Distributional Importance Sampling for Approximate Weighted Model Counting

Jessica Davies and Fahiem Bacchus

Department of Computer Science, University of Toronto, Toronto, Canada.
[jdavies|fbacchus]@cs.toronto.edu

Abstract. We present a sampling method to approximate the weighted model count of Boolean satisfiability problems. Our method is based on distributional importance sampling, where a subset of the variables are randomly set according to a backtrack-free distribution, and the remaining sub-formula is counted exactly. By using distributional samples (also known as Rao-Blackwellised samples), we can improve the accuracy of the approximation by reducing the variance of the samples. As well, distributional sampling allows us to exploit the power of dynamic component analysis performed by state-of-the-art exact counters. We discuss several techniques for providing a measure of confidence in the resulting estimates, including an analysis based on the Central Limit Theorem. Experiments on unweighted and weighted benchmarks demonstrate the promising performance of this approach.

1 Introduction

#SAT, or model counting, is the problem of counting the number of satisfying truth assignments to a CNF formula ϕ. Weighted model counting is the simple, but very useful, extension where each truth assignment π has a weight $wt(\pi)$, and the problem becomes that of computing the sum of the weights of the satisfying truth assignments (models) rather than simply counting their number.

Since there are a finite number of truth assignments (2^n where n is the number of variables in ϕ), we can assume without loss of generality that the sum of $wt(\pi)$ over all 2^n truth assignments is one. In this case, $wt(\pi)$ can be viewed as being a probability distribution over the truth assignments such that $P(\pi) = wt(\pi)$, and the weighted model count of a formula ϕ, $wt(\phi)$, can be viewed as being the probability (under that distribution) that ϕ is TRUE, $P(\phi)$. If we choose the uniform distribution where every truth assignment has weight $1/2^n$, then $P(\phi)$ is equal to the number of models (the unweighted model count) divided by 2^n.

This probabilistic interpretation naturally leads to capturing various forms of probabilistic inference using SAT encodings, an approach that has proved to very effective on many probabilistic reasoning problems, e.g., [22, 4]. This is arguably the most compelling application of #SAT.

There are different ways of encoding the truth assignment weights in a CNF formula, but one particularly simple method, proposed by Sang et al. [22], is to encode the weights by assigning a weight to each literal, so that the weight of any model π becomes simply the product of the weights of the literals it makes TRUE. Using this encoding (and others, e.g., [4]) most exact model counters can be easily modified to perform weighted model counting with no additional overhead.

However, despite the success of exact weighted model counters, application instances often lie beyond their reach. Therefore, much recent work has focused on approximation techniques for model counting [17, 14, 13, 10] and inference in probabilistic models [12, 9, 11], which can yield useful estimates when the exact answer cannot be computed.

All of these approximation techniques are based on the idea of using samples to estimate the model count or probability of a formula ϕ. Theoretically, the number of solutions of a Boolean formula can be estimated given a way to sample the solutions uniformly at random [16]. Sampling based methods for approximate #SAT such as SampleSat [25] and SampleCount [14] are based on this inspiration, augmented by various techniques to overcome the difficulty of generating solutions uniformly at random in practice. Additionally, traditional methods of approximating the expectation of a random variable such as importance sampling [8] have been applied to model counting [10], by defining a random variable whose expectation is the solution count.

In this paper we introduce a sampling method for *weighted* model counting. Since other probabilistic queries as well as Bayesian inference can be encoded as weighted model counting, our resulting approximator is of more general applicability. Like the work of [10] we utilize the technique of importance sampling, however in addition we introduce the use of distributional (or Rao-Blackwellised) samples to better exploit the recent advances in exact-model counting systems (e.g., Cachet [21]). The advantage of this approach is that each sample now covers more of the solution space, which can reduce the variability between samples yielding better estimates.

We also discuss three techniques for providing estimates of the true weighted count based on the samples, along with confidence measures. For example, following [14] we consider using the Markov inequality to provide a lower bound on the true count with high probability. We also present a statistical argument, based on the assumption that the samples come from a log-normal distribution, which results in a confidence interval that should contain the true count if the assumption is true. The assumption can be tested using the sample data. This is similar in spirit to the #SAT upper-bound produced in [17]. Finally, we show that the Central Limit Theorem can be applied to our sampling scheme, and use it to produce a confidence interval under the assumption that we have generated enough samples.

Our distributional importance sampling algorithm has been implemented in Minisat, using Cachet as the exact weighted counter. We compare the performance of our techniques with SampleCount [14] on unweighted model counting, and with IJGPSampleSearch [11] on weighted CNF's encoding probability of evidence queries in Bayesian networks.

The paper is organized as follows. In Section 2, we review the weighted model counting problem, and distributional importance sampling in the context of probability theory. Section 3 presents our algorithm for approximate weighted model counting, followed in Section 4 by a discussion of several possible confidence measures. A description of our implementation and the experimental results appears in Section 6. Section 5 compares our techniques to related work. The paper ends with a conclusion and plans for future work in Section 7.

2 Preliminaries

Weighted Model Counting: We use the following definitions and notation. Given a Boolean formula ϕ (which we assume to be in Conjunctive Normal Form) over the variables $\boldsymbol{V}$, we denote the set of literals of $\boldsymbol{V}$ by $L(\boldsymbol{V})$. A truth assignment π for the variables $\boldsymbol{V}$ is an assignment of TRUE or FALSE to each variable. A model π is a satisfying truth assignment for ϕ if ϕ is TRUE under the truth assignments made by π; this is denoted by $\pi \models \phi$. There will be $2^{|\boldsymbol{V}|}$ truth assignments, and we denote the set of all truth assignments for $\boldsymbol{V}$ by $\mathcal{M}(\boldsymbol{V})$, or simply by $\mathcal{M}$ when the set $\boldsymbol{V}$ is understood. We denote the set of models of ϕ, i.e., $\{\pi \in \mathcal{M} | \pi \models \phi\}$, by $sol(\phi)$.

In weighted model counting there is a weight function wt that assigns to each model $\pi \in \mathcal{M}$ a real-valued weight $wt(\pi)$. The weighted model counting problem is to determine $wt(\phi) = \sum_{\pi \in sol(\phi)} wt(\pi)$ for any formula ϕ: the sum of the weights of the satisfying models of ϕ. Typically, the weight function wt is encoded by adding to ϕ a new set of variables and clauses yielding an extended formula ϕ^+. Various encodings exist (see, e.g., [4]) but a particularly convenient one for our purposes is the encoding proposed by Sang et al. [22]. In this encoding each literal ℓ in the extended formula ϕ^+ is assigned a weight $wt(\ell)$ in the range [0–1], and the weight of any model π (of the extended formula ϕ^+) is simply the product of the weights of the literals it makes TRUE. The encoding ensures that $wt(\phi^+) = wt(\phi)$ and that the weight function is a probability distribution. Note that under this encoding of weights an unweighted model counting problem ϕ can be easily translated to a weighted problem. No new literals or clauses need to be introduced. Instead one need only assign weight 0.5 to each of the literals of the original formula. The model count can then be computed by multiplying the weighted model count by $2^{|\boldsymbol{V}|}$. It is not difficult to further refine the encoding of Sang et al. to ensure that for every literal, $wt(\ell) + wt(\neg\ell) = 1$. We use this addition refinement in our work.

Our sampling method utilizes partial truth assignments ρ. We denote the set of models π that extend a partial assignment ρ by $ext(\rho)$, and the set of satisfying models of a formula ϕ that extend ρ by $sol(\phi, \rho)$.

Distributional Importance Sampling: Suppose that $\overline{X}$ is a set of random variables. We use the corresponding lower case $\overline{x}$ to indicate a corresponding set of values for these random variables. The various values $\overline{x}$ of $\overline{X}$ occur with frequency given by the probability distribution $P(\overline{X})$. We are interested in a function f of $\overline{X}$, and particularly in the expected value of $f(\overline{X})$ under the distribution $P(\overline{X})$, i.e. $\mathrm{E}_P[f(\overline{X})] = \sum_{\overline{x}} P(\overline{x}) f(\overline{x})$. However, often we cannot calculate this expectation directly, for example, it is generally too time consuming to calculate $f(\overline{x})$ for the exponential number of sets of values $\overline{x}$.

We can, however, often approximate $\mathrm{E}_P[f(\overline{X})]$ using the technique of distributional importance sampling as follows. First we partition $\overline{X}$ into two sets of variables, the

prefix variables $\overline{X}_p$ and the remaining variables $\overline{X}_d$. Then

$$\begin{aligned}
\mathrm{E}_P[f(\overline{X})] &= \sum_{\overline{x}} P(\overline{x}) f(\overline{x}) = \sum_{\overline{x}_p, \overline{x}_d} P(\overline{x}_p, \overline{x}_d) f(\overline{x}_p, \overline{x}_d) \\
&= \sum_{\overline{x}_p, \overline{x}_d} P(\overline{x}_p) P(\overline{x}_d | \overline{x}_p) f(\overline{x}_p, \overline{x}_d) = \sum_{\overline{x}_p} P(\overline{x}_p) \sum_{\overline{x}_d} P(\overline{x}_d | \overline{x}_p) f(\overline{x}_p, \overline{x}_d) \\
&= \sum_{\overline{x}_p} P(\overline{x}_p) \mathrm{E}_{P(\overline{X}_d | \overline{x}_p)}[f(\overline{x}_p, \overline{X}_d)] \\
&= \mathrm{E}_P[g(\overline{X}_p)] \text{ where } g(\overline{x}_p) = \mathrm{E}_{P(\overline{X}_d | \overline{x}_p)}[f(\overline{x}_p, \overline{X}_d)]
\end{aligned}$$

We can now use importance sampling [8] to approximate $\mathrm{E}_P[g(\overline{X}_p)]$ by generating random sample assignments $\overline{x}_p$ to the variables $\overline{X}_p$. The idea of importance sampling is to draw samples from a sampling distribution Q different from the target distribution P. For example, Q can be chosen in order to avoid generating 0-probability samples, i.e. $\overline{x}_p$ such that $g(\overline{x}_p) = 0$. The only restriction on Q is that it must dominate P, i.e. if $P(\overline{x}_p) > 0$ then $Q(\overline{x}_p) > 0$. So

$$\begin{aligned}
\mathrm{E}_P[g(\overline{X}_p)] &= \sum_{\overline{x}_p} P(\overline{x}_p) g(\overline{x}_p) \\
&= \sum_{\overline{x}_p} P(\overline{x}_p) g(\overline{x}_p) \frac{Q(\overline{x}_p)}{Q(\overline{x}_p)} \qquad \text{since } Q(\overline{x}_p) = 0 \implies P(\overline{x}_p) = 0 \\
&= \mathrm{E}_Q[\frac{P(\overline{X}_p)}{Q(\overline{X}_p)} g(\overline{X}_p)] = \mathrm{E}_Q[w(\overline{X}_p) g(\overline{X}_p)] = \mathrm{E}_Q[\xi(\overline{X}_p)],
\end{aligned}$$

where $w(\overline{X}_p) = P(\overline{X}_p)/Q(\overline{X}_p)$ is the weighting function that corrects for the fact that we are sampling from Q not P, and $\xi(\overline{X}_p) = w(\overline{X}_p) g(\overline{X}_p)$ is a random variable representing the weighted samples. Note that by the above $\mathrm{E}_Q[\xi(\overline{X}_p)] = \mathrm{E}_P[f(\overline{X})]$.

Therefore, to approximate $\mathrm{E}_P[f(\overline{X})]$, we can generate M sample settings of $\overline{X}_p$, $\{\overline{x}_p[1], \ldots, \overline{x}_p[M]\}$, each randomly generated with probability $Q(\overline{x}_p[i])$ and calculate $\{g(\overline{x}_p[1]), \ldots, g(\overline{x}_p[M]\}$ exactly. Define the **unnormalized importance sampling estimator** to be

$$\hat{\mathrm{E}}_Q[w \cdot g] = \sum_{m=1}^{M} \xi(\overline{x}_p[m]) \Big/ M$$

Then $\hat{\mathrm{E}}_Q[w \cdot g]$ converges to $\mathrm{E}_P[f(\overline{X})]$ as $M \to \infty$ [8]. Its accuracy depends on the variability of the sample weights which increases as Q differs from P (as well as the variability of $g(\overline{X}_p)$). Therefore it is desirable to choose a sampling distribution Q that is as close to P as possible.

The key to this approach is that by reducing the original function f by setting the variables $\overline{X}_p$ to $\overline{x}_p$, we can then effectively compute all of the terms involved in the above equation, whereas with all variables unassigned it is too hard to compute $\mathrm{E}_P[f(\overline{X})]$ exactly.

3 Approximating Weighted Model Counting

In this section we present our sampling method for approximate weighted model counting. We cast the weighted model counting problem as the calculation of the expected value of a function over a distribution, and use distributional importance sampling to produce an estimate.

Let ϕ be a Boolean formula over variables $\boldsymbol{V}$ with associated weight function wt such that each literal ℓ is assigned a weight $wt(\ell) \in [0, 1]$ and each model π has weight equal to the product of the weights of the literals it makes TRUE (as described above). Under these conditions wt is a probability distribution P over $\mathcal{M}(\boldsymbol{V})$ (the models of $\boldsymbol{V}$). Let $f : \mathcal{M}(\boldsymbol{V}) \rightarrow \{0, 1\}$ be such that $f(\pi) = 1$ if and only if $\pi \in sol(\phi)$. Then

$$\begin{aligned}\sum_{\pi \in sol(\phi)} wt(\pi) &= \sum_{\pi \in \mathcal{M}(\boldsymbol{V})} wt(\pi) f(\pi) \\ &= \sum_{\pi \in \mathcal{M}(\boldsymbol{V})} P(\pi) f(\pi) \\ &= \mathrm{E}_P[f(\boldsymbol{V})]\end{aligned}$$

Since it is often infeasible to calculate this expectation exactly, we use distributional importance sampling to estimate it instead, thus obtaining an estimate for the weighted model count of ϕ. The distributional importance sampling approach outlined in the previous section requires us to partition the variables $\boldsymbol{V}$ into two sets, a set of prefix variables $\boldsymbol{V}_p$ and the remaining variables $\boldsymbol{V}_d$. We then generate M random samples $\{\rho[1], \ldots, \rho[M]\}$ from $\mathcal{M}(\boldsymbol{V}_p)$ using some distribution Q over these models. The derivation in Section 2 gives

$$\begin{aligned}\xi(\rho[m]) &= \frac{P(\rho[m])}{Q(\rho[m])} g(\rho[m]) \\ &= \frac{P(\rho[m])}{Q(\rho[m])} \mathrm{E}_{P(\boldsymbol{V}_d | \rho[m])}[f(\rho[m], \boldsymbol{V}_d)] \\ &= \frac{P(\rho[m])}{Q(\rho[m])} \sum_{\boldsymbol{v}_d} P(\boldsymbol{v}_d | \rho[m]) f(\rho[m], \boldsymbol{v}_d) \\ &= \frac{P(\rho[m])}{Q(\rho[m])} \sum_{\boldsymbol{v}_d} \frac{P(\rho[m], \boldsymbol{v}_d)}{P(\rho[m])} f(\rho[m], \boldsymbol{v}_d) \\ &= \frac{1}{Q(\rho[m])} \sum_{\boldsymbol{v}_d} P(\rho[m], \boldsymbol{v}_d) f(\rho[m], \boldsymbol{v}_d) \\ &= \frac{1}{Q(\rho[m])} \sum_{\pi \in sol(\phi, \rho[m])} wt(\pi)\end{aligned}$$

Thus to compute the estimator $\hat{\mathrm{E}}_Q[w \cdot g] = \sum_{m=1}^{M} \xi(\rho[m]) / M$ it is necessary to know for each sample $\rho[m]$, the probability of producing it (i.e. $Q(\rho[m])$), and the weighted

model count of the solutions to ϕ extending it (i.e. $\sum_{\pi \in sol(\phi, \rho[m])} wt(\pi)$). We describe below how to compute $Q(\rho[m])$ by describing how we generate random samples; the term $\sum_{\pi \in sol(\phi, \rho[m])} wt(\pi)$ can be calculated by an exact weighted model counter given the original formula ϕ with unit clauses added for each literal in the prefix, i.e., $\phi \cup \{(\ell)\}_{\ell \in \rho[m]}$. The method works when the simplified formula is significantly easier to count than the original formula. This can occur in practice when, for example, the setting $\rho[m]$ sufficiently reduces the tree-width of the formula, or sufficiently constrains the formula so that it becomes easy to solve.

It remains to choose a set $\boldsymbol{V}_p$ from $\boldsymbol{V}$ to be the prefix variables, and to specify a distribution Q over $\boldsymbol{V}_p$. These choices are of primary importance to the efficiency and accuracy of the resulting algorithm. For example, if $\boldsymbol{V}_p$ is a backdoor set [19], then exactly counting the residual formula $\phi|_{\rho[m]}$ will be possible in polynomial time. Although there is currently no practical method of finding backdoors for model counting, it may be possible to reduce the complexity of the formula by choosing variables that reduce the tree-width of the formula since exact model counters like Cachet operate efficiently on formulas of low tree-width [1].

Besides making it easier to compute the exact weighted model count of the residual formula, an additional consideration is reducing the variance among the samples $\rho[m]$. Methods such as Iterative Join Graph Propagation [10], Belief Propagation [17], or solution sampling [14] can be used to estimate the probability a variable is TRUE among the set of satisfying models of ϕ, i.e., $P(v|\phi) = P(v, \phi)/P(\phi) = \sum_{\pi \in sol(\phi, v)} wt(\pi) / \sum_{\pi \in sol(\phi)} wt(\pi)$. Variables that have probability around 0.5 in the solution space may be suitable for inclusion in the prefix, since setting them to either truth value leaves similarly weighted sub-formulas. This can help to reduce the variability of the sample values, leading to a more accurate estimation of the true weighted count [18].

Once the prefix variables are chosen, the sampling distribution Q over $\boldsymbol{V}_p$ is computed with the goal of avoiding the generation of samples that cannot be extended to positive-weight solutions of ϕ. Otherwise, the predominance of non-solutions to ϕ among the truth assignments $\mathcal{M}(\boldsymbol{V})$ may lead to an under-approximation of the true weighted solution count, since most of the samples generated will produce $g(\boldsymbol{v}_p[m]) = 0$. Formally, Q is defined by first setting an ordering o over the prefix variables. Then Q is given in factored form by $Q(v_{o(i)}|v_{o(1)}, ..., v_{o(i-1)})$ for $1 \leq i \leq |\boldsymbol{V}_p|$. Letting ϕ_i^+ be $\phi \cup \{o(j)\}_{j=1,...,i-1}$ then $Q(v_{o(i)}|v_{o(1)}, ..., v_{o(i-1)})$ is equal to

1. $wt(v_{o(i)})$ if both $P(\phi_i^+ \cup \{(v_{o(i)})\}) > 0$ and $P(\phi_i^+ \cup \{(\neg v_{o(i)})\}) > 0$.
2. 1 if $P(\phi_i^+ \cup \{(\neg v_{o(i)})\}) = 0$.
3. 0 if $P(\phi_i^+ \cup \{(v_{o(i)})\}) = 0$.

This is equivalent to the backtrack-free distribution P^F of [10]. A SAT-solver can be used to generate the samples, by setting the variables of $\boldsymbol{V}_p$ in the order o, randomly according to $Q(v_{o(i)}|v_{o(1)}, ..., v_{o(i-1)})$. The SAT-solver is invoked at each step to determine if at least one positive-weight solution π to ϕ lies under each setting of $v_{o(i)}$. Note that if the generated solution π is saved at each step, it is only necessary to check one setting of $v_{o(i)}$, since the other is already known to be extendable to the value in π.

In practice, it can sometimes be hard to perform the SAT tests required to follow the Q distribution. In our system we utilized Minisat at each stage to perform the SAT

test, however in some cases a local search engine like WalkSat [24] might have better performance. If the SAT test is too difficult, one could always use $wt(v_{o(i)})$ as the value for the Q distribution. This will not affect the correctness of the derivation given above, but it might lead to generating zero weight samples.

4 Confidence Measures

In the previous section, we presented a method to generate samples, $\rho[m]$, compute a corresponding set of weighted samples, $\xi[m]$, and use the weighted samples to compute an estimate $\hat{\mathrm{E}}_Q[w \cdot g]$ that approaches the true weighted model count $wt(\phi)$ as the number of samples goes to infinity. However, this is not the only way weighted samples computed from our sampling procedure can be used to estimate $wt(\phi)$. In this section we examine three different ways to provide estimates along with confidence measures that indicate how likely it is that the estimate stands in a particular relationship with the true value of $wt(\phi)$.

Markov Lower Bound: This method has been used extensively in the context of approximate model counting [15, 14, 17, 10] and probability of evidence in Bayes nets [9]. Markov's inequality states that for any random variable X, $P(|X|/\alpha > \mathrm{E}[|X|]) < 1/\alpha$, where $|X|$ is the sample mean of X. In our context, $\xi(\boldsymbol{V}_p)$ is a random variable from which we generate M independent samples $\{\xi(\rho[1]), \ldots, \xi(\rho[M])\}$, and which is always ≥ 0. Let $c = \min(\xi(\rho[1]), \ldots, \xi(\rho[M]))$. Therefore by Markov's inequality

$$P(c/\alpha > \mathrm{E}_Q[\xi(\boldsymbol{V}_p)]) < \frac{1}{\alpha^M},$$

since $\xi(\boldsymbol{V}_p)$ yielded M independent samples all greater than or equal to c. As shown in Section 2, $\mathrm{E}_Q[\xi(\boldsymbol{V}_p)] = \mathrm{E}_P[f(\boldsymbol{V})]$, i.e., the weighted model count of ϕ. Therefore we obtain a lower bound on the true weighted model count. For example, to obtain a lower bound on the true model with 99% probability using M samples, we take c/α where $\alpha = 10^{2/M}$. The advantage to this approach is that only a few samples need to be taken in order to calculate a high probability lower bound, for example, with 100 samples $1/\alpha = 1/10^{1/50} = 0.955$, so the 99% probability estimate is still 95.5% of the minimum valued sample. However, if the weighted samples $\xi(\rho[m])$ vary considerably, their minimum value may be a very conservative lower bound on the true weighted count.

Cox's Confidence Interval for the Log-Normal Mean: This method is based on the observation that the samples $\{\xi(\rho[m])\}_{m=1}^{M}$ often exhibit characteristics of the log-normal distribution, mainly, that most values are close together and small relative to the sample mean, with just a few much larger outliers. If we believe that a random variable X is likely to be log-normally distributed, then we can use statistical methods to estimate the mean of X.

Therefore, the first step is to determine if $\xi(\boldsymbol{V}_p)$ is likely to be log-normal, which is the case exactly when $Y(\boldsymbol{V}_p) = \log(\xi(\boldsymbol{V}_p))$ is normally distributed. To test this hypothesis, we apply the Wilk-Shapiro test for normality to the $\{Y_m = \log(\xi(\rho[m]))\}_{m=1}^{M}$ [23,

20]. This test outputs the probability that the $\{Y_m\}$ are observed by sampling Y, given the hypothesis that Y is normally distributed. If it is very unlikely that these values would be produced by sampling a normal distribution, then the test fails. If the test for normality is passed, we then calculate a confidence interval for the expectation of $\xi(\boldsymbol{V}_p)$, which will contain the true value with 99% probability, assuming that $\xi(\boldsymbol{V}_p)$ is truly log-normal. Several methods of constructing confidence intervals for the mean of a log-normal random variable have been compared in [27]. We use Cox's method, which gives the confidence interval

$$e^{\bar{Y}+\frac{S^2}{2}-Z_{1-\alpha/2}\sqrt{\frac{S^2}{M}+\frac{S^4}{2(M-1)}}} \leq \mathrm{E}_Q[\xi(\boldsymbol{V}_p)] \leq e^{\bar{Y}+\frac{S^2}{2}+Z_{1-\alpha/2}\sqrt{\frac{S^2}{M}+\frac{S^4}{2(M-1)}}},$$

where $\bar{Y} = (1/M)\sum_{m=1}^{M} Y_m$ is the sample mean of $Y = \log(\xi(\boldsymbol{V}_p))$, $S^2 = \sum_{m=1}^{M}(Y_m - \bar{Y})^2/(M-1)$ is the sample variance of Y, and $Z_{1-\alpha/2}$ is the $(1-\alpha/2)$-percentile of the standard normal distribution. We take $\alpha = 0.01$ to obtain 99% confidence.

There are two main advantages to this approach. First, since the estimate is based on all M samples, it may be less conservative than the Markov lower bound, and secondly it also provides an upper bound on the weighted model count. However, its 99% confidence is undermined by our uncertainty of the distribution of $\xi(\boldsymbol{V}_p)$, since the Wilk-Shapiro test does not *guarantee* log-normality. If the $\{\xi(\rho[m])\}$ do not come from a log-normal distribution, then the mean of $\xi(\boldsymbol{V}_p)$ could be well outside the calculated bounds.

Central Limit Theorem: Consider repeating our distributional importance sampling algorithm many times on the same formula ϕ. How similar would the sample means $\hat{\mathrm{E}}_Q[w \cdot g] = \sum_{m=1}^{M} \xi(\rho[m])/M$ be over these many runs? Or more specifically, what is the distribution of the $\{\hat{\mathrm{E}}_Q[w \cdot g]\}$? The Central Limit Theorem answers this question in the limit as M goes to infinity.

Theorem 1 (e.g., [7]). *The Central Limit Theorem: Given M independent, identically distributed random variables $\{X_m\}_{m=1}^{M}$ with finite mean μ and finite standard deviation σ, the distribution of $\Psi = \frac{(\sum_{m=1}^{M} X_m) - M\mu}{\sigma\sqrt{M}}$ converges to the standard normal distribution $N(0,1)$ as M goes to infinity.*

In other words, $\lim_{M\to\infty} P(\Psi \leq \beta) = \Phi_{0,1}(\beta)$, where $\Phi_{0,1}$ is the cumulative distribution function of the standard normal distribution. We can rearrange this to produce $\lim_{M\to\infty} P(\hat{X} - \mu \leq \beta\sigma/\sqrt{M}) = \Phi_{0,1}(\beta)$, where $\hat{X}$ is the average of the random variables $\{X_m\}$. Therefore, if our number of samples M is large enough, we obtain an interval around the sample mean that contains the true mean with probability $1-\alpha$:

$$\mathrm{E}[X] = \mu \in [\hat{X} \pm \Phi^{-1}(1-\alpha)\sigma/\sqrt{M}],$$

where X has the same distribution as the $\{X_m\}$. If the standard deviation σ of X is also unknown, we can take the sample standard deviation as an estimate, $\sigma \approx S = \sqrt{\sum_{m=1}^{M}(X_m - \hat{X})^2/(M-1)}$, as the sample standard deviation also approaches the true standard deviation as M goes to infinity.

In our context, it is possible to prove that $\xi(\boldsymbol{V}_p)$ satisfies all of the conditions of the theorem, and we can use the above equation with the sample mean and sample variances (in place of $\hat{X}$ and σ respectively) to bound $\mathrm{E}_Q[\xi(\boldsymbol{V}_p)]$.

This estimate introduces uncertainty in that we do not know how large M has to be for the sample mean to become normally distributed, or for the sample variance to be a reasonable estimate of the true variance. However, the estimated interval may be tighter than that provided by Cox's method, and this analysis can still be applied when the weighted samples $\{\xi(\rho[m])\}$ are very unlikely to be log-normally distributed as indicated by the Wilk-Shapiro test. Furthermore, one can perform various tests on random subsets of the samples to see if the distribution of the sample means of these subsets is close to normal.

5 Related Work

In this section we discuss the existing work in approximate model counting and constraint-based methods of approximating probability of evidence in Bayes nets that is closest to our distributional importance sampling technique.

In [14, 17], a combination of distributional samples and an exact model counter is used to generate samples for approximate model counting, and the Markov inequality is applied to produce lower bounds. In these respects the work is similar to ours, but the distribution from which they generate the samples is rather different. For example, every sample has a different set of prefix variables, and no work is done to check the unsatisfiability of alternative settings of the prefix variables. In fact the distribution from which the samples are drawn is unknown, so importance sampling is not applicable. The advantage to this approach is that the samples may be faster to generate, and the sampling method is more flexible since each sample's prefix can contain different variables.

Kroc et al. also used the Wilk-Shapiro test for normality, and a method of calculating a confidence interval for the log-normal mean, similar to our analysis [17]. However, they apply this analysis to a different random variable (the number of choice-points a SAT-solver encounters on a solution branch) to obtain an upper bound on the solution count.

Gogate and Dechter directly apply the importance sampling framework to the model counting problem in [10], and use a backtrack-free distribution as the sampling distribution. However, they do not use distributional samples, so each sample is one solution to the boolean formula. When solutions exist under each setting of a variable, the setting is randomly chosen according to an Iterative Join graph propagation estimate of its marginal probability. Also, they consider using an approximation to the backtrack-free distribution when the exact calculation requires too much search.

Gogate and Dechter also apply very similar techniques to the problem of approximating the probability of evidence in Bayes nets [11]. In addition to the differences mentioned in the context of model counting, their approach only encodes the 0-probability entries of the Bayes net's conditional probability tables as constraints, rather than our encoding of the entire problem to weighted CNF. We intend to investigate the advantages of a full CNF encoding further.

In [13, 9], the authors also address the issue of providing a confidence measure for the estimates generated by importance sampling. The Markov inequality is invoked and

refinements are investigated, both in the context of approximate model counting and probability of evidence in Bayes nets.

6 Experiments

We implemented the distributional sampling algorithm described in Section 3 and the three confidence measures from Section 4 using a modified version of Minisat [6] and Cachet [22]. We first describe the details of the implementation, which we call WAC (weighted approximate counting), and then present the results of tests on unweighted and weighted CNF instances, in comparison to two existing techniques.

Implementation: We use Cachet-wmc version 2.0 [26] to perform the exact weighted counting of the residual formulas. Once the prefix variables and ordering are chosen, we use a modified version of Minisat to test the satisfiability of each setting of the variables and set variables that have solutions on both sides randomly according to their weight, in order to generate samples from the distribution Q. For probability of evidence in Bayes nets, we use the encoding of [22]. We use two methods of selecting the prefix variables. For the weighted benchmarks that encode the probability of evidence in Bayes nets, we randomly choose variables from among the State variables of the encoding, which correspond to the original Bayes net nodes. For the unweighted benchmarks, we found that a completely random approach produced prefixes that were too long and thus expensive to sample according to Q, while shorter prefixes made counting the residual formula with Cachet infeasible. Therefore, we use C2D [5] to generate a decomposition tree, and randomly choose variables that occur in the largest separator sets in order to exploit the structure of the instances. In both the weighted and unweighted case, we determine the best size for the prefix by generating one sample for each possible size, in increments of 5% of the maximum size. We take the prefix size that minimizes the time for the Minisat and Cachet steps combined. This search for the best prefix size is limited by a strict timeout: each possible prefix size was tested for a max of 60s.

Experiments: All experiments were conducted on a cluster of 8 Intel Xeon 2.00GHz processors with a total of 16GB of RAM. For the unweighted model counting case, we compare our implementation against the approximate model counter SampleCount [14] on the benchmarks used in that paper [2]. We omit the results for those cases where our technique could not generate at least one sample within the 5000s timeout. The remaining results are shown in Table 1. The values in the column 'True Count' are taken from [14]. The 'SampleCount' columns show the lower bounds calculated by SampleCount and its runtimes. The next five columns show our estimates (lower and upper bounds) as given by the three techniques from Section 4. The estimates using Cox's method are omitted for cases where the Wilk-Shapiro test for log-normality failed. The last three columns of the table show the size of the prefix as a percentage of the total number of boolean variables, the number of samples M, and the time taken to generate the samples by our distributional importance sampling method.

Comparing the SampleCount lower bounds with our Markov lower bounds, we see that on every instance our technique is several orders of magnitude less. This could be due to the fact that SampleCount employs techniques aimed at reducing the variance of the samples (and thus tending to generate fewer outlier small minimums) that are currently not part of our implementation. The Wilk-Shapiro test for normality is passed by 8 out of 17 cases. Four of the upper bounds generated by the Cox technique are less than

		SampleCount		WAC							
	True	Lower	Time	Markov	Cox		CLT		$\frac{\lvert V_p \rvert}{\lvert V \rvert}$		Time
Instance	Count	Bound	(s)	LB	LB	UB	LB	UB	%	M	(s)
2bitmax	$2.1e^{29}$	$6.39e^{27}$	5	$1.41e^{22}$	–	–	$\mathbf{5.60e^{28}}$	$1.53e^{29}$	10	1000	11
fclq-18	$\geq 2.4e^{33}$	$8.07e^{46}$	52	$6.77e^{38}$	–	–	0	$1.81e^{53}$	55	902	210
fclq-20	$\geq 8.6e^{38}$	$1.07e^{59}$	84	$1.36e^{50}$	$1.97e^{59}$	$1.49e^{63}$	0	$6.35e^{59}$	40	139	5049
w.3.100	$1.8e^{21}$	$1.29e^{20}$	1941	$1.47e^{16}$	$\mathbf{1.79e^{21}}$	$5.11e^{21}$	$1.60e^{20}$	$\mathbf{5.08e^{21}}$	47	1000	8
w.3.150	$1.4e^{14}$	$1.47e^{12}$	2	$3.26e^{05}$	$\mathbf{1.56e^{13}}$	$1.00e^{14}$	0	$4.55e^{13}$	25	1000	16
w.4.100	$\geq 1.0e^{14}$	$1.53e^{15}$	3909	$2.51e^{07}$	$4.33e^{16}$	$3.84e^{17}$	0	$1.16e^{17}$	60	1000	12
ls8	$5.4e^{11}$	$7.78e^{09}$	4	$4.18e^{06}$	–	–	$\mathbf{1.62e^{11}}$	$4.95e^{11}$	45	1000	43
ls9	$3.8e^{17}$	$4.98e^{14}$	7	$1.71e^{10}$	$\mathbf{4.19e^{16}}$	$1.83e^{17}$	0	$2.89e^{17}$	45	1000	193
ls10	$7.6e^{24}$	$2.50e^{21}$	15	$5.25e^{13}$	–	–	$\mathbf{2.81e^{22}}$	$2.73e^{23}$	50	1000	1093
ls11	$5.4e^{33}$	$\mathbf{4.74e^{26}}$	25	$2.87e^{17}$	–	–	0	$5.81e^{28}$	15	1000	7836
ls12	$\geq 4.6e^{07}$	$5.78e^{36}$	54	$2.01e^{27}$	–	–	$1.69e^{31}$	$2.08e^{32}$	30	22	5169
lang12	$1.0e^{05}$	$1.20e^{03}$	57	$6.37e^{01}$	–	–	$\mathbf{1.89e^{04}}$	$7.85e^{04}$	15	1000	152
lang15	$3.0e^{07}$	$1.55e^{04}$	203	$2.54e^{02}$	–	–	$\mathbf{4.73e^{05}}$	$1.44e^{06}$	15	1000	995
lang19	$2.1e^{11}$	$\mathbf{1.40e^{08}}$	640	$7.03e^{03}$	$2.72e^{06}$	$5.37e^{08}$	$4.93e^{05}$	$2.36e^{07}$	10	30	209
lang20	$2.6e^{12}$	$\mathbf{3.56e^{08}}$	909	$1.64e^{04}$	–	–	0	$2.58e^{08}$	5	2	5019
lang23	$3.7e^{15}$	$\mathbf{5.74e^{09}}$	7704	$6.64e^{03}$	$2.64e^{07}$	$1.40e^{12}$	0	$1.52e^{09}$	5	22	422
lang28	$\geq 1.1e^{04}$	–	–	$5.09e^{05}$	$2.46e^{11}$	$1.30e^{13}$	0	$3.33e^{12}$	10	152	1594

Table 1. Approximate model counting results on the unweighted benchmarks. '–' in the SampleCount column indicates that it found no solutions with its cutoff set to 100000. In the Cox columns, '–' indicates that the Wilk-Shapiro test failed. Bold font indicates the best correct lower and upper bounds when the exact count is known.

the true count, but on w.3.100 the true count is contained within the estimated interval. Another advantage of the Cox method is that all but two lower bounds it generated are correct, and these are significantly higher than SampleCount's lower bounds. This suggests that the Cox technique can be best applied to generation of lower bounds. Similarly, the upper bounds derived using the Central Limit Theorem analysis are all lower than the true count except for the w.3.100 instance. The CLT lower bounds are mostly about an order of magnitude smaller than the Cox lower bounds, unless no lower bound is given, which occurred in nine of the 17 instances. The runtimes for our method and SampleCount are comparable, with each technique doing better on different instances.

For the weighted case, we compare our technique against IJGPSampleSearch [11], on the probability of evidence, P(e), instances from the Uncertainty in Artificial Intelligence 2006 approximate inference competition [3]. The exact probability of evidence, and other characteristics of these instances including the number of Bayes net nodes, the size of the largest variable domain and the number of evidence variables, are shown in Table 2. We encoded the Bayes nets into weighted CNF using the encoding of [22], and we can see from the table that most problems are infeasible for Cachet within the 5000s timeout. The results of the approximate methods are shown in Table 3. We do not show the results for instances where our method could not generate at least one sample within the 5000s timeout. We ran our distributional importance sampling method until either 1000 samples were generated or the 5000s time limit was reached. The number of samples is shown in the last column of the table. We then ran IJGPSampleSearch for the same time as our method had run. This time is shown in the third column of the table. IJGPSampleSearch produces an estimate without guarantees, and this value is shown in the 'IJGPSS P(e)' column. The next five columns show the lower and upper bounds generated by the confidence measure techniques in Section 4 given our samples generated by distributional importance sampling. The estimates using Cox's method are

Instance (N, |D|, |E|)	True P(e)	Cachet Time (s)	Instance (N, |D|, |E|)	True P(e)	Cachet Time (s)
BN_0 (100,2,26)	$1.28e^{-009}$	36	BN_46 (499,2,10)	$1.92e^{-003}$	711
BN_1 (100,2,18)	$6.65e^{-007}$	–	BN_69 (777,36,78)	$5.28e^{-054}$	–
BN_2 (100,2,22)	$3.02e^{-008}$	–	BN_70 (2315,36,159)	$2.00e^{-071}$	N/A
BN_3 (100,2,36)	$2.76e^{-013}$	200	BN_71 (1740,36,202)	$5.12e^{-111}$	N/A
BN_4 (100,2,51)	$3.59e^{-018}$	38	BN_72 (2155,36,252)	$4.21e^{-150}$	N/A
BN_5 (125,2,55)	$1.84e^{-019}$	409	BN_73 (2140,36,216)	$2.26e^{-113}$	N/A
BN_6 (125,2,71)	$4.29e^{-026}$	83	BN_74 (749,36,66)	$3.75e^{-045}$	–
BN_7 (95,2,30)	$9.63e^{-008}$	1107	BN_75 (1820,36,155)	$5.88e^{-091}$	N/A
BN_8 (100,2,9)	$4.08e^{-003}$	–	BN_76 (2155,36,169)	$4.93e^{-110}$	N/A
BN_9 (105,2,13)	$2.71e^{-004}$	–	BN_77 (1020,45,135)	$6.88e^{-079}$	–
BN_10 (85,2,17)	$6.24e^{-006}$	362	BN_78 (54,2,10)	$1.83e^{-003}$	5
BN_11 (105,2,46)	$7.96e^{-018}$	4062	BN_80 (360,2,50)	$1.31e^{-003}$	723
BN_12 (90,2,11)	$2.46e^{-004}$	–	BN_82 (360,2,50)	$5.57e^{-007}$	2254
BN_13 (125,2,9)	$4.78e^{-003}$	–	BN_84 (360,2,50)	$1.81e^{-001}$	4944
BN_14 (115,2,30)	$9.66e^{-010}$	–	BN_86 (422,2,50)	$4.11e^{-001}$	–
BN_15 (120,2,19)	$1.99e^{-006}$	–	BN_88 (422,2,50)	$7.61e^{-001}$	–
BN_16 (2127,6,100)	$8.33e^{-001}$	–	BN_92 (422,2,50)	$8.06e^{-001}$	–
BN_18 (2127,6,100)	$8.21e^{-001}$	–	BN_94 (53,50,6)	$4.49e^{-011}$	–
BN_42 (880,2,10)	$4.31e^{-003}$	277	BN_96 (54,50,5)	$2.30e^{-009}$	–
BN_43 (880,2,10)	$4.90e^{-003}$	942	BN_98 (57,50,6)	$2.14e^{-011}$	–
BN_44 (880,2,10)	$2.05e^{-004}$	17	BN_100 (58,50,8)	$1.89e^{-014}$	–
BN_45 (880,2,10)	$1.28e^{-002}$	1180	BN_102 (76,50,15)	$1.96e^{-026}$	–

Table 2. The exact probability of evidence for UAI'06 competition Bayes nets. The number of nodes, maximum domain size of a variable, and number of evidence variables are given by (N, |D|, |E|) for each instance. Cachet was run with a 5000 second timeout. 'N/A' in the Cachet columns indicates that Cachet crashed.

omitted for cases where the Wilk-Shapiro test failed. The prefix sizes used for the distributional samples are shown in column '$|V_p|$ % of N' as a percentage of the number of Bayes net variables.

The Markov lower bounds generated by our distributional samples are all correct, except for the BN_80 instance, where the true probability of evidence is $1.31e^{-3}$ and the estimated lower bound is $1.36e^{-3}$. The lower bounds produced by Cox's methods are all higher than the estimate returned by IJGPSampleSearch, but many are also larger than the true count so this estimation method may be misleading. However, the lower bounds produced by the Central Limit Theorem analysis are all correct in the sense that they are lower than the true probability of evidence. Additionally, the Central Limit Theorem lower bounds are significantly higher than the Markov inequality ones, indicating that the CLT analysis might be better for lower bound estimation than the Markov inequality. In 14 of the 44 cases, the CLT gives no useful lower bound(see e.g. BN_2). In 21 of the 44 instances, the CLT upper bound is also correct, which shows that this method of generating an interval estimate can be useful for approximating the probability of evidence. But in general we see that the estimates most often error on the low side.

7 Conclusion and Future Work

In this paper we have investigated an approximation technique applicable to weighted model counting. The method exploits SAT based techniques to improve sampling in two ways. First, it employs a SAT solver to allow samples to be generated from a backtrack free distribution. This guides the sampling process away from zero weight samples,

Instance	True P(e)	Time (s)	IJGPSS P(e)	WAC Markov LB	WAC Cox LB	WAC Cox UB	WAC CLT LB	WAC CLT UB	$\|V_p\|$ % N	M
ALARM										
BN_0	$1.28e^{-009}$	20	$8.88e^{-012}$	$5.40e^{-0018}$	$6.02e^{-009}$	$5.38e^{-008}$	$\mathbf{2.32e^{-010}}$	$\mathbf{1.88e^{-009}}$	60	1000
BN_1	$6.65e^{-007}$	29	$4.21e^{-009}$	$5.85e^{-0015}$	–	–	$\mathbf{1.39e^{-007}}$	$4.22e^{-007}$	65	1000
BN_2	$3.02e^{-008}$	41	$3.74e^{-011}$	$\mathbf{2.43e^{-0018}}$	–	–	0	$\mathbf{4.04e^{-008}}$	60	1000
BN_3	$2.76e^{-013}$	27	$5.80e^{-014}$	$5.85e^{-0020}$	$\mathbf{1.15e^{-013}}$	$\mathbf{4.38e^{-013}}$	0	$6.71e^{-013}$	45	1000
BN_4	$3.59e^{-018}$	31	$8.44e^{-019}$	$1.49e^{-0023}$	$\mathbf{3.12e^{-018}}$	$9.04e^{-018}$	$1.62e^{-018}$	$\mathbf{5.53e^{-018}}$	50	1000
BN_5	$1.84e^{-019}$	43	$7.83e^{-021}$	$1.09e^{-0029}$	–	–	$\mathbf{7.94e^{-021}}$	$3.32e^{-020}$	65	1000
BN_6	$4.29e^{-026}$	62	$3.70e^{-027}$	$2.99e^{-0031}$	$\mathbf{3.44e^{-026}}$	$9.01e^{-026}$	$9.35e^{-027}$	$\mathbf{7.29e^{-026}}$	50	1000
BN_7	$9.63e^{-008}$	36	$2.26e^{-011}$	$1.07e^{-0015}$	–	–	$\mathbf{2.81e^{-008}}$	$\mathbf{1.22e^{-007}}$	50	1000
BN_8	$4.08e^{-003}$	30	$6.53e^{-006}$	$6.20e^{-0016}$	$2.01e^{-002}$	$\mathbf{5.48e^{-001}}$	$\mathbf{3.65e^{-005}}$	$1.21e^{-003}$	75	1000
BN_9	$2.71e^{-004}$	33	$2.49e^{-006}$	$\mathbf{1.99e^{-0015}}$	$5.49e^{-004}$	$6.52e^{-003}$	0	$\mathbf{2.18e^{-003}}$	70	1000
BN_10	$6.24e^{-006}$	24	$2.71e^{-006}$	$1.53e^{-0015}$	–	–	$\mathbf{1.79e^{-006}}$	$\mathbf{7.66e^{-006}}$	55	1000
BN_11	$7.96e^{-018}$	44	$3.60e^{-021}$	$2.92e^{-0026}$	–	–	$\mathbf{2.39e^{-018}}$	$7.89e^{-018}$	70	1000
BN_12	$2.46e^{-004}$	46	$6.38e^{-007}$	$\mathbf{1.51e^{-0014}}$	$4.75e^{-004}$	$5.75e^{-003}$	0	$\mathbf{3.28e^{-004}}$	70	1000
BN_13	$4.78e^{-003}$	68	$2.55e^{-007}$	$1.24e^{-0017}$	–	–	$\mathbf{9.83e^{-005}}$	$5.77e^{-004}$	85	1000
BN_14	$9.66e^{-010}$	37	$2.41e^{-012}$	$\mathbf{6.27e^{-0018}}$	$1.53e^{-009}$	$9.90e^{-009}$	0	$\mathbf{3.52e^{-009}}$	60	1000
BN_15	$1.99e^{-006}$	62	$2.85e^{-011}$	$\mathbf{9.04e^{-0018}}$	$7.22e^{-006}$	$1.42e^{-004}$	0	$\mathbf{4.11e^{-006}}$	75	1000
BN_16	$8.33e^{-001}$	3045	$8.85e^{-070}$	$1.95e^{-2049}$	–	–	$\mathbf{9.04e^{-1775}}$	$2.27e^{-1774}$	65	380
BN_18	$8.21e^{-001}$	2645	$6.61e^{-065}$	$1.73e^{-1906}$	–	–	$\mathbf{1.06e^{-1704}}$	1	60	130
ISCAS85										
BN_42	$4.31e^{-003}$	89	$3.82e^{-007}$	$5.56e^{-0009}$	–	–	$\mathbf{2.19e^{-003}}$	$\mathbf{6.68e^{-003}}$	5	1000
BN_43	$4.90e^{-003}$	50	$4.00e^{-004}$	$2.22e^{-0007}$	–	–	$\mathbf{2.82e^{-003}}$	$\mathbf{7.66e^{-003}}$	5	1000
BN_44	$2.05e^{-004}$	47	$4.12e^{-005}$	$1.48e^{-0008}$	–	–	$\mathbf{1.03e^{-004}}$	$\mathbf{2.08e^{-004}}$	5	1000
BN_45	$1.28e^{-002}$	52	$7.26e^{-004}$	$1.39e^{-0009}$	–	–	$\mathbf{2.17e^{-003}}$	$9.93e^{-003}$	5	1000
BN_46	$1.92e^{-003}$	34	$2.49e^{-004}$	$2.43e^{-0004}$	–	–	$\mathbf{1.68e^{-003}}$	$\mathbf{1.99e^{-003}}$	10	1000
LINKAGE										
BN_69	$5.28e^{-054}$	1997	$1.87e^{-089}$	$\mathbf{1.58e^{-0121}}$	$5.60e^{-054}$	$\mathbf{2.37e^{-043}}$	0	$1.44e^{-078}$	55	1000
BN_70	$2.00e^{-071}$	5340	$1.03e^{-118}$	$1.02e^{-0243}$	$4.29e^{-052}$	$\mathbf{4.31e^{-011}}$	$\mathbf{8.86e^{-172}}$	$8.86e^{-172}$	45	1000
BN_71	$5.12e^{-111}$	4213	$1.81e^{-210}$	$6.09e^{-0266}$	$\mathbf{3.91e^{-112}}$	$\mathbf{4.32e^{-083}}$	$8.92e^{-195}$	$8.92e^{-195}$	85	1000
BN_72	$4.21e^{-150}$	5000	$7.93e^{-182}$	$2.52e^{-0291}$	–	–	$\mathbf{5.29e^{-246}}$	$3.14e^{-245}$	35	77
BN_73	$2.26e^{-113}$	5279	$8.65e^{-145}$	$3.56e^{-0296}$	$8.67e^{-082}$	$\mathbf{2.24e^{-036}}$	$\mathbf{4.94e^{-219}}$	$1.24e^{-218}$	70	1000
BN_74	$3.75e^{-045}$	1486	$5.58e^{-145}$	$\mathbf{4.98e^{-0185}}$	$6.89e^{-011}$	1	0	$1.05e^{-110}$	70	1000
BN_75	$5.88e^{-091}$	3516	$1.37e^{-179}$	$2.89e^{-0238}$	$\mathbf{8.54e^{-093}}$	$\mathbf{2.99e^{-063}}$	$3.18e^{-172}$	$3.18e^{-172}$	55	1000
BN_76	$4.93e^{-110}$	4994	$2.71e^{-106}$	$8.86e^{-0321}$	$\mathbf{6.65e^{-117}}$	$\mathbf{2.83e^{-056}}$	$1.62e^{-247}$	$4.13e^{-247}$	65	550
BN_77	$6.88e^{-079}$	3668	$9.31e^{-138}$	$\mathbf{6.16e^{-0177}}$	$1.15e^{-067}$	$\mathbf{1.28e^{-046}}$	0	$9.15e^{-127}$	60	1000
D-QMR										
BN_78	$1.83e^{-003}$	28	$2.02e^{-004}$	$8.19e^{-0006}$	$1.91e^{-003}$	$2.80e^{-003}$	$\mathbf{1.76e^{-003}}$	$\mathbf{2.37e^{-003}}$	70	1000
BN_80	$1.31e^{-003}$	680	$1.36e^{-003}$	$\mathbf{8.80e^{-0107}}$	–	–	0	$2.34e^{-047}$	45	1000
BN_82	$5.57e^{-007}$	1062	$4.14e^{-007}$	$3.58e^{-0161}$	–	–	$\mathbf{5.86e^{-093}}$	$2.02e^{-092}$	70	1000
BN_84	$1.81e^{-001}$	911	$2.30e^{-003}$	$\mathbf{7.86e^{-0101}}$	–	–	0	$4.41e^{-043}$	45	1000
BN_86	$4.11e^{-001}$	4962	$5.92e^{-010}$	$\mathbf{4.55e^{-0236}}$	–	–	0	$2.33e^{-128}$	40	65
BN_88	$7.61e^{-001}$	5048	$1.25e^{-003}$	$\mathbf{6.23e^{-0259}}$	–	–	0	$3.94e^{-150}$	45	110
BN_92	$8.06e^{-001}$	5230	$9.55e^{-008}$	$\mathbf{1.25e^{-0234}}$	–	–	0	$3.20e^{-155}$	45	85
RANDOM										
BN_94	$4.49e^{-011}$	595	$1.02e^{-013}$	$2.22e^{-0020}$	–	–	$\mathbf{1.57e^{-011}}$	$\mathbf{5.53e^{-011}}$	85	1000
BN_96	$2.30e^{-009}$	371	$3.86e^{-012}$	$4.73e^{-0023}$	–	–	$\mathbf{8.61e^{-011}}$	$\mathbf{7.94e^{-009}}$	90	1000
BN_98	$2.14e^{-011}$	577	$3.13e^{-014}$	$5.06e^{-0019}$	–	–	$\mathbf{7.40e^{-012}}$	$\mathbf{4.61e^{-011}}$	85	1000
BN_100	$1.89e^{-014}$	759	$1.36e^{-017}$	$2.22e^{-0020}$	–	–	$\mathbf{6.59e^{-015}}$	$\mathbf{2.40e^{-014}}$	70	1000
BN_102	$1.96e^{-026}$	4975	$2.59e^{-030}$	$9.34e^{-0038}$	$1.98e^{-025}$	$3.31e^{-024}$	$\mathbf{4.09e^{-027}}$	$\mathbf{5.04e^{-026}}$	70	830

Table 3. Probability of evidence estimates for the UAI'06 competition Bayes nets. All methods were run for the same amount of time. '–' in the Cox columns indicates that the Wilk-Shapiro test for normality failed. The best correct lower and upper bounds are shown in bold.

which would otherwise increase the variance of the sample data. Second, it employs #SAT solvers to "complete" the samples. That is, our technique samples only a subset of the variables, and uses a #SAT solver to sum out the remaining variables. This technique, known as distributional sampling, generally requires a technique for summing out the unset variables in the sample. Our approach shows that a #SAT solver can be an effective tool for this purpose. In addition to our sampling technique we have proposed the use of the Central Limit Theorem for providing confidence intervals on the estimates. These intervals proved to be quite useful in our experiments with weighted model counting.

Although the unweighted model counting case is simply a special case of the weighted case, out technique currently seems to be more applicable to weighted model counting, exhibiting promising performance on the important problem of computing the probability of evidence (from this conditional probabilities can be computed).

Much work remains to improve the approach. First our current technique for choosing the prefix variables is quite primitive, techniques that ensure that the variables are "balanced," like those employed in SampleCount [14] should help our performance on the unweighted case and perhaps also on the weighted case. In some problem instances it is hard to test the backtrack free distribution, and various possibilities exist for addressing this problem, including using local search to test satisfiability rather than Minisat, and more efficient reuse of previous SAT tests, e.g., [10]. In addition, work can be done on improving the confidence intervals. In particular, a deeper understanding of when these different error estimates are applicable is an important open problem that needs to be addressed.

References

1. F. Bacchus, S. Dalmao, and T. Pitassi. Algorithms and complexity results for# SAT and Bayesian inference. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 340–351, 2003.
2. SampleCount Unweighted Benchmarks. http://www.cs.cornell.edu/ sabhar/software/benchmarks/ijcai07-suite.tgz.
3. UAI 2006 P(e) Benchmarks. http://ssli.ee.washington.edu/~bilmes/uai06inferenceevaluation/uai06-repository/pe/.
4. Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.
5. A. Darwiche. A Compiler for Deterministic Decomposable Negation Normal Form. In *Proceedings of the AAAI National Conference (AAAI)*, pages 627–634, 2002.
6. N. Eén and N. Sörensson. An Extensible SAT-solver. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2003.
7. W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley Eastern Private Ltd., 3rd edition, 1968.
8. J. Geweke. Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica*, 57(6):1317–1339, Nov. 1989.
9. V. Gogate, B. Bidyuk, and R. Dechter. Studies in Lower Bounding Probability of Evidence using the Markov Inequality. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

10. V. Gogate and R. Dechter. Approximate Counting by Sampling the Backtrack-free Search Space. In *Proceedings of the AAAI National Conference (AAAI)*, 2007.
11. V. Gogate and R. Dechter. SampleSearch: A Scheme that Searches for Consistent Samples. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2007.
12. V. Gogate and R. Dechter. AND/OR Importance Sampling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.
13. V. Gogate and R. Dechter. Studies in Solution Sampling. In *Proceedings of the AAAI National Conference (AAAI)*, 2008.
14. C. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman. From Sampling to Model Counting. In *Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI)*, 2007.
15. C. Gomes, A. Sabharwal, and B. Selman. Model Counting: A New Strategy for Obtaining Good Bounds. In *Proceedings of the AAAI National Conference (AAAI)*, 2006.
16. M.R. Jerrum, L.G. Valiant, and V.V.Vazirani. Random Generation of Combinatorial Structures from a Uniform Distribution. *Theoretical Computer Science*, 43:169–188, 1986.
17. L. Kroc, A. Sabharwal, and B. Selman. Leveraging Belief Propagation, Backtrack Search, and Statistics for Model Counting. In *Proceedings of the International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR)*, 2008.
18. R.M. Neal. Annealed Importance Sampling. *Statistics and Computing*, 11:125–139, 2001.
19. N. Nishimura, P. Ragde, and S. Szeider. Solving #SAT Using Vertex Covers. *Acta Informatica*, 44:509–523, 2007.
20. P. Royston. Remark AS R94: A Remark on Algorithm AS 181: The W-test for Normality. *Applied Statistics*, 44(4):547–551, 1995.
21. T. Sang, F. Bacchus, P. Beame, H. Kautz, and T. Pitassi. Combining Component Caching and Clause Learning for Effective Model Counting. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2004.
22. T. Sang, P. Beame, and H. Kautz. Solving Bayesian Networks by Weighted Model Counting. In *Proceedings of the AAAI National Conference (AAAI)*, 2005.
23. H. C. Thode. *Testing for Normality*. Marcel Dekker, Inc., 2002.
24. W. Wei and B. Selman. Accelerating Random Walks. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, 2002.
25. W. Wei and B. Selman. A New Approach to Model Counting. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2005.
26. Cachet wmc 2.0 Website. http://www.cs.washington.edu/homes/sang/cachet/.
27. X.-H. Zhou and S. Gao. Confidence Intervals for the Log-Normal Mean. *Statistics in Medicine*, 16:783–790, 1997.