_s .
3. For $j = 1, ..., N$ ($N \gg M$), draw $\mathbf{x}_j$ from $g(\mathbf{X})$ according to Steps 4 and 5.
4. Go over each node in the topological order of the network, and instantiate it to a state that is sampled from its distribution conditional on its parents.
5. If the cumulative weight w becomes larger than the split threshold c_s at a rejection node, we split it into $k = \lfloor w/c_s + 1 \rfloor$ samples, each with weight w/k.

It is easy to show that the new estimator is still unbiased.

Theorem 1. *The importance sampling estimator with adaptive split control is unbiased.*

There are two purposes for the adaptive split control technique. First, we can prevent a sample weight from blowing up and get samples with more uniform weights. Second, after we split the sample, the resulting samples enable us to explore a wider range of the sample space, and we are more likely to get better samples. Intuitively, adaptive split control tries to modify the old importance function such that the new importance function $g_s(\mathbf{X})$ approximately follows the following distribution:

$$g_s(\mathbf{X}) = q_{c_s}^{-1} \max\{g(\mathbf{X}), \frac{f(\mathbf{X})}{c_s}\} , \tag{6}$$

where q_{c_s} is defined as

$$q_{c_s} = \int \max\{1, \frac{w(\mathbf{x})}{c_s}\} g(\mathbf{x}) d\mathbf{x} , \tag{7}$$

where $w(x) = f(x)/g(x)$. The result is only approximate because of the way that we calculate the number of split samples. Suppose the target density and the importance function are both normalized, the following theorem shows that adaptive split control reduces the χ^2 distance between the two distributions, which is defined as

$$\chi^2(f, g) = \int_{\Omega} \frac{[f(\mathbf{x}) - g(\mathbf{x})]^2}{g(\mathbf{x})} d\mathbf{x} \equiv var_g[\frac{f(\mathbf{X})}{g(\mathbf{X})}] . \tag{8}$$

Theorem 2. *The χ^2 distance between the modified importance function in Equation 6 and the target density is smaller than that between the original importance function and the target density; that is,*

$$var_{g_s}[\frac{f(\mathbf{X})}{g_s(\mathbf{X})}] \leq var_g[\frac{f(\mathbf{X})}{g(\mathbf{X})}] . \tag{9}$$

Proof. Let

$$\begin{aligned} h(w_1, w_2) = {} & [\max\{w_1, c_s\} - \max\{w_2, c_s\}] \times \\ & [w_1 \min\{w_1, c_s\} - w_2 \min\{w_2, c_s\}] . \end{aligned}$$

$h(w_1, w_2)$ is always nonnegative, because

$$h = \begin{cases} 0 \geq 0, & \max(w_1, w_2) < c_s, \\ c_s(w_1 - w_2)^2 \geq 0, & c_s < \min\{w_1, w_2\}, \\ (c_s - w_2)(w_1^2 - w_2 c_s) \geq 0, & w_1 \leq c_s \leq w_2, \\ (w_1 - c_s)(w_1 c_s - w_2^2) \geq 0, & w_2 \leq c_s \leq w_1. \end{cases}$$

Hence, $\max\{w(\mathbf{x}), c_s\}$ and $\min\{w(\mathbf{x}), c_s\} w(\mathbf{x})$ are positively correlated. Therefore,

$$\begin{aligned} & c_s[1 + var_{g_s}\{\frac{f(\mathbf{x})}{g_s(\mathbf{x})}\}] \\ & = q_{c_s} E_g[\min\{w(\mathbf{x}), c_s\} w(\mathbf{x})] \\ & = E_g[\min\{w(\mathbf{x}), c_s\} w(\mathbf{x})] E_g \max\{c_s, w(\mathbf{x})\} \\ & \leq E_g[\{\min\{w(\mathbf{x}), c_s\} \max\{c_s, w(\mathbf{x})\} w(\mathbf{x})\}] \\ & = c_s E_g[w^2(\mathbf{x})] \\ & = c_s[1 + var_g\{\frac{f(\mathbf{x})}{g(\mathbf{x})}\}] . \end{aligned}$$

Although splitting samples may introduce correlations among the samples, we can look at it in a different way. After we increase the mass of the importance function in the parts where large-weight samples come from, these weights will have smaller weights. In the mean time, we also have more chances to hit these samples. In some sense, we are performing selective resampling; we are only resampling from samples with extreme weights. Also, there is an interesting connection between adaptive split control and *Rao-Blackwellization* [8]. The idea of Rao-Blackwellization is, if we can perform part of the multiple integral in Equation 1 analytically, we can reduce the variance of the estimator in Equation 3. In adaptive split control, instead of performing exact integration, we draw multiple samples to estimate that part of the integral, which can also reduce the variance of the estimator.

3.2 Adaptive Rejection Control

Samples with extremely large weights have a large impact on the variance of an importance sampling estimator. Similarly, samples with extremely small weights also have impact on the variance. Since they are very small, they do not play much role in the estimator but only make its variance larger. Simply throwing them away is not good because that introduces bias. To adjust these samples, we can apply a technique called *rejection control* (RC) [8]. Suppose we have drawn samples $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ from $g(\mathbf{X})$. Let $w_j = f(\mathbf{X}_j)/g(\mathbf{X}_j)$. Rejection control conducts the following operation for any given threshold value $c_r > 0$.

Rejection Control:

1. For $j = 1, ..., N$, accept $\mathbf{x}_j$ with probability

$$r_j = \min\{1, \frac{w_j}{c_r}\} . \tag{10}$$

2. If the jth sample $\mathbf{x}_j$ is accepted, its weight is updated to $w_{*j} = q_{c_r} w_j / r_j$, where

$$q_{c_r} = \int \min\{1, \frac{w(\mathbf{x})}{c_r}\} g(\mathbf{x}) d\mathbf{x} . \tag{11}$$

It can also be easily shown that rejection control is an unbiased operation.

Theorem 3. *The importance sampling estimator with rejection control is unbiased.*

Similarly to adaptive split control, rejection control adjusts the importance function so that the new importance function $g_r(\mathbf{X})$ is expected to be closer to the target function $f(\mathbf{X})$. In fact, it is easily seen that

$$g_{c_r}(\mathbf{X}) = q_{c_r}^{-1} \min\{g(\mathbf{X}), \frac{f(\mathbf{X})}{c_r}\} . \tag{12}$$

The following theorem can be proven analogously to Theorem 2.

Theorem 4. *[8] The χ^2 distance between the target density and the modified importance function in Eqn. 12 is smaller than that between the target density and the original importance function; that is,*

$$var_{g_r}[\frac{f(\mathbf{X})}{g_r(\mathbf{X})}] \leq var_g[\frac{f(\mathbf{X})}{g(\mathbf{X})}] . \tag{13}$$

Theorem 4 is a very strong result, because it holds essentially for any c_r. However, it has also been shown that rejection control can be looked on as an importance sampling algorithm in a higher dimension, and the χ^2 distance between the target density and the modified importance function for the new algorithm is larger than that of the importance sampling algorithm without rejecting any samples [1].

To make rejection control useful, we propose the *adaptive rejection control* technique, which extends rejection control in two ways. First, to draw one sample for a Bayesian network, we need to go over all the nodes in the network. However, we need not wait until the end in order to figure out that some samples may have extremely low weights. If we find that they become very small and not promising before we finish drawing them, we can reject them early. Therefore, the first extension works as follows.

– *Early rejection:* Instead of using rejection control after we get a complete sample, we apply this technique for each, say 50th, node when drawing each sample.

This extension reduces the cost of rejecting samples and improves the efficiency in obtaining effective samples; It is the key element that makes rejection control useful.

Second, we notice that it is hard to specify a c_r in advance when we apply rejection control to importance sampling in Bayesian networks, because the posterior distribution is only known up to a constant. Therefore, the second extension is:

– *Adaptive rejection:* Instead of setting a rejection threshold c_r in advance, we draw an initial sample set with size M, sort the sample weights w_is, and let $c_r = w_{\lfloor \alpha \times M \rfloor}$, where α is a chosen *rejection percentile*.

The bigger α is, the more samples are likely to be rejected, so we need to draw more samples in order to obtain a predefined number of samples. The cost is that the running time of the algorithm will inevitably increase. Therefore, we can fine-tune the choice of α in order to achieve a satisfactory tradeoff between precision and efficiency.

To summarize, the adaptive rejection control technique works as follows:

Adaptive Rejection Control:

1. For $i = 1, ..., M$, draw $\mathbf{x}_i$ from $g(\mathbf{X})$, and keep track of the cumulative weights for every 50th node.
2. Sort the weights ascendingly, and let $c_r = w_{\lfloor \alpha_r \times M \rfloor}$ using the choose *rejection percentile* α_r for all the rejection nodes.
3. For $j = 1, ..., N$ ($N \gg M$), draw $\mathbf{x}_j$ from $g(\mathbf{X})$, during which we apply rejection control technique each time when we come across a rejection node.

3.3 Adaptive Split-Rejection Control

We call the resulting technique by putting adaptive split control and adaptive rejection control together *adaptive split-rejection control*. The following corollary follows from Theorem 1 and 3.

Corollary 1. *The importance sampling estimator with adaptive split-rejection control is unbiased.*

```
Algorithm: ASREPIS-BN
Input: Bayesian network B with variables $\mathbf{X}$, a set of evidence variables $\mathbf{E}$;
Output: The marginal distributions of non-evidence variables.

1. Order the nodes in their topological order.
2. Initialize parameters $\alpha_s$, $\alpha_r$, $M$, $N$, and $\epsilon$.
3. Use several steps of LBP to calculate an importance function.
4. Enhance the importance function by $\epsilon$-cutoff.
5. for $j \leftarrow 1$ to $M$ do
     for each $X_j$ in $\mathbf{X}$
       Sample $\mathbf{x}_j$ according to $P(X_j|\mathrm{P}A(X_j))$.
       Calculate the partial score $w_{jScore}$.
       if $j\%50 == 0$, store $w_{jScore}$.
     end for
   end for
6. Calculate thresholds $c_s$ and $c_r$ for all rejection nodes.
7. Set $i = 0$.
8. while $i < N$ do
     for each $X_j$ in $\mathbf{X}$
       Sample $\mathbf{x}_j$ according to $P(X_j|\mathrm{P}A(X_j))$.
       Calculate the partial score $w_{jScore}$.
       if $j\%50 == 0$ and $w_{jScore} < c_r^j$
         Accept $\mathbf{x}_j$ with $p = w_{jScore}/c_r^j$.
         if accepted,$w_{jScore} = c_r^j$; else,$w_{jScore} = 0$.
       else if $j\%50 == 0$ and $w_{jScore} > c_s^j$
         Split $\mathbf{x}_j$ into $k = w_{jScore}/c_s^j$ samples.
         Assign each sample weight $w_{jScore}/k$.
       end if
     end for
     if $w_{iScore} > 0$
       Add $w_{iScore}$ to the score tables.
       $i \leftarrow i + 1$;
     end if
   end while
9. Normalize each score table, and output the estimated beliefs for each node.
```

Fig. 1. The Adaptive Split-Rejection Controlled Evidence Pre-propagation Importance Sampling (ASREPIS-BN) Algorithm

Furthermore, it is easily seen that the new importance function of the adaptive split-rejection control has the following form.

$$g_{c_s r}(\mathbf{X}) = (q_{c_s} q_{c_r})^{-1} \min\{\max\{g(\mathbf{X}), \frac{f(\mathbf{X})}{c_s}\}, \frac{q_{c_s} f(\mathbf{X})}{c_r}\} . \tag{14}$$

The following corollary immediately follows from Theorems 2 and 4.

Corollary 2. *The χ^2 distance between the target density and the modified importance function in Eqn. 14 is smaller than that between the target density and the original importance function; that is,*

$$var_{g_{sr}}[\frac{f(\mathbf{X})}{g_{sr}(\mathbf{X})}] \leq var_g[\frac{f(\mathbf{X})}{g(\mathbf{X})}] \; . \tag{15}$$

This technique not only allows us to discard some poor samples well before we finish drawing them, but also enables us to explore a wider range of the sample space. The technique is very general, because it can be applied in any existing importance sampling algorithm for Bayesian networks.

However, we should not blindly apply adaptive split-rejection control the same way under all circumstances. When the original sample set is already good, the correlation introduced by the technique may become dominant in comparison to the reduction of the sample variance, so the results may become worse. In that case, it is desirable to adjust the split and rejection percentiles or simply switch it off altogether in order to reduce the number of samples involved in the control. The quality of a sample set can be evaluated by the *coefficient of variation* ($cv^2(w)$) [8]. The $cv^2(w)$ of the unnormalized weights is defined as follows:

$$cv^2(w) = \frac{\sum\limits_{j=1}^{N}(w(\mathbf{x}_j) - \overline{w})^2}{(N-1)\overline{w}^2} \; . \tag{16}$$

$cv^2(w)$ is a good estimator of the χ^2 distance between the importance function and the target distribution. We recommend to increase the split percentile and the decrease rejection percentile as $cv^2(w)$ decreases, and we switch off the adaptive split-rejection control as $cv^2(w)$ drops below some threshold value cv_c.

We extend the EPIS-BN algorithm using the above adaptive rejection control technique, which results in the*Adaptive Split-Rejection Controlled Evidence Pre-propagation Importance Sampling* (ASREPIS-BN) algorithm, outlined in Figure 1. The algorithm has three main stages. The first stage (Steps 1-4) applies loopy belief propagation and ϵ-cutoff to calculate an importance function. The second stage (Steps 5-6) draws an initial sample set to estimate the split thresholds c_s and rejection thresholds c_r. The third stage (Steps 7-9) applies adaptive split-rejection control to do importance sampling.

4 Experimental Results

To test the performance of the ASREPIS-BN algorithm, we compared it against the EPIS-BN algorithm. We did experiments on the ANDES, CPCS, and PATHFINDER networks. Our comparison was based on *Hellinger's distance* [6] between exact answers and sampling results. Hellinger's distance yields results identical toKullback-Leibler divergence in most cases, but its major advantage is that it can handle zero probabilities, which are not uncommon in Bayesian networks. We implemented our algorithm in C++ and performed our tests on a 2.8 GHz Xeon Windows XP computer with 2GB memory.

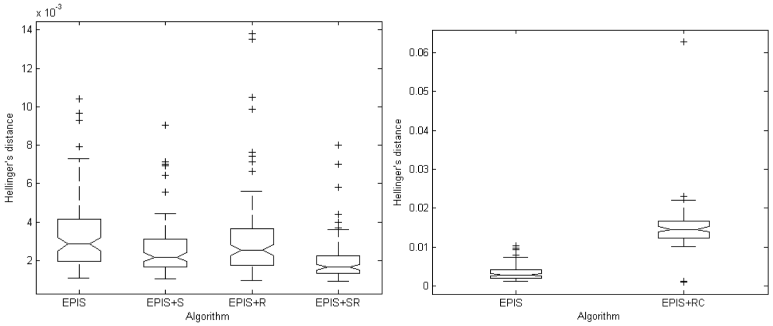
4.1 Review of the Performance of EPIS

Experimental results in [14,15] show that the EPIS-BN algorithm achieves a considerable improvement over the previous state of the art algorithm, the AIS-BN [2].

Furthermore, the results in [14,15] also show that the EPIS-BN algorithm already approaches the limit that sampling algorithms can achieve on CPCS and PATHFINDER, because the precision that it achieves on these networks is already in the same order as those of probabilistic logic sampling on the same networks without evidence. In the latter case, since there is no evidence in the networks, logic sampling samples from the optimal importance function, the prior distribution. We believe that precision so achieved is the limit of sampling algorithms.

4.2 Results of Proposed Heuristics on ANDES Network

In this experiment, we generated a total of 75 test cases for the ANDES network. These cases consisted of five sequences of 15 cases each. For each sequence, we randomly chose a different number of evidence nodes: $15, 20, 25, 30, 35$ respectively. We set α_r to be 0.8 and α_s to be 0.99. We switched off adaptive split-rejection control when the $cv^2(w)$ of the initial sample set was less than $cv_c = 3.0$. Since adaptive split-rejection control has two heuristics: adaptive split control (S) and adaptive rejection control (R), we performed experiments that aimed at disambiguating their roles. We denote EPIS-BN without any heuristic method as the EPIS algorithm, EPIS-BN with only adaptive split control as EPIS+S, EPIS-BN with only adaptive rejection control as EPIS+R, and EPIS-BN with both heuristics as EPIS+SR (ASREPIS-BN). We also tested the EPIS-BN with pure rejection control method proposed in [8], which is denoted as EPIS+RC. Again, the difference between our proposed adaptive rejection control and pure rejection control is that the former method applies rejection control periodically before finishing a complete sample, while the latter applies rejection control to complete samples. We compared the performance of EPIS, EPIS+S, EPIS+R, EPIS+SR and EPIS+RC and tested them on the same 75 test cases generated as described above. In order to be fair for all the algorithms, we let EPIS-BN run for $320K$ samples and let



(a) **(b)**

Fig. 2. (**a**) Results of our proposed heuristics, adaptive split control (S) and adaptive rejection control (R), when applied in the EPIS-BN algorithm for a fixed running time (same as **a**) on ANDES network; (**b**)The results of EPIS-BN and EPIS-BN with pure rejection control for a fixed running time (around 12s); Pure rejection control made the performance of EPIS-BN worse on ANDES network other algorithms run for the same amount of time. The size of the initial sample sets of EPIS+S, EPIS+R, EPIS+SR, and EPIS+RC are all $4K$. Boxplots of the errors of the algorithms are shown in Figure 2.

From the Figure 2**b**, we can see that the pure rejection control [8] made the performance of EPIS-BN worse, which is rightly pointed out in [1]. However, our proposed adaptive rejection control with several extensions, helped EPIS-BN achieve a better precision as Figure 2**a** shows. Figure 2**a** also shows that adaptive split control also helped EPIS-BN achieve a better precision. More dramatically, the two heuristics amplified each other in EPIS+SR (ASREPIS-BN) and helped it achieve a much better precision. The median errors for EPIS-BN, EPIS+S, EPIS+R and EPIS+SR are 0.0029, 0.0022, 0.0025, and 0.0017 respectively. A paired one-tail t-test indicates that that the improvement of ASREPIS-BN over EPIS-BN is significant at $p = 4.44 \times 10^{-18}$ level. We would like to stress that the improvement is achieved within the same time and with fewer samples. The ASREPIS-BN algorithm only generated, on the average, $170K$ samples. Therefore, we conclude that the quality of the samples generated by ASREPIS-BN is much better than that of EPIS-BN.

4.3 Fixed Number of Samples

In this experiment, we let both EPIS-BN and ASREPIS-BN run for $320K$ samples. Obviously, since we let ASREPIS-BN run for more samples than the last experiment, we can get even bigger improvement. Figure 3(a) shows the overall error of the results of the two algorithms. The median errors were 0.0032 for EPIS-BN and 0.0013 for ASREPIS-BN. ASREPIS-BN achieves an overall error that is less than half of that of EPIS-BN.

4.4 Results on CPCS and PathFinder

Since EPIS-BN already performs almost optimally on the CPCS and PathFinder networks, we used more conservative parameters in this experiment; We let rejection percentile α_r to be 0.2, split percentile α_s to be 0.999, and switch-off threshold cv_c to

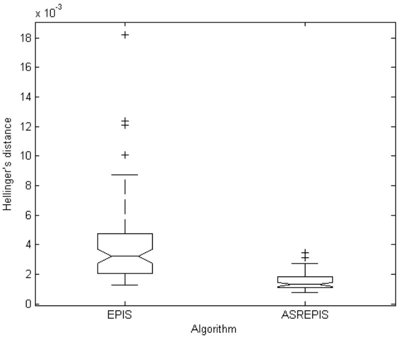
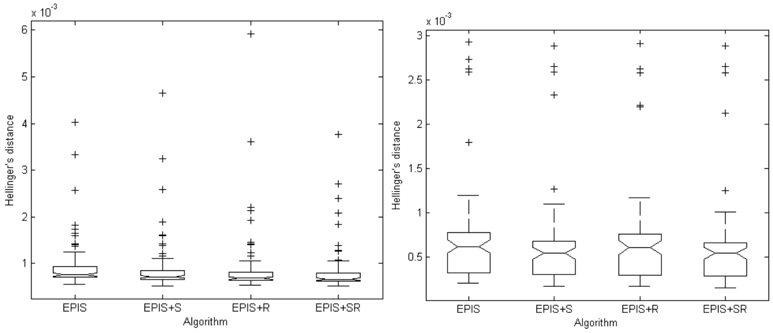


Fig. 3. Hellinger's distance of the EPIS-BN and ASREPIS-BN algorithms on ANDES network given a fixed number of samples (320K)



(a) **(b)**

Fig. 4. Results of our proposed heuristics, adaptive split control (S) and adaptive rejection control (R), when applied in the EPIS-BN algorithm for a fixed running time on (**a**) CPCS and (**b**) PATHFINDER

be 3.0. The results are shown in Figure 4. We can see that adaptive split-rejection control also brings some improvements for CPCS and PATHFINDER. The median errors for EPIS-BN, EPIS+S, EPIS+R, and EPIS+SR were 0.00076, 0.00072, 0.00069, 0.0006 on CPCS and 0.00063, 0.00055, 0.00061, 0.00055 on PATHFINDER respectively. Although the improvement is not much in comparison to that on ANDES, we again believe that this is due to ceiling effect: EPIS-BN already does very well on the two networks.

5 Conclusion

We propose the *adaptive split-rejection control* technique to stochastically reject samples with extremely small weights and adaptively split samples with extremely large weights for importance sampling in Bayesian networks. Our results show that the technique significantly improves the performance of EPIS-BN. Although adaptive split-rejection control may introduce correlation among the samples, the correlation is minimal in comparison to the reduction in the overall sample variance for sample sets with much oscillation. However, how to choose optimal split and rejection percentiles under different circumstances needs further exploration.

Acknowledgement

This work was partly done when the first author was at University of Pittsburgh and supported by the Air Force Office of Scientific Research grant FA9550–06–1–0243.

References

1. Y. Chen. Another look at rejection sampling through importance sampling. Technical report, Institute of Statistics and Decision Sciences, Duke University, 2004.
2. J. Cheng and M. J. Druzdzel. BN-AIS: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188, 2000.

3. G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, Mar. 1990.
4. J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339, 1989.
5. P. Grassberger. Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Physical Review E*, 56:3682–3693, Sept. 1997.
6. G. Kokolakis and P. Nanopoulos. Bayesian multivariate micro-aggregation under the Hellinger's distance criterion. *Research in official statistics*, 4(1):117–126, 2001.
7. F. Liang. Dynamically weighted importance sampling in monte carlo computation. *Journal of the American Statistical Association*, 97:807–821, 2002.
8. J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York, 2001.
9. J. S. Liu, R. Chen, and W. H. Wong. Rejection control and sequential importance sampling. *Journal of the American Statistical Association*, 93(443):1022–1031, 1998.
10. K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, pages 467–475, San Francisco, CA, 1999. Morgan Kaufmann Publishers.
11. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.
12. M. N. Rosenbluth and A. W. Rosenbluth. Monte Carlo calculation of the average extension of molecular chains. *Journal of Chemical Physics*, 23(256), 1955.
13. R. Y. Rubinstein. *Simulation and the Monte Carlo Method.* John Wiley & Sons, 1981.
14. C. Yuan and M. J. Druzdzel. An importance sampling algorithm based on evidence prepropagation. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI–03)*, pages 624–631, Morgan Kaufmann Publishers San Francisco, California, 2003.
15. C. Yuan and M. J. Druzdzel. Importance sampling algorithms for Bayesian networks: Principles and performance. *Mathematical and Computer Modelling*, 43:1189–1207, 2006.