Fast and Flexible Probabilistic Model Counting

Dimitris Achlioptas[1,2], Zayd Hammoudeh[1], and Panos Theodoropoulos[2(✉)]

[1] Department of Computer Science, University of California,
Santa Cruz, Santa Cruz, CA, USA
{dimitris,zayd}@ucsc.edu
[2] Department of Informatics and Telecommunications,
University of Athens, Athens, Greece
ptheodor@di.uoa.gr

Abstract. We present a probabilistic model counter that can trade off running time with approximation accuracy. As in several previous works, the number of models of a formula is estimated by adding random parity constraints (equations). One key difference with prior works is that the systems of parity equations used correspond to the parity check matrices of Low Density Parity Check (LDPC) error-correcting codes. As a result, the equations tend to be much shorter, often containing fewer than 10 variables each, making the search for models that also satisfy the parity constraints far more tractable. The price paid for computational tractability is that the statistical properties of the basic estimator are not as good as when longer constraints are used. We show how one can deal with this issue and derive rigorous approximation guarantees by performing more solver invocations.

1 Introduction

Given a CNF formula F with n variables, let $S = S(F)$ denote the set of its satisfying assignments (models). One way to estimate $|S|$ is to proceed as follows. For a fixed integer $0 \leq i \leq n$, let $R_i \subseteq \{0,1\}^n$ be a random set such that $\Pr[\sigma \in R_i] = 2^{-i}$ for all $\sigma \in \{0,1\}^n$. Markov's inequality implies that if $|S| < 2^{i-1}$, then $\Pr[S \cap R_i \neq \emptyset] < 1/2$. Therefore, if we select independent random sets $R_i^1, R_i^2, \ldots, R_i^t$ and find that the intersection with S is non-empty for the majority of them, we can declare that $|S| \geq 2^{i-1}$ with confidence $1 - \exp(-\Theta(t))$.

What happens if in the majority of the trials we find the intersection to be empty? Can we similarly draw the conclusion that $|S|$ is unlikely to be much more than 2^i? Unfortunately, no. The informativeness of $S \cap R_i = \emptyset$ depends on significantly more refined statistical properties of the random set R_i than the property that $\Pr[\sigma \in R_i] = 2^{-i}$, i.e., uniformity. For example, imagine that $|S| = 2^i$ and that the distribution of R_i is uniform but such that either

$S \cap R_i = \emptyset$ or $S \cap R_i = S$, always. Then, the number of trials needed to have a reasonable chance of ever witnessing $S \cap R_i \neq \emptyset$ is $\Omega(2^i)$. In other words, with this distribution for R_i, we can not distinguish between an unsatisfiable formula and one with 2^i models.

In the above example, the distribution of the random set R_i is such that the random variable $X = |S \cap R_i|$ exhibits extreme variance, a so-called "lottery phenomenon": it typically equals 0, but with very small probability it is huge. (Nearly) at the other end of the spectrum are distributions for the set R_i that exhibit *pairwise independence*, i.e.,

$$\Pr[\sigma \in R_i \wedge \tau \in R_i] = \Pr[\sigma \in R_i] \cdot \Pr[\tau \in R_i] \quad \text{for every} \quad \sigma \neq \tau \in \{0,1\}^n. \quad (1)$$

To get a feel for (1), fix any $\sigma \in \{0,1\}^n$ and sample R_i. Observe that conditional on $\sigma \in R_i$, the probability that $\tau \in R_i$ must be the same whether τ is at Hamming distance 1 from σ, or at distance, say, $n/2$ (throughout, distance will mean Hamming distance). In other words, the characteristic function of the set R_i must decorrelate in a *single step*!

It is possible to show that Eq. (1) implies that $\Pr[S \cap R_i \neq \emptyset] \geq (\mathbb{E}X)/(1 + \mathbb{E}X)$ and, thus, that if $|S| > 2^i$, then $\Pr[S \cap R_i \neq \emptyset] > 1/2$. Therefore, if, as before, we repeat the experiment t times and find the intersection to be empty in the majority of the trials, now we can declare that $|S| \leq 2^{i+1}$ with confidence $1 - \exp(-\Theta(t))$. Combined with the lower bound argument for $|S|$ outlined earlier, we see that in order to efficiently approximate $|S|$ within a factor of 4 it suffices to have a distribution of sets R_i for which (1) holds and for which checking whether $S \cap R_i = \emptyset$ or not can be done efficiently. Indeed, given such a distribution one can estimate $|S|$ within a $(1 \pm \epsilon)$ factor, for any $\epsilon > 0$, and any desired confidence $1 - \delta$, in $O(\epsilon^{-2} \log(1/\delta))$ trials.

In order to be able to check efficiently whether $S \cap R_i = \emptyset$ we must, at a minimum, be able to represent the random sets R_i compactly, in spite of their exponential size. The key to this is to represent each set R_i *implicitly* as the set of solutions to a system of i random parity (XOR) constraints (linear equations modulo 2). More precisely, for any fixed matrix $A \in \{0,1\}^{i \times n}$, consider the partition (hashing) of $\{0,1\}^n$ induced by the value of $A\sigma \in \{0,1\}^i$. Let

$$R_i = \{\sigma \in \{0,1\}^n : A\sigma = b\} \quad \text{where } b \in \{0,1\}^i \text{ is uniformly random.} \quad (2)$$

Observe that even though the 2^i parts may have dramatically different sizes, the uniformity in the choice of b in (2) implies that $\Pr[\sigma \in R_i] = 2^{-i}$, for every $\sigma \in \{0,1\}^n$, as desired. At the same time, checking whether $S \cap R_i = \emptyset$ or not can be done by converting the i parity constraints to clauses and using a SAT solver, or, more recently, by using a SAT solver supporting parity constraints, e.g., CryptoMiniSat [14].

From the above discussion we see that the only issue left is how the choice of the matrix A affects the variance of the sizes of the different parts and, thus, the variance of $|S \cap R_i|$. To that end, it is not hard to prove that if A is a uniformly random element of $\{0,1\}^{i \times n}$ (equivalently, if each element A_{ij} is set to 0/1 independently with equal probability), then membership in R_i enjoys pairwise

independence, i.e., (1) holds. As mentioned above, this is essentially perfect from a statistical point of view. Unfortunately, though, under this distribution for A each parity constraint contains $n/2$ variables, on average, and changing any variable in a parity constraint immediately changes its truth value (whereas in clauses that's not the case, typically, motivating the two watched literals heuristic [11]). As a result, the branching factor of the search for satisfying assignments (models) that also satisfy the parity equations gets rapidly out of hand as the number of variables in the formula increases.

All ideas presented so far, including in particular the choice of a uniformly random matrix $A \in \{0,1\}^{i \times n}$, first appeared in the pioneering theoretical works by Sipser [13], Stockmeyer [15], and Valiant and Vazirani [17]. As we discuss in Sect. 2, there has since been a long line of works aiming to make the approach practical. Specifically, the limitations posed by *long* parity constraints, i.e., those of (average) length $n/2$, was already recognized in the very first works in the area [7,8]. Later works [6,18] tried to remedy the problem by considering parity equations where each constraint includes each variable independently with probability $p < 1/2$. While such sparsity helps the solver in finding elements of $S \cap R$, the statistical properties of the resulting random sets deteriorate rapidly as p decreases. Crucially, in *all* these works, different constraints (parity equations) select their set of variables independently of one another.

In [1] we introduced the idea of using random matrices $A \in \{0,1\}^{i \times n}$ with *dependent* entries, by selecting A uniformly from an ensemble of Low Density Parity Check (LDPC) matrices. A simplest such ensemble comprises all matrices where every row (equation) contains the same number $\mathtt{l}$ of ones *and* every column contains the same number $\mathtt{r} \geq 3$ of ones. We gave a first mathematical analysis of the statistical properties of the resulting sets R_i and some experimental evidence that their actual statistical properties are probably much better than what is suggested by the mathematical analysis.

A key idea motivating our work here and in [1] is the realization that to prove mathematically rigorous *lower* bounds, the random sets R_i do *not* need to come with any statistical guarantees (besides the trivial requirement of uniformity). The obligation to use distributions $\mathscr{D}_i$ with statistical guarantees exists only for upper bounds and, crucially, only concerns their behavior over sets of size 2^i or greater. When i/n is not tiny we will see that short parity constraints have provably good statistical behavior.

In this paper we present[1] an approximate model counter, called F2, with rigorous guarantees based on these ideas. F2 has three modes of operation, trading accuracy for computation time. To discuss these modes, let us foreshadow that the statistical demerit of a distribution on matrices $A \in \{0,1\}^{i \times n}$ in our context will be captured by a scalar quantity $B = B(i,n) \geq 1$ that increases as the average constraint length decreases, with $B = 1$ corresponding to pairwise independence (and average constraint length $n/2$).

Given any $\delta > 0$, let $q = \ln(1/\delta)$. Given any $\epsilon \in (0, 1/3]$, with probability at least $1 - \delta$, all of the following will occur, in sequence:

---

[1] F2 source code available at https://github.com/ptheod/F2.git.

1. After $O(q + \log_2 n)$ solver invocations, F2 will return a number $\ell \leq \log_2 |S|$ and B.
2. After $O(qB)$ solver invocations, F2 will return a number $u \geq \log_2 |S|$.
3. After $O(qB^2/\epsilon^4)$ solver invocations, F2 will return a number $Z \in (1 \pm \epsilon)|S|$.

Observe that while the bounds $\ell \leq \log_2 |S| \leq u$ are guaranteed (with probability $1 - \delta$), no a priori bound is given for $u - \ell$. In other words, in principle the algorithm may offer very little information on $\log_2 |S|$ at the end of Step 2. As we will see, in practice, this is not the case and, in fact, we expect that in most practical applications Step 3 will be unnecessary. We give a detailed experimental performance of F2 in Sect. 10. The main takeaway is that F2 dramatically extends the range of formulas for which one can get a rigorous model count approximation.

2   Previous Work

The first work on practical approximate model counting using systems of random parity equations was by Gomes et al. [8]. Exactly along the lines outlined in the introduction, they proved that when $A \in \{0,1\}^{i \times n}$ is uniformly random, i.e., when each entry of A is set to 1 independently with probability $p = 1/2$, one can rigorously approximate $\log_2 |S|$ within an additive constant by repeatedly checking if $S \cap R_i = \emptyset$, for various values of i. They further proved that if each entry of A is set to 1 with probability $p < 1/2$ one get a rigorous lower bound, but one which may be arbitrarily far from the truth. In [7], Gomes et al. showed experimentally that it can be possible to achieve good accuracy (without guarantees) using parity constraints of length $k \ll n/2$.

Interest in the subject was rekindled by works of Chakraborty et al. [3] and of Ermon et al. [5]. Specifically, a complete, rigorous, approximate model counter, called ApproxMC, was given in [3] which takes as input any $\delta, \epsilon > 0$, and with probability at least $1 - \delta$ returns a number in the range $(1 \pm \epsilon)|S|$. In [5] an algorithm, called WISH, is given with a similar (δ, ϵ)-guarantee for the more general problem of approximating sums of the form $\sum_{\sigma \in \{0,1\}^n} w(\sigma)$, where w is a non-negative real-valued function over Ω^n, where Ω is a finite domain. Both ApproxMC and WISH also use uniformly random $A \in \{0,1\}^{i \times n}$, so that the resulting parity equations have average length $n/2$, limiting the range of problems they can handle.

ApproxMC uses the satisfiability solver CryptoMiniSat (CMS) [14] which has native support and sophisticated reasoning for parity constraints. CMS can, moreover, take as input a cutoff value $z \geq 1$, so that it will run until it either finds z solutions or determines the number of solutions to be less than z. ApproxMC makes use of this capability in order to target i such that $|S \cap R_i| = \Theta(\delta^{-2})$, instead of i such that $|S \cap R_i| \approx 1$. Our algorithms make similar use of this capability, using several different cutoffs.

The first effort to develop rigorous performance guarantees when $p < 1/2$ was made by Ermon et al. in [6], where an explicit expression was given for the smallest allowed p as a function of $|S|, n, \delta, \epsilon$. The analysis in [6] was

recently improved by Zhao et al. in [18] who, among other results, showed that when $\log_2 |S| = \Omega(n)$, one can get rigorous approximation guarantees with $p = O((\log n)/n)$, i.e., average constraint length $O(\log n)$. While, prima facie, this seems a very promising result, we will see that the dependence on the constants involved in the asymptotics is very important in practice. For example, in our experiments we observe that already setting $p = 1/8$ yields results whose accuracy is much worse than those achieved by LDPC constraints.

Finally, in [4] Chakraborty et al. introduced a very nice idea for reducing the number of solver invocations without any compromise in approximation quality. It amounts to using *nested* sequences of random sets $R_1 \supseteq R_2 \supseteq R_3 \supseteq \cdots \supseteq R_n$ in the search for $i \approx \log_2 |S|$. The key insight is that using nested (instead of independent) random sets R_i means that $|S \cap R_i|$ is deterministically non-increasing in i, so that linear search for i can be replaced with binary search, reducing the number of solver invocations from linear to logarithmic in n. We use the same idea in our work.

2.1   Independent Support Sets

A powerful idea for mitigating the severe limitations arising from long parity constraints was proposed by Chakraborty et al. in [2]. It is motivated by the observation that formulas arising in practice often have a small set of variables $I \subseteq V$ such that every value-assignment to the variables in I has at most one extension to a satisfying assignment. Such a set I is called an *independent support* set. Clearly, if $S' \subseteq \{0,1\}^I$ comprises the value assignments to the variables in I that can be extended to satisfying assignments, then $|S| = |S'|$. Thus, given I, we can rethink of model counting as the task of estimating the size of a subset of $\{0,1\}^I$, completely oblivious to the variables in $V - I$. In particular, we can add random parity constraints only over the variables in I, so that even if we use long constraints each constraint has $|I|/2$ instead of $|V|/2$ variables on average. Since independent support sets of small size can often be found in practice [9], this has allowed ApproxMC to scale to certain formulas with thousands of variables.

In our work, independent support sets are also very helpful, but per a rather "dual" reasoning: for any fixed integers i, k, the statistical quality of random sets defined by systems of i parity constraints with k variables each, decreases with the number of variables over which the constraints are taken. Thus, by adding our short constraints over only the variables in an independent support set, we get meaningful results on formulas for which $|I|/2$ is too large (causing CMS and thus ApproxMC to timeout), but for which $|I|/|V|$ is sufficiently large for our short parity constraints to have good statistical properties.

Variable Convention. *In the rest of the paper we will think of the set of variables V of the formula F being considered as being some independent support set of F (potentially the trivial one, corresponding to the set of all variables). Correspondingly, n will refer to the number of variables in that set V.*

3   Our Results

In [1], the first and last authors showed that systems of parity equations based on LDPC codes can be used both to derive a rigorous lower bound for $|S|$ quickly, and to derive a (δ, ϵ)-approximation of $|S|$ with $O(qB^2/\epsilon^4)$ solver invocations, as per Step 3 of F2. The new contributions in this work are the following.

– In Sect. 5 we show how to compute a rigorous *upper* bound for $|S|$ with a number of solver invocations that is *linear* in B. While the bound does not come with any guarantee of being close to $|S|$, in practice it is remarkably accurate. Key to our approach is a large deviations inequality bounding the lower tail of a random variable as a function of the ratio between its second moment and the square of its first moment. Notably, the analogue of this inequality does *not* hold for the upper tail. Recognizing and leveraging this asymmetry is our main intellectual contribution.
– In Sect. 6 we simplify and streamline the analysis of the (δ, ϵ)-approximation algorithm of [1], showing also how to incorporate the idea of nested sampling sets.
– In Sects. 7–9 we refine the analysis of [1] for B, resulting in significantly better bounds for it. Getting such improved bounds is crucial for making our aforementioned upper-bounding algorithm fast in practice (as it is linear in B).
– Finally, we give a publicly available implementation, called F2.

4   First a Lower Bound

To simplify exposition we only discuss lower bounds of the form $|S| \geq 2^i$ for $i \in \mathbb{N}$, deferring the discussion of more precise estimates to Sect. 6. For any distribution $\mathscr{D}$, let $R \sim \mathscr{D}$ denote that random variable R has distribution $\mathscr{D}$.

Definition 1. *Let $\mathscr{D}$ be a distribution on subsets of a set U and let $R \sim \mathscr{D}$. We say that $\mathscr{D}$ is i-uniform if $\Pr[\sigma \in R] = 2^{-i}$ for every $\sigma \in U$.*

Algorithm 1 below follows the scheme presented in the introduction for proving lower bounds, except that instead of asking whether typically $S \cap R \neq \emptyset$, it asks whether typically $|S \cap R| \geq 2$. To do this, $|S \cap R|$ is trimmed to 4 in line 5 (by running CryptoMiniSat with a cutoff of 4), so that the event $Z \geq 2t$ in line 8 can only occur if the intersection had size at least 2 in at least $t/2$ trials.

Theorem 1 ([1])**.** $\Pr[\textit{The output of Algorithm 1 is incorrect}] \leq e^{-t/8}$.

To get a lower bound for $|S|$ we can invoke Algorithm 1 with $i = 1, 2, \ldots, n$ sequentially and keep the best lower bound returned (if any). To accelerate this linear search we can invoke Algorithm 1 with $i = 1, 2, 4, 8, \ldots$ until the first "Don't know" occurs, say at $i = 2^u$. At that point we can perform binary search in $\{2^{u-1}, \ldots, 2^u - 1\}$, treating every "Don't know" answer as a (conservative) imperative to reduce the interval's upper bound to the midpoint and every "Yes"

Algorithm 1. Given i, t decides if $|S| \geq 2^i$ with error probability $e^{-t/8}$

```
1:  Z ← 0
2:  j ← 0
3:  while j < t and Z < 2t do          ▷ The condition Z < 2t is an optimization
4:      Sample R_j ∼ 𝒟_i             ▷ 𝒟_i can be any i-uniform distribution
5:      Y_j ← min{4, |S ∩ R_j|}        ▷ Run CryptoMiniSat with cutoff 4
6:      Z ← Z + Y_j
7:      j ← j + 1
8:  if Z ≥ 2t then
9:      return "Yes"
10: else
11:     return "Don't know"
```

answer as an allowance to increase the interval's lower bound to the midpoint. We call this scheme "doubling binary search." In Step 1 of F2 this is further accelerated by invoking Algorithm 1 with a very small number of trials, t, in the course of the doubling-binary search. The result of the search is treated as a "ballpark" estimate and a proper binary search is done in its vicinity, by using for each candidate i the number of iterations suggested by Theorem 1.

5   Then an Upper Bound

As discussed in the introduction, lottery phenomena may cause Algorithm 1 and, thus, Step 1 of F2 to underestimate $\log_2 |S|$ arbitrarily. To account for the possibility of such phenomena we bound the "lumpiness" of the sets $R_i \sim \mathscr{D}_i$ by the quantity defined in (3) below, measuring lumpiness at a scale of M.

Definition 2. *Let $\mathscr{D}$ be any distribution on subsets of $\{0, 1\}^n$ and let $R \sim \mathscr{D}$. For any fixed $M \geq 1$, let*

$$\text{Boost}(\mathscr{D}, M) = \max_{\substack{S \subseteq \{0,1\}^n \\ |S| \geq M}} \frac{1}{|S|(|S| - 1)} \sum_{\substack{\sigma, \tau \in S \\ \sigma \neq \tau}} \frac{\Pr[\sigma, \tau \in R]}{\Pr[\sigma \in R] \Pr[\tau \in R]}. \qquad (3)$$

To develop intuition for (3) observe that the ratio inside the sum is the factor by which the a priori probability that a truth assignment belongs in R is modified by conditioning on some other truth assignment belonging in R. So, if membership in R is pairwise independent, then $\text{Boost}(\mathscr{D}, \cdot) = 1$. Note also that since $|S| \geq M$ instead of $|S| = M$ in (3), the function $\text{Boost}(\mathscr{D}, \cdot)$ is non-increasing in M. As we will see, the critical quantity for an i-uniform distribution $\mathscr{D}_i$ is $\text{Boost}(\mathscr{D}_i, 2^i)$, i.e., an i-uniform distribution can be useful even if $\text{Boost}(\mathscr{D}_i)$ is huge for sets of size less than 2^i.

To analyze Algorithm 2 we will use the following inequality of Maurer [10].

Algorithm 2. Given $\delta > 0$ and $L \leq |S|$ returns $Z \geq |S|$ with probability $1 - \delta$

1: $\ell \leftarrow \lfloor \log_2 L \rfloor$
2: $\mathscr{D}_\ell \leftarrow$ any ℓ-uniform distribution
3: $B \leftarrow$ any upper bound for $\mathrm{Boost}(\mathscr{D}_\ell, 2^\ell)$
4: $t \leftarrow \lceil 8(B+1)\ln(1/\delta) \rceil$
5: $Z \leftarrow 0$
6: **for** j from 1 to t **do**
7:     Sample $R_j \sim \mathscr{D}_\ell$
8:     $X_j \leftarrow |S \cap R_j|$ $\triangleright$ Run CryptoMiniSat without cutoff
9:     $Z \leftarrow Z + X_j$
10: **return** "$|S| \leq 2^{\ell+1}(Z/t)$"

Lemma 1. *Let $X_1, \ldots, X_t$ be non-negative i.i.d. random variables. Let $Z = \sum_{i=1}^{t} X_i$. If $\mathbb{E}X_1^2/(\mathbb{E}X_1)^2 \leq B$, then for any $\alpha \geq 0$,*

$$\Pr[Z \leq (1-\alpha)\mathbb{E}Z] \leq \exp\left(-\frac{\alpha^2 t}{2B}\right).$$

Theorem 2. $\Pr[$*The output of Algorithm 2 is correct*$] \geq 1 - \delta$.

Proof. Let Z be the random variable equal to the value of variable Z in line 9, right before line 10 is executed. If $Z = z$, in order for the output to be wrong it must be that $|S| > 2^{\ell+1}(z/t)$, implying $\mathbb{E}Z = t|S|2^{-\ell} > 2z$ and, therefore, that the event $Z \leq \mathbb{E}Z/2$ occurred. Since Z is the sum of i.i.d. non-negative random variables $X_1, \ldots, X_t$, we can bound $\Pr[Z \leq \mathbb{E}Z/2]$ via Lemma 1.

To bound $\mathbb{E}X_1^2/(\mathbb{E}X_1)^2$, we write $X_1 = \sum_{\sigma \in S} \mathbf{1}_{\sigma \in R_1}$ and observe that

$$\mathbb{E}X_1^2 = \sum_{\sigma, \tau \in S} \Pr[\sigma, \tau \in R_1]$$

$$= \sum_{\sigma \in S} \Pr[\sigma \in R_1] + \sum_{\substack{\sigma, \tau \in S \\ \sigma \neq \tau}} \Pr[\sigma, \tau \in R_1]$$

$$\leq \sum_{\sigma \in S} \Pr[\sigma \in R_1] + 2^{-2i}|S|(|S|-1)\mathrm{Boost}(\mathscr{D}, |S|)$$

$$\leq \mathbb{E}X_1 + \mathrm{Boost}(\mathscr{D}, |S|)(\mathbb{E}X_1)^2.$$

Since $|S| \geq L \geq 2^\ell$ and $\mathrm{Boost}(\mathscr{D}_\ell, M)$ is non-increasing in M, we see that

$$\frac{\mathbb{E}X_1^2}{(\mathbb{E}X_1)^2} \leq \frac{1}{\mathbb{E}X} + \mathrm{Boost}(\mathscr{D}, |S|) \leq 1 + \mathrm{Boost}(\mathscr{D}_\ell, 2^\ell). \qquad (4)$$

Therefore, applying Lemma 1 with $\alpha = 1/2$ and recalling the definitions of B and t in lines 3 and 4 of Algorithm 2, we see that $\Pr[Z \leq \mathbb{E}Z/2] \leq \delta$, as desired.

F2. Given $L \leq |S| \leq U$, $\delta, \theta > 0$ returns $Z \in (1 \pm \delta)|S|$ with probability $1 - \theta$

1: **if** $L < 4/\delta$ **then**
2:    $E \leftarrow$ number of solutions found by CryptoMiniSat ran with cutoff $4/\delta$
3:    **if** $E < 4/\delta$ **then return** E ▷ In this case $|S| = E$

4:
5: $\ell \leftarrow \lfloor \log_2(\delta L/4) \rfloor$
6: $u \leftarrow \lceil \log_2 U \rceil$
7: $B \leftarrow$ Any upper bound for $\max_{\ell \leq i \leq u-2} \mathrm{Boost}(\mathscr{D}_i, 2^i)$

8:
9: $\delta \leftarrow \min\{\delta, 1/3\}$
10: $\xi \leftarrow 8/\delta$
11: $b \leftarrow \lceil \xi + 2(\xi + \xi^2(B-1)) \rceil$ ▷ If $B = 1$, then $b = \lceil 24/\delta \rceil$
12: $t \leftarrow \lceil (2b^2/9) \ln(5/\theta) \rceil$
13:
14: $Z_\ell, Z_{\ell+1}, \ldots, Z_u \leftarrow 0$
15:
16: **for** j from 1 to t **do**
17:    $M \leftarrow$ a uniformly random element of an LDPC ensemble over $\{0,1\}^{u \times n}$
18:    $y \leftarrow$ a uniformly random element of $\{0,1\}^u$
19:    **for** i from ℓ to u **do**
20:       Let M_i, y_i comprise the first i rows of M and y, respectively
21:       $R_{i,j} \leftarrow \{\sigma \in \{0,1\}^n : M_i\sigma = y_i\}$ ▷ Enforce the first i parity constraints
22:       $Y_{i,j} \leftarrow \min\{b, |S \cap R_{i,j}|\}$ ▷ Run CryptoMiniSat with cutoff b
23:       $Z_i \leftarrow Z_i + Y_{i,j}$

24:
25: $j \leftarrow \max\{-1, \max\{\ell \leq i \leq u : Z_i \geq t(1-\delta)(4/\delta)\}\}$
26:
27: **if** $j \neq -1$ **then return** $2^j(Z_j/t)$
28: **else return** "Fail"

---

6   Finally a $(1 \pm \delta)|S|$ Approximation

Given any bounds $L \leq |S| \leq U$, for example derived by using Algorithms 1 and 2, algorithm F2 below yields a rigorous approximation of $|S|$ within $1 \pm \delta$ with a number of solver invocations proportional to B^2/δ^4, where

$$B = \max_{\ell \leq i \leq u-2} \mathrm{Boost}(\mathscr{D}_i, 2^i),$$

where $\ell \approx \log_2(\delta L)$ and $u \approx \log_2 u$. (If $B = 1$, the iterations drop to $O(\delta^{-2})$.)

Theorem 3. $\Pr[F2 \text{ returns } Z \in (1 \pm \delta)|S|] \geq 1 - \theta$.

To prove Theorem 3 we will need the following tools.

Lemma 2 (Hoeffding's Inequality). *If $Z = Y_1 + \cdots + Y_t$, where $0 \leq Y_i \leq b$ are independent random variables, then for any $w \geq 0$,*

$$\Pr[Z/t \geq \mathbb{E}Z/t + w] \leq e^{-2t(w/b)^2} \quad and \quad \Pr[Z/t \leq \mathbb{E}Z/t - w] \leq e^{-2t(w/b)^2}. \tag{5}$$

Lemma 3 ([1]). *Let $X \geq 0$ be an arbitrary integer-valued random variable. Write $\mathbb{E}X = \mu$ and $\mathrm{Var}(X) = \sigma^2$. For some integer $b \geq 0$, define the random variable $Y = \min\{X, b\}$. For any $\lambda > 0$, if $b \geq \mu + \lambda\sigma^2$, then $\mathbb{E}Y \geq \mathbb{E}X - 1/\lambda$.*

Lemma 4 ([1]). *Let $\mathscr{D}$ be any i-uniform distribution on subsets of $\{0,1\}^n$. For any fixed set $S \subseteq \{0,1\}^n$, if $R \sim \mathscr{D}$ and $X = |S \cap R|$, then $\mathrm{Var}(X) \leq \mathbb{E}X + (\mathrm{Boost}(\mathscr{D}, |S|) - 1)(\mathbb{E}X)^2$.*

Proof. If $|S| < 4/\delta$, the algorithm returns exactly $|S|$ and exits. Otherwise, the value ℓ defined in line 5 is non-negative and $q := \lfloor \log_2(\delta|S|/4) \rfloor \geq \ell$ since $L \leq |S|$.

Let $A_i = Z_i/t$. We will establish the following propositions:

(a) $\Pr[A_q 2^q \notin (1 \pm \delta)|S|] \leq 2e^{-9t/(2b^2)}$.
(b) $\Pr[A_{q+1} 2^{q+1} \notin (1 \pm \delta)|S|] \leq 2e^{-9t/(2b^2)}$.
(c) If $A_q 2^q \in (1 \pm \delta)|S|$, then $j \geq q$ in line 25 (deterministically).
(d) $\Pr[j \geq q + 2] \leq e^{-8t/b^2}$.

Given propositions (a)–(d) the theorem follows readily. If $A_{q+k} 2^{q+k}$ is in the range $(1 \pm \delta)|S|$ for $k \in \{0, 1\}$ but for $k \geq 2$ it is less than $(1 - \delta)(4/\delta)$, then the algorithm will report either $A_q 2^q$ or $A_{q+1} 2^{q+1}$, both of which are in $(1 \pm \delta)|S|$. Thus, the probability that the algorithm does not report a number in $(1 \pm \delta)|S|$ is at most $2 \cdot 2e^{-9t/(2b^2)} + e^{-8t/b^2}$ which, by our choice of t, is less than θ.

To establish propositions (a)–(d) we start by noting the following facts:

(i) $R_{i,j}$ is sampled from an i-uniform distribution for every i, j.
(ii) The sets $R_{i,1}, \ldots, R_{i,t}$ are independent for every i.
(iii) $R_{\ell,j} \supseteq R_{\ell+1,j} \supseteq \cdots \supseteq R_{u-1,j} \supseteq R_{u,j}$ for every j.

Now, fix any $i = q + k$, where $k \geq 0$. Let $X_{i,j} = |S \cap R_{i,j}|$ and write $\mathbb{E}X_{i,j} = \mu_i$, $\mathrm{Var}(X_{i,j}) = \sigma_i^2$. By fact (ii), Z_i is the sum of t independent random variables $0 \leq Y_{i,j} \leq b$. Since $\mathbb{E}Z_i/t \leq \mu_i$, Hoeffding's inequality implies that for all $i \geq q$,

$$\Pr[Z_i/t \geq (1 + \delta)\mu_i] \leq \exp\left(-2t\left(\frac{\delta\mu_i}{b}\right)^2\right). \tag{6}$$

To bound $\Pr[Z_i/t \geq (1 - \delta)\mu_i]$ for $k \in \{0, 1\}$ we first observe that $|S| \geq 2^{q+1}$, since $\delta \leq 2$. Since $\mathrm{Boost}(\mathscr{D}, M)$ is non-increasing in M and $q \leq u - 2$ we see that

$$\max_{k \in \{0,1\}} \mathrm{Boost}(\mathscr{D}_{q+k}, |S|) \leq \max\{\mathrm{Boost}(\mathscr{D}_q, 2^{q+1}), \mathrm{Boost}(\mathscr{D}_{q+1}, 2^{q+1})\}$$

$$\leq \max\{\mathrm{Boost}(\mathscr{D}_q, 2^q), \mathrm{Boost}(\mathscr{D}_{q+1}, 2^{q+1})\}$$

$$\leq \max_{\ell \leq i \leq u-2} \mathrm{Boost}(\mathscr{D}_i, 2^i)$$

$$\leq B. \tag{7}$$

Fact (i) implies that X_{ij} satisfies the conditions of Lemma 4. Therefore, for $i \in \{q, q+1\}$, Lemma 4 combined with (7) implies $\sigma_i^2 \leq \mu_i + (B - 1)\mu_i^2$. Since

$\mu_i < 8/\delta$ for all $i \geq q$ while $\xi = 8/\delta$, we see that $b = \lceil \xi + 2(\xi + \xi^2(B-1)) \rceil \geq \mu_i + 2\sigma_i^2$. Thus, for $i \in \{q, q+1\}$ the random variables $X_{i,j}, Y_{i,j}$ satisfy the conditions of Lemma 3 with $\lambda = 2$, implying $\mathbb{E}Y_{i,j} \geq \mathbb{E}X_{i,j} - 1/2$. Therefore, $\mathbb{E}Z_i/t \geq \mu_i - 1/2$ for $i \in \{q, q+1\}$ so that Hoeffding's inequality implies

$$\Pr[Z_i/t \leq (1-\delta)\mu_i] \leq \exp\left(-2t\left(\frac{\delta\mu_i - 1/2}{b}\right)^2\right). \tag{8}$$

To establish propositions (a) and (d) observe that $\mu_{q+k} \geq 2^{2-k}/\delta$ by Fact (i). Therefore, (6) and (8) imply that for $k \in \{0, 1\}$, the probability that $A_{q+k}2^{q+k}$ is outside $(1 \pm \delta)|S|$ is at most

$$2\exp\left(-2t\left(\frac{2^{2-k} - 1/2}{b}\right)^2\right) < 2\exp(-9t/(2b^2)) .$$

To establish proposition (c) note that if $A_q \geq (1-\delta)\mu_q$, then $A_q \geq (1-\delta)(4/\delta)$ and, thus, $j \geq q$. Finally, to establish proposition (d) observe that, by Fact (iii), the random variables Z_i are non-increasing in i, so that $j \geq q + 2$ implies $A_{q+2}2^{q+2} < (1-\delta)(4/\delta)$. To bound the probability of this event we note that $\mu_{q+2} < 2/\delta$. Thus, $\mu_{q+2} + w \geq (1-\delta)(4/\delta)$, implies $w > 2(1-2\delta)/\delta$, which, since $\delta \leq 1/3$, implies $w > 2$. Therefore, (5) implies $\Pr[j \geq q+2] \leq e^{-8t/b^2}$.

7   Homogeneous Distributions

Our goal in Sects. 7–9 is to derive an upper bound for B when the random matrix A corresponds to the parity check matrix of an LDPC code. To that end, in this section we derive an expression for B valid for any random set distribution that satisfies certain symmetry properties. In Sect. 8 we relate the sets R_i corresponding to codewords of LDPC codes to these properties. Finally, in Sect. 9 we discuss how to deal with miscellaneous technical issues arising from the need to be able to work with formulas with an arbitrary number of variables and clauses, while retaining mathematical rigor in our bounding of B.

The analysis in this section is identical to the one in [1] except for requiring that $f(n) = 0$ in the definition of tractability. This has the effect of changing the lower index of summation in the definition of B in Theorem 4 from 0 to 1 which, in turn, makes a significant difference in practice.

Definition 3. *An i-uniform distribution, $\mathscr{D}_i$ is* homogeneous *if there exists a function f, called the* density *of $\mathscr{D}_i$, such that for all $\sigma, \tau \in \{0,1\}^n$, if $R \sim \mathscr{D}_i$, then $\Pr[\tau \in R \mid \sigma \in R] = f(\mathrm{Hamming}(\sigma, \tau))$.*

Definition 4. *A homogenous distribution is* tractable *if its density f satisfies: $f(j) \geq f(j+1)$ for $j < n/2$, $f(j) \leq f(n-j)$ for $j \geq n/2$, and $f(n) = 0$.*

For any $S \subset \{0,1\}^n$ and $\sigma \in S$, let $H_\sigma^S(d)$ denote the number of elements of S at Hamming distance d from σ. In [1] it was shown that for any homogenous distribution $\mathscr{D}_i$, and any $M \geq 1$,

$$\text{Boost}(\mathscr{D}_i, M) \leq \max_{\substack{S \subseteq \{0,1\}^n \\ |S| \geq M \\ \sigma \in S}} \frac{2^i}{|S|-1} \sum_{d=1}^n H_\sigma^S(d) f(d). \tag{9}$$

To bound (9), we assume that $|S| \geq 2n + 1$ so that there exists $2 \leq z \leq n/2$ such that $(|S|-1)/2 = \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{z-1} + \alpha\binom{n}{z}$, for some $\alpha \in [0,1)$. (If $|S| < 2n+1$, then we can estimate $|S$ by using a handful of long parity constraints.) Fact $f(j) \leq f(n-j)$ for $j \geq n/2$ implies (10). Facts $f(j) \geq f(j+1)$ for $j < n/2$ and $f(n) = 0$ imply (11). Finally, the fact $f(z-1) \geq f(z)$ implies (13).

$$\frac{\sum_{d=1}^n H_\sigma^S(d) f(d)}{|S|-1} \leq \frac{\sum_{d=1}^{n/2} H_\sigma^S(d) f(d) + \sum_{d>n/2} H_\sigma^S(d) f(n-d)}{|S|-1} \tag{10}$$

$$\leq \frac{2\left(\sum_{d=1}^{z-1} \binom{n}{d} f(d) + \alpha\binom{n}{z} f(z)\right)}{|S|-1} \tag{11}$$

$$= \frac{\sum_{d=1}^{z-1} \binom{n}{d} f(d) + \alpha\binom{n}{z} f(z)}{\sum_{d=1}^{z-1} \binom{n}{d} + \alpha\binom{n}{z}} \tag{12}$$

$$\leq \frac{\sum_{d=1}^{z-1} \binom{n}{d} f(d)}{\sum_{d=1}^{z-1} \binom{n}{d}} \tag{13}$$

$$:= B(z). \tag{14}$$

To bound $B(z)$ observe that since $f(j) \geq f(j+1)$ for $j < n/2$ it follows that $B(j) \geq B(j+1)$ for $j < n/2$. Thus, to bound $B(z)$ from above it suffices to bound z from below. Let $h : x \mapsto -x \log_2 x - (1-x) \log_2 x$ be the binary entropy function and let $h^{-1} : [0,1] \mapsto [0,1]$ map y to the smallest number x such that $h(x) = y$. It is well-known that $\sum_{d=1}^z \binom{n}{d} \leq 2^{nh(z/n)}$, for every integer $1 \leq z \leq n/2$. Therefore, $z \geq \lceil nh^{-1}(\log_2(|S|/2)/n)\rceil$, which combined with (9) and (14) implies the following.

Theorem 4. *If $\mathscr{D}_i$ is a tractable distribution with density f, then*

$$\text{Boost}(\mathscr{D}_i, M) \leq 2^i B\left(\left\lceil nh^{-1}\left(\frac{\log_2 M - 1}{n}\right)\right\rceil\right), \tag{15}$$

where $B(z) = \sum_{d=1}^{z-1} \binom{n}{d} f(d)/\sum_{d=1}^{z-1} \binom{n}{d}$ and $h^{-1} : [0,1] \mapsto [0,1]$ maps y to the smallest number x such that $h(x) = y$, where h is the binary entropy function.

8   Low Density Parity Check Codes

We will consider the set of all matrices $\{0,1\}^{i \times n}$ where:

(i) Every column (variable) has exactly $\mathtt{l} \geq 3$ non-zero elements.
(ii) Every row (equation) has $\lfloor \mathtt{r} \rfloor$ or $\lceil \mathtt{r} \rceil$ non-zero elements, where $\mathtt{r} = \mathtt{l}n/i$.

Given n, i, and $\mathtt{l}$, let i_0 denote the number of equations with $\lfloor \mathtt{r} \rfloor$ variables and let $i_1 = i - i_0$. Let A be selected uniformly at random[2] among all matrices satisfying (i)–(ii). Let $R = \{\sigma \in \{0,1\}^n : A\sigma = b\}$, where $b \in \{0,1\}^i$ is uniformly random. Lemma 3.157 of [12] implies that for every $\sigma \in \{0,1\}^n$, if $\sigma \in R$, then the expected number of codewords at distance d from σ, denoted by codewords(d), is independent of σ (due to the row- and column-symmetry in the distribution of A) and equals the coefficient of $x^{d\mathtt{l}}$ in the polynomial

$$\binom{n}{d} \frac{\left(\sum_j \binom{r}{2j} x^{2j} \right)^{i_0} \left(\sum_j \binom{r+1}{2j} x^{2j} \right)^{i_1}}{\binom{n\mathtt{l}}{d\mathtt{l}}}.$$

If $\mathscr{D}_i$ denotes the distribution of R, the uniformity in the choice of b implies that $\mathscr{D}_i$ is i-uniform. The fact that for every $\sigma \in \{0,1\}^n$, conditional on $\sigma \in R$, the expected number of codewords at distance d from σ is independent of σ implies that for any fixed $\tau \neq \sigma$, $\Pr[\text{both } \sigma, \tau \in R] = 2^{-i} f(d)$, where $f(d) = $ codewords$(d)/\binom{n}{d}$, making $\mathscr{D}_i$ homogeneous with density f.

Regarding tractability, we begin by noting that if any equation has an odd number of variables, then the complement of a codeword can not be a codeword, implying codewords$(n) = 0$. When $\mathtt{r}$ is an ever integer we achieve $i_1 > 0$ by adding a single dummy Boolean variable to the formula (and reducing all our estimates of $|S|$ by 2). To simplify exposition in the following we assume $i_1 > 0$.

It is also well-known [12] that codewords$(j) \geq $ codewords$(j+1)$ for $j < n/2$, so that we are left to establish $f(j) \geq f(j+1)$ for all $0 \leq j < n/2$. Unfortunately, this is not strictly true for a trivial reason: in the vicinity of $n/2$ the function f is non-monotone, exhibiting minuscule fluctuations (due to finite-scale-effects) around its globally minimum value at $n/2$. While this prevents us from applying Theorem 4 immediately, it is easy to overcome. Specifically, for the proof of Theorem 4 to go through it is enough that $f(j) \geq f(j+1)$ for all $1 \leq j < z$ (instead of all $1 \leq j < n/2$), something which for most sets of interest holds, as $z \ll n/2$. Thus, to provide a rigorous upper bound on Boost, it is enough to verify the monotonicity of f up to z while evaluating $B(z)$.

9   Bounding B in Practice

In defining our systems of parity equations based on LDPC codes in the previous sections, we made sure that every variable participates in an even number of equations, we used equations whose lengths are successive integers, and we insisted on always having at least one equation of odd length. These seemingly

---

[2] This can be done by selecting a uniformly random permutation of size $[\mathtt{l}n]$ and using it to map each of the $\mathtt{l}n$ non-zeros to equations; when $\mathtt{l}, \mathtt{r} \in O(1)$, the variables in each equation will be distinct with probability $\Omega(1)$, so that a handful of trials suffice to generate a matrix as desired.

minor tricks make a very big difference in the bound of Boost in Theorem 4. Unfortunately, the number of iterations, t, needed by our (δ, ϵ)-approximation algorithm of Sect. 6 has a very large leading constant factor, in order to simplify the mathematical analysis. (This is *not* the case for our upper-bounding algorithm of Sect. 5.) For example, if the approximation factor $\delta = 1/3$ and the error probability $\theta = 1/5$, even in the ideal case where $B = 1$, i.e., the case of pairwise independence, $t = 3,709$. In reality, when $B = 1$, a dozen repetitions are more than enough to get an approximation with this δ, θ. Far worse, when $B = 2$, the number of repetitions t explodes to over 1 million, making the derivation of rigorous (δ, ϵ)-approximations via Theorem 4 unrealistic. That said, we believe that further sharpening of Theorem 4 is within grasp.

Luckily, our algorithms for deriving rigorous upper and lower bounds have much better constant-factor behavior. Moreover, as we will see experimentally, the heuristic estimate for $|S|$ that can be surmised from their (ultra-fast) execution appears to be *excellent* in practice. Below we describe a set of experiments we performed showing that one can get rigorous results in realistic times using our tools for formulas that are largely outside the reach of all known other model counters.

10   Experiments

We compare Algorithms 1, 2, i.e., our lower and upper bounding algorithms, with the deterministic, exact model counter `sharpSAT` [16] and the probabilistic, approximate model counter ApproxMC2 (AMC2) [4]. We consider the same 387 formulas as [4] except for 2 unsatisfiable formulas and 10 formulas whose number of solutions (and, thus, equations) is so small that our parity equations devolve into long XOR equations. Of the remaining 375 formulas, `sharpSAT` solves 245 in under 2 s, in every case significantly faster than all other methods. At the other extreme, 40 formulas are not solved by any method within the given time limit of 8 h. We report on the remaining 90, most interesting, formulas. All experiments were run on a modern cluster of 13 nodes, each with 16 cores and 128 GB RAM.

We use an improved implementation of CryptoMiniSat [14] tuned for hashing-based algorithms by Mate Soos and Kuldeep Meel, which is pending publication. This also allows to deal with the fact that 10 of the 90 formulas come with a *sampling set*, i.e., a subset of variables V such that the goal is to count the size of the projection of the set of all models on V. Since `sharpSAT` does not provide such constrained counting functionality, we do not run it on these formulas.

To provide a sense of the tradeoff between the length of the parity constraints and B, we note that when every variable appears in 6 parity constraints, then $B < 30$ for all but 3 formulas, while for all but 1 formula all equations have length at most 16. When every variable appears in 12 parity constraints, then $B < 3$ for all but 3 formulas, while for all but 6 formulas all equations have length at most 28.

Our Algorithms 1, 2 terminated within the allotted time for 87 of the 90 formulas, providing a rigorous lower bound and a rigorous upper bound. By
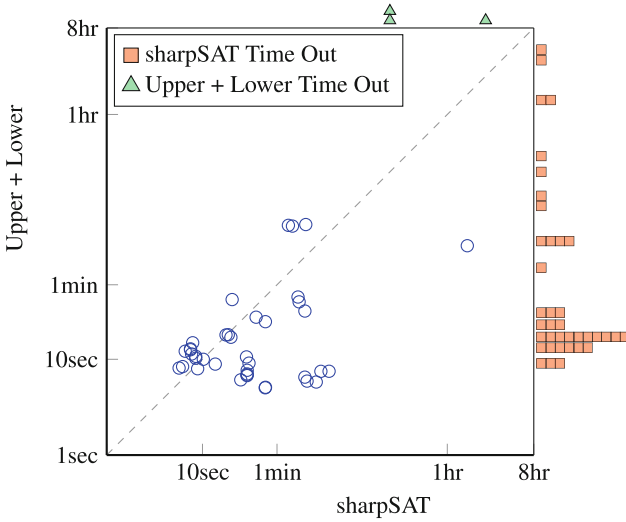
Fig. 1. The sum of the running times of the lower and upper bounding algorithms in F2 vs. the running time of sharpSAT.
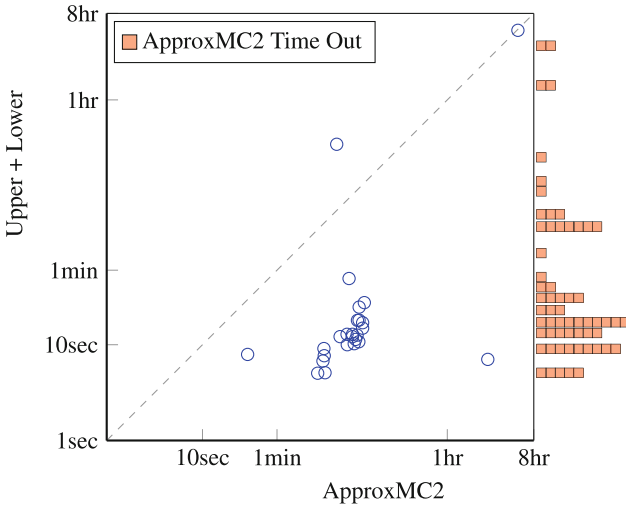
Fig. 2. The sum of the running times of the lower and upper bounding algorithms in F2 vs. the running time of ApproxMC2.

comparison, sharpSAT terminated on 45 formulas (out of $90 - 10 = 80$), while ApproxMC2 on 25 of 90.

For most formulas the ratio between our two rigorous bounds is between 8 and 16 and for none more than 64. For the 48 formulas for which the model count is known, either exactly via sharpSAT or approximately via ApproxMC2,

the ratio between our upper bound and the known count was *typically less than 2 and never more than 3.* This is in spite of the fact that the time to derive it is often just a handful of *seconds* for formulas for which ApproxMC2 and/or `sharpSAT` time out given 8 h.

In Figs. 1 and 2, we plot the sum of the running time of our two algorithms, against the running time of sharpSat and ApproxMC2, respectively. (Marks outside the 8hr × 8hr box, indicate a time-out and only one of their two coordinates is meaningful.)

11   Conclusions

We have shown that by using systems off parity constraints corresponding to LDPC matrices, one can get rigorous lower bounds *and* rigorous upper bounds. While these bounds do not come with a priori guarantees about how close they will be to one another, in practice they are typically within a small multiplicative factor, e.g., 2–3. We believe that for many practical applications such bounds will be quite useful, as they are both rigorous and fast to derive. In particular, when $(\log_2 |S|)/n$ is not too small, the constraint lengths can remain bounded, for arbitrarily large n. As a result, our tool F2 can deliver rigorous results for formulas that appear outside the reach of tools based on long parity equations, such as ApproxMC2.

References

1. Achlioptas, D., Theodoropoulos, P.: Probabilistic model counting with short XORs. In: Gaspers, S., Walsh, T. (eds.) SAT 2017. LNCS, vol. 10491, pp. 3–19. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66263-3_1
2. Chakraborty, S., Fremont, D.J., Meel, K.S., Seshia, S.A., Vardi, M.Y.: Distribution-aware sampling and weighted model counting for SAT. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 27–31 July 2014, Québec City, Canada, pp. 1722–1730. AAAI Press (2014)
3. Chakraborty, S., Meel, K.S., Vardi, M.Y.: A scalable approximate model counter. In: Schulte, C. (ed.) CP 2013. LNCS, vol. 8124, pp. 200–216. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40627-0_18
4. Chakraborty, S., Meel, K.S., Vardi, M.Y.: Algorithmic improvements in approximate counting for probabilistic inference: from linear to logarithmic SAT calls. In: Kambhampati, S. (ed.) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 3569–3576. IJCAI/AAAI Press (2016)
5. Ermon, S., Gomes, C.P., Sabharwal, A., Selman, B.: Taming the curse of dimensionality: discrete integration by hashing and optimization. In: Proceedings of the 30th International Conference on Machine Learning (ICML) (2013)

6. Ermon, S., Gomes, C.P., Sabharwal, A., Selman, B.: Low-density parity constraints for hashing-based discrete integration. In: Proceedings of the 31st International Conference on Machine Learning (ICML), pp. 271–279 (2014)
7. Gomes, C.P., Hoffmann, J., Sabharwal, A., Selman, B.: Short XORs for model counting: from theory to practice. In: Theory and Applications of Satisfiability Testing (SAT), pp. 100–106 (2007)
8. Gomes, C.P., Sabharwal, A., Selman, B.: Model counting: a new strategy for obtaining good bounds. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), pp. 54–61 (2006)
9. Ivrii, A., Malik, S., Meel, K.S., Vardi, M.Y.: On computing minimal independent support and its applications to sampling and counting. Constraints **21**(1), 41–58 (2016)
10. Maurer, A.: A bound on the deviation probability for sums of non-negative random variables. JIPAM J. Inequal. Pure Appl. Math. **4**(1), 6 (2003). Article 15
11. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: engineering an efficient SAT solver. In: Proceedings of the 38th Annual Design Automation Conference, DAC 2001, pp. 530–535. ACM, New York (2001)
12. Richardson, T., Urbanke, R.: Modern Coding Theory. Cambridge University Press, Cambridge (2008)
13. Sipser, M.: A complexity theoretic approach to randomness. In: Proceedings of the 15th ACM Symposium on Theory of Computing (STOC), pp. 330–335 (1983)
14. Soos, M.: CryptoMiniSat - a SAT solver for cryptographic problems (2009). https://www.msoos.org/cryptominisat5/
15. Stockmeyer, L.: On approximation algorithms for #P. SIAM J. Comput. **14**(4), 849–861 (1985)
16. Thurley, M.: sharpSAT – counting models with advanced component caching and implicit BCP. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, pp. 424–429. Springer, Heidelberg (2006). https://doi.org/10.1007/11814948_38
17. Valiant, L.G., Vazirani, V.V.: NP is as easy as detecting unique solutions. Theor. Comput. Sci. **47**, 85–93 (1986)
18. Zhao, S., Chaturapruek, S., Sabharwal, A., Ermon, S.: Closing the gap between short and long XORs for model counting. In: Schuurmans, D., Wellman, M.P. (eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 12–17 February 2016, Phoenix, Arizona, USA, pp. 3322–3329. AAAI Press (2016)