A flat histogram method for inference with probabilistic and deterministic constraints

Stefano Ermon, Carla Gomes, Ashish Sabharwal and Bart Selman
Department of Computer Science
Cornell University
Ithaca, NY 14850
{ermonste,gomes,sabhar,selman}@cs.cornell.edu

Abstract

Reasoning in a context where both probabilistic and deterministic dependencies are present at the same time is a challenging task with many real-world applications. Classic inference methods like Gibbs Sampling and Message Passing algorithms tend to give poor results in the presence of deterministic constraints, while purely logical reasoning techniques are not designed for probabilistic dependencies.

In this paper we show how to reduce many inference problems to that of computing the so-called density of states of a weighted Boolean formula for an appropriate energy function, where the density of states is defined as a function that for each energy level E gives the number of configurations with that energy.

We propose a novel Markov Chain Monte Carlo algorithm to compute the density of states that is based on flat histogram methods and naturally overcomes the ergodicity problems associated with deterministic constraints. Our experiments show that this method often converges quickly to a very accurate solution and can outperform general purpose techniques such as Gibbs sampling and specialized methods such as MC-SAT in a marginal computation task.

1 Introduction

Reasoning in a context where both probabilistic and deterministic dependencies are present at the same time is a challenging task with many real-world applications. Markov Chain Monte Carlo (MCMC) methods provide a general framework for sampling and probabilistic inference from complex probability distributions, as captured, for example, in a graphical model representation. However, in the presence of a set of hard constraints (i.e., deterministic dependencies), it often becomes difficult to even reach states that satisfy all such dependencies in the the Markov Chain.

We will consider a novel MCMC sampling strategy, inspired by the Wang-Landau method ([1]), which is a so-called *flat histogram* sampling strategy from statistical physics. Given a combinatorial space and an energy function (for instance, that describes the log-likelihood of each configuration), a *flat histogram* method is a sampling strategy based on an Adaptive Markov Chain that converges to a steady state where it samples uniformly from sets of configurations with equal energy. This form of sampling will spend approximately the same amount of time in areas with low density configurations (usually, low energy states) and in high density areas of the search space. Such a sampling strategy generally leads to a much broader coverage of the state space compared to more traditional MCMC approaches, such as Metropolis-Hastings or Gibbs. In particular, it solves the problems caused by near-deterministic dependencies, that greatly slow down inference by creating low probability regions that are difficult to traverse (and eventually breaking down the ergodicity in the limit of deterministic dependencies). Another advantage of this method is that we obtain

the full density of states distribution. This represents a rich description of the state space and we will show that upon defining the right energy function, the density of states can be used to infer complex statistical properties of the probability distribution, such as marginals for all possible levels of softness of the constraints.

Building on [2], we will consider a modification of the Wang-Landau method that incorporates a random-walk style component to focus the Markov Chain more quickly on areas where all hard constraints are satisfied. By enforcing a detailed balance condition, we maintain uniform sampling across the different energy levels and the consistency of the method. We will provide empirical data to show the practical effectiveness of our method by comparing it in a marginal computation task with general purpose techniques such as Gibbs sampling and specialized methods such as MC-SAT [3].

2 Probabilistic model

The focus of this paper is on complex probability distributions defined over a set of *possible worlds* represented by a set of N Boolean predicates (or propositional variables) $x_1, \ldots, x_N$. The probability is specified through combinatorial features or constraints that are represented as (CNF) formulas over the Boolean variables. Such constraints can be either *hard* or *soft*. The former are called hard because a world $x \in \{0,1\}^N$ has probability 0 unless it satisfies all of them. If it does, its probability is given by

$$P(x) = \exp\left(-\sum_{i \in \mathcal{C}} w_i \chi_i(x)\right)$$

where $\mathcal{C}$ is the set of soft constraints, w_i is the weight corresponding to the i-th *soft* constraint and $\chi_i(x) = 1$ if and only if x violates the i-th constraint. As $w_i \to \infty$, a *soft* constraint effectively becomes a *hard* constraint.

This factored representation is closely related to a graphical model where we use weighted Boolean formulas to specify clique potentials [4]. This is a natural framework to combine purely logical and probabilistic inference and in many cases it allows for a more natural way to encode the probabilistic dependencies of the system, compared for example to conditional probability tables. An example of such a setup is a grounded Markov Logic Network (see [3, 5]), that can be applied to study a variety of problems such as link prediction, collective classification, entity resolution, social network analysis and many others.

3 Density of states: problem definition

In statistical physics, given a system and an energy function $E(\cdot)$, the *density of states* (DOS) $n(\cdot)$ is the function $n : [0, \ldots, m] \to \mathbb{N}$ that maps energy levels to the number of configurations or microstates with that energy level:

$$E \mapsto |\{\sigma \in \{0,1\}^N : E(\sigma) = E\}|.$$

In our context, we are interested in computing the number of possible worlds (or configurations) that satisfy certain properties that are specified using an appropriate energy function, whose specific definition depends on the statistical properties we want to infer. For instance, we might define the *energy* of a configuration $E(\sigma)$ to be the number of constraints that are unsatisfied by σ (this is known as the density of states for unweighted boolean formulas [2]). Other possibilities include the sum of the weights of the violated *soft* constraints, or the number of unsatisfied *hard* constraints, or a combination of them, or any easy to compute function of a configuration. We formally define the specific form of the energy functions needed to compute marginals and for weight learning below. However, the MCMC algorithm we use to compute the density of states does not make any assumption about what the energy is actually representing. At least in principle, the only thing we need is a partitioning of the state space, where the energy is just an indexing over the subsets that compose the partition. In particular, as opposed to traditional sampling techniques such as Metropolis, the value of the energy does not guide the search in any way. For this reason, we think our approach will find many other applications to a variety of inference and learning tasks.

4 The flat histogram method

Building on [2], we propose a Markov Chain Monte Carlo method to compute the density of states based on the flat histogram idea that is inspired by recent developments of statistical physics [1] to avoid Metropolis sampling. The central idea of this method is that if we perform a random walk in the configuration space $\{0,1\}^N$ such that the probability of visiting a given energy level E is inversely proportional to the density of states $n(E)$, then a flat *visit histogram* is generated for the energy distribution of the states visited. Suppose we define a random walk with the following transition probability

$$p_{\sigma\to\sigma'} = \min\left\{1, \frac{n(E)}{n(E')}\right\} \tag{1}$$

of going from a configuration σ with energy E to a configuration σ' with energy E'. The detailed balance equation $P(\sigma)p_{E\to E'} = P(\sigma')p_{E'\to E}$ is satisfied when $P(\sigma) \propto 1/n(E)$. This leads to a flat histogram of the energies of the states visited because $P(E) = \sum_{\sigma:E(\sigma)=E} P(\sigma) = \text{const}$.

Since the density of states is unknown a priori, and computing it is precisely the goal of the algorithm, it is not possible to construct a random walk with transition probability (1). However it is possible to start from an initial guess of the DOS and keep changing the current estimate $g(\cdot)$ in a systematic way to produce a flat energy histogram and simultaneously make the density of states converge to the true value $n(E)$.

MCMC-FLATSAT(ϕ)
1 Start with a guess $g(E) = 1$ for all E
2 Start with a modification factor $F = F_0$
3 **repeat**
4 **repeat**
5 Generate a new state and accept with prob. given by eq. (1)
6 Adjust $g(E)$: $g(E) = g(E) \times F$
7 Increase visit histogram $H(E) \leftarrow H(E) + 1$
8 **until** until H is flat
9 Reduce F
10 Reset the visit histogram H
11 **until** F is close enough to 1
12 Normalize g
13 **return** g as estimate of n

The modification factor F plays a critical role because it controls the trade-off between the convergence rate of the algorithm and its accuracy. Large initial values of F imply a substantial diffusion rate and therefore fast convergence to a rather inaccurate solution. This rough initial estimate is subsequently refined as the value of F decreases until $F \approx 1$, at which point when a flat histogram is produced $g(E)$ has converged to the true density $n(E)$.

Due to statistical fluctuations, a perfectly flat histogram occurs with an extremely low probability. Therefore in our implementation we use a flatness parameter; in our experiments it is set so that an histogram is considered flat when all the values are between 90% and 100% of the maximum value, independently of F. The value of F is reduced according to the schedule $F \leftarrow \sqrt{F}$, with an initial value $F_0 = 1.5$; the impact of the schedule on the convergence rate is an open research question. By construction the DOS is obtained only up to constant factors: the normalization of g ensures that $\sum_E g(E) = 2^N$, where N is the number of variables in the formula.

4.1 Focused Random Walk

There are many ways in which a new state can be generated. The simplest strategy is to generate a new configuration by randomly flipping a variable, a method that in [2] is shown to be quite effective. Notice however that the detailed balance equation still holds when using an acceptance probability

$$\min\left\{1, \frac{g(E)T_{\sigma\to\sigma'}}{g(E')T_{\sigma'\to\sigma}}\right\} \tag{2}$$

instead of eq. (1), where $T_{\sigma\to\sigma'}$ is the probability of generating a configuration σ' while in state σ. Clearly we also need to ensure that whenever $T_{\sigma\to\sigma'} > 0$, $T_{\sigma'\to\sigma} > 0$ and that the connectivity of the state space is preserved.

Given that we are mostly interested in the regions of the state space where the *hard* constraints are satisfied (conventionally, low energy regions), a *greedy* approach in generating new configurations can significantly improve the convergence rate. An useful heuristic inspired by local search SAT solvers to generate new states is the following: given a truth assignment σ, if it is a satisfying assignment, flip a variable at random (so $T_{\sigma\to\sigma'} = 1/N$ when the Hamming distance $d_H(\sigma,\sigma') = 1$, zero otherwise). If σ is not a solution, then with probability p a variable is chosen from a violated clause and then flipped, and with probability $1-p$ a variable is flipped at random. With this approach, the probability is

$$T_{\sigma\to\sigma'} = (1-p)\frac{1}{N} + p\frac{\sum_{c\in C|i\in c}\chi_c(\sigma)}{\sum_{c\in C}\chi_c(\sigma)} = (1-p)\frac{1}{N} + p\frac{\#\text{violated clauses with variable } i}{\text{violated clauses}}$$

where σ and σ' differ only on the i-th variable. This approach can greatly reduce the number of steps needed for the Markov Chain to reach low energy configurations and solutions (that are frequently lower density states), leading in our experiments to convergence rates several times smaller than the ones obtained with random flipping.

5 Inference and learning with the density of states

Many inference tasks can be solved by defining an appropriate energy function (equivalently, a partitioning of the state space) and by computing the density of states associated with it. For instance, a fundamental inference task is that of computing the probability that a property encoded by a formula F_1 holds, given that we observe some evidence encoded by another formula F_2:

$$P(F_1|F_2) = \frac{P(F_1 \wedge F_2)}{P(F_2)} = \frac{\sum_{x\in X_{F_1\cap F_2}} P(x)}{\sum_{x\in X_{F_2}} P(x)}$$

where X_{F_i} is the set of worlds where F_i holds. We define a tuple-valued energy function $E'(\sigma)$ in the following way:

$$E'(\sigma) = \begin{cases} (0, \sum w_i\chi_i(\sigma)) & \text{if } \sigma \text{ satisfies } F_1 \wedge F_2 \\ (1, \sum w_i\chi_i(\sigma)) & \text{if } \sigma \text{ satisfies } F_2 \text{ but not } F_1 \\ (2, 0) & \text{if } \sigma \text{ does not satisfy } F_2 \end{cases}$$

Let $g(\cdot)$ be the density of states associated with E'. Then

$$P(F_1|F_2) = \frac{\sum_E g(0,E)\exp(-E)}{\sum_E (g(0,E) + g(1,E))\exp(-E)}$$

As a particular case, when $F_1 = x_i$ *we obtain the marginal of the* i*-th variable*. Notice that a finer grained partitioning (for instance, distinguishing how many hard constraints are violated) can speed up the convergence because it provides guidance to the search procedure, even if it is extracting more information.

5.1 Weight learning

The richer information provided by the entire density of states can be used to solve inference and learning tasks that go beyond the capabilities of conventional sampling techniques [6]. Suppose we have a set of soft constraints $\mathcal{S}_i \subseteq \mathcal{C}$ all with the same weight w_i that we want to learn from data (for instance, these constraints might all be groundings of the same first order formula in a Markov Logic Network). The gradient of the log-likelihood with respect to the weights is

$$\frac{\partial}{\partial w_i}\log P_w(X = x) = m_i(x) - \sum_{x'} m_i(x')P_w(X = x') \tag{3}$$

where m_i is the number of false groundings of the constraints $\mathcal{S}_i$ in the data x and P_w is the probability distribution with a vector of weights w. Let's define a partitioning of the state space (using an appropriate energy function) and a corresponding density g as follows

$$g(E,k) = |\{x \in \{0,1\}^N : m_i(x) = k, E\tau \le \sum_{j\neq i} w_j m_j(x) < (E+1)\tau\}|$$

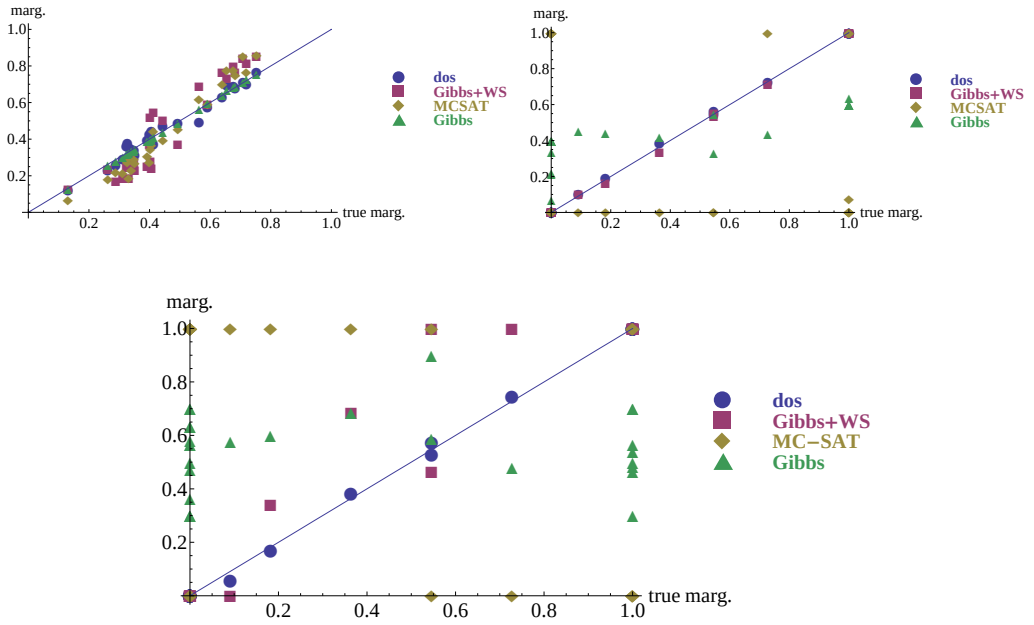

Figure 1: Correlation plots for the variable marginals as the weights increase. All constraints are soft. All methods are effective when the weights are small ($w = 1$, upper left picture). Pure Gibbs sampling begins to fail when the weights increase ($w = 5$, upper right). The `DOS` method provides the most accurate estimate when $w = 10$, bottom picture.

for some small enough step size τ. The second part of equation (3) can be rewritten as

$$\sum_{x'} m_i(x') P_w(X = x') \approx \sum_k \sum_E k g(E,k) \frac{1}{Z} e^{-w_i k} e^{-E\tau}$$

where $Z \approx \sum_k \sum_E g(E,k) e^{-w_i k} e^{-E\tau}$ (the approximation might be caused by the discretization with step size τ). This means that once we have computed the density $g(E,k)$, we can compute the i-th component of the gradient of the log-likelihood essentially at no cost. Moreover we can do it for all values of w_i, which means that we can try to solve for $\frac{\partial}{\partial w_i} \log P_w(X = x) = 0$.

6 Experimental results

In [2] we demonstrated the effectiveness of `MCMC-FlatSat` to compute the density of states of boolean formulas, both in terms of accuracy and efficiency. Here we explore its application to inference tasks, in particular for the computation of marginals. We compare the Focused Random Walk method (`DOS`) with Gibbs sampling with and without Walksat initialization (`Gibbs+WS` and `Gibbs`) and a specialized method `MC-SAT` [3]. We use the implementations found in the Alchemy system [6], and all the methods are run for the same amount of time.

6.1 All soft constraints

We start with an unweighted instance (spin glass) from a MAX-SAT Competition, and we add the same weight w to all the constraints (even though they are identical, changing the weights affects the marginals) . The instance has 27 variables so we can get ground truth by direct enumeration. In figure 1 we compare the results as the weights become larger in a correlation plot, where in the ideal case the estimated marginals would lie on the diagonal. While the `DOS` method is effective in all the weight ranges, Gibbs sampling breaks down when the constraints become almost deterministic. Moreover, the estimate obtained with the `DOS` method is more accurate than the one obtained with `MC-SAT`.

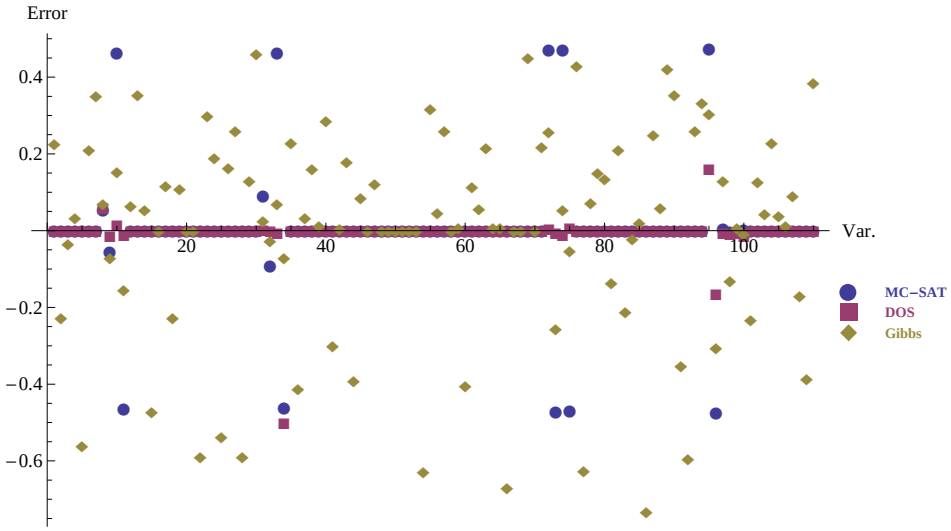

Figure 2: Errors of the estimated marginals for the 110 variables of the logistic instance with hard and soft constraints.

6.2 Hard and soft constraints

We start with a planning instance with 110 variables generated with SAT-Plan, with 461 hard constraints used to encode valid plans. Then we add some random soft constraints, encoding e.g. preferences of the planner. As an effect of the soft constraints, valid plans (solutions to all hard constraints) have different probabilities. As we can see figure 2, Gibbs sampling breaks down with deterministic constraints, and the estimate obtained with the `DOS` method is the most accurate one.

7 Conclusion

By designing a suitable partitioning of the state space, the density of states can be used to solve many inference and learning problems, such as marginal computation and weight learning. We introduced `MCMC-FlatSat`, a Markov Chain Monte Carlo technique based on the flat histogram method with a random walk style component to estimate the density of states of Boolean formulas with hard and soft constraints. We demonstrated the effectiveness of this approach on the marginal computation problem, where it outperforms current state of the art methods. Given the generality of this method, we expect to see many other applications both to learning and inference problems.

References

[1] F. Wang and DP Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053, 2001.

[2] S. Ermon, C. Gomes, and B. Selman. Computing the density of states of Boolean formulas. In *Proceedings of the 16th International Conference on Principles and Practice of Constraint Programming*, 2010.

[3] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proc. of AAAI-06*, volume 21, page 458, 2006.

[4] Michael I. Jordan. Graphical models. *Statistical Science*, 19(1):pp. 140–155, 2004.

[5] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[6] P. Domingos, S. Kok, H. Poon, M. Richardson, and P. Singla. Unifying logical and statistical ai. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 2–7. AAAI Press, 2006.