Importance Sampling over Sets: A New Probabilistic Inference Scheme

Stefan Hadjis, Stefano Ermon
Department of Computer Science
Stanford University, Stanford, CA, USA
{shadjis, ermon}@cs.stanford.edu

Abstract

Computing expectations in high-dimensional spaces is a key challenge in probabilistic inference and machine learning. Monte Carlo sampling, and importance sampling in particular, is one of the leading approaches. We propose a generalized importance sampling scheme based on randomly selecting (exponentially large) subsets of states rather than individual ones. By collecting a small number of extreme states in the sampled sets, we obtain estimates of statistics of interest, such as the partition function of an undirected graphical model. We incorporate this idea into a novel maximum likelihood learning algorithm based on cutting planes. We demonstrate empirically that our scheme provides accurate answers and scales to problems with up to a million variables.

1 INTRODUCTION

Probabilistic inference is one of the key computational challenges in statistical machine learning and Bayesian statistics. The key computational bottleneck for many statistical inference problems of interest lies in the computation of high-dimensional integrals. Examples include computing posterior probabilities, model averaging, and evaluating partition functions of undirected graphical models. The field is dominated by two main paradigms tracing their roots to statistics and physics: Monte Carlo sampling methods [1, 15, 19] and variational techniques [16, 29]. Monte Carlo sampling is an extremely influential idea leveraged by numerous algorithms which have found an enormous number of applications in many fields of scientific computation, with application ranging from machine learning, statistics, and physics. It is often ranked among the most important algorithms of all time in polls and surveys [4]. The basic idea is to estimate properties of a high-dimensional space (e.g., the integral of a function) by looking at a small number of representative states. The major difference between Monte Carlo schemes lies in how these representative states are selected and weighted.

Importance sampling (IS) is one of the most popular Monte Carlo sampling schemes. It is a simple and elegant idea, which is at the core of other widely used techniques such as Markov Chain Monte Carlo, Annealed Importance Sampling [21], and others [10]. The approach is very general: one can choose the samples randomly according to any desired *proposal distribution* (some mild restrictions have to be met), and IS provides a recipe to properly weight the samples and obtain an estimate for the original high-dimensional integral. The choice of the proposal distribution affects the variance of the estimate, and the number of samples required to obtain a statistically reliable estimate can grow exponentially in the problem size if the proposal distribution is poorly chosen. Unfortunately, designing a good proposal distribution is generally hard.

We introduce a more general scheme which we call importance sampling over sets (ISS) where we randomly select (large) subsets of states (rather than individual samples) using a generalized notion of proposal distribution called set-proposal distribution. Like traditional importance sampling, we provide a way to re-weight the samples and obtain an unbiased estimator for the original high-dimensional integral of interest. Intuitively, the idea is that by considering a very large (potentially, even exponential in the dimensionality of the problem) number of samples, one can significantly reduce the variance. Unfortunately, simply enumerating the samples would take exponential space. We therefore consider specially structured set-proposal distributions such that the set of samples can be represented in an *implicit* and *compact* way. The second main obstacle to overcome is that it is no longer possible to do enumeration-based inference on the samples. We therefore propose an approximation based on the importance weight of the heaviest configuration in the sampled set. For many classes of probabilistic models, e.g. log-supermodular [6], we can compute these statistics efficiently, e.g. using graphcuts. Surprisingly, we can show some strong formal guarantees

for this approximation. In particular, we identify a natural link between our scheme and some recently introduced probabilistic inference schemes based on randomized hashing and optimization [7, 8]. By reformulating these prior results within our framework, we show that there exists set-proposal distributions that are in some sense *universal* - they are guaranteed to give accurate answers using a small number of samples no matter what the underlying probabilistic model is.

We improve the accuracy and efficiency of our approach by developing a class of *adaptive* set-proposal distributions that can be tailored to the specific target probabilistic model leveraging the samples we draw from the model. We show that this approach provides very accurate estimates for the partition function of undirected graphical models on a range of benchmark problems. Our method is also extremely scalable: we are able to estimate the partition function for models with *up to one million variables* in a matter of minutes. Finally, we develop a new maximum likelihood parameter learning scheme based on our probabilistic inference framework. Our technique is very different from standard gradient descent approaches, and resembles structured prediction schemes such as structured SVM learning. We empirically show the effectivness of our technique on the standard MNIST handwritten digits dataset.

2 SETUP

Given an undirected graphical model with n binary variables, let $\mathcal{X} = \{0,1\}^n$ be the set of all possible configurations (variable assignments or possible states of the world). Define a weight function $w : \mathcal{X} \rightarrow \mathbb{R}^+$ that assigns to each configuration x a score proportional to its probability $p(x)$: $w(x) = \prod_{\alpha \in \mathcal{I}} \psi_\alpha(\{x\}_\alpha)$. The weight function is compactly represented as a product of factors or potentials. The *partition function* of the model Z is defined as $Z = \sum_{x \in \mathcal{X}} w(x) = \sum_{x \in \mathcal{X}} \prod_{\alpha \in \mathcal{I}} \psi_\alpha(\{x\}_\alpha)$. It is a normalization constant used to guarantee that $p(x) = w(x)/Z$ sums to one. Computing Z is typically intractable because it involves a sum over an exponential number of configurations, and is often the most challenging inference task for many families of graphical models. Computing Z is required for many inference and learning tasks, such as evaluating the likelihood of data for a given model, computing marginal probabilities, and comparing competing models of data [29, 17].

Given that probabilistic inference problems are intractable in the worst case [23], a number of approximate inference algorithms have been developed. There are two main families of algorithms: Monte Carlo sampling techniques and variational approximations. Variational methods are based on approximating the target distribution p using a family of tractable approximating distributions, and minimizing a notion of divergence. Sampling techniques are randomized approaches where the key idea is to estimate statistics of interest by looking at a small number of representative states.

3 IMPORTANCE SAMPLING

The simplest (naive) approach is to sample $x_1, \cdots x_M$ uniformly from $\mathcal{X}$, and estimate $\widehat{Z} = \frac{1}{M} \sum_{i=1}^{M} w(x_i) 2^n$. This is an unbiased estimator of Z as $\mathbb{E}[\widehat{Z}] = \frac{1}{M} \sum_{i=1}^{M} \sum_{x \in \mathcal{X}} \frac{1}{2^n} 2^n w(x) = Z$. The variance of this estimator can be very large since we are limited to a small number of samples M, while the number of possible configurations $|\mathcal{X}|$ is exponential in n. The variance can be reduced using *importance sampling* (IS) techniques, i.e. sampling using a proposal distribution (which is closer to $p(x)$) rather than uniformly [1, 15, 19]. Here, $x_1, \cdots x_M$ are sampled from $\mathcal{X}$ according to some proposal distribution $q(x)$, and weighted by their inverse likelihood, $\widehat{Z} = \frac{1}{M} \sum_{i=1}^{M} \frac{w(x_i)}{q(x_i)}$. This is also an unbiased estimator for Z.

Unfortunately, it is usually the case that the closer the proposal distribution q is to the original intractable $p(x)$, the harder it gets to sample from it. Markov Chain Monte Carlo sampling is one of the leading approaches for sampling from arbitrary distributions [1, 15, 19]. The key idea is to draw proper representative samples from $p(x)$ by setting up a Markov Chain over the entire state space which has to reach an equilibrium distribution. For many statistical models of interest, reaching the equilibrium distribution will require simulating the chain for a number of steps which is exponential in n. Unless there are special regularity conditions, if the random walk does not visit all the possible states it might miss some important parts. In practice, the approach will therefore only give approximate answers. There is generally little or no information on the quality of the approximation. In fact, the Markov Chain may get trapped in less relevant areas and completely miss important parts of the state space.

Most similar to our approach is Greedy Importance Sampling (GIS) [27], a reformulation of IS which achieves variance reduction by sampling blocks of variables from a proposal distribution and then searching for highly weighted regions in the target distribution. The blocks of points are non-overlapping and points within a block are ordered, allowing points in a block to be selected using a greedy search. This search increases the probability of blocks containing highly weighted points and outperforms naive methods which are unlikely to observe such points by chance. These blocks can be seen as a special-case of sets in the ISS technique, in which the sets are selected through search. Whereas GIS blocks are likely to contain highly weighted points due to explicit search, sets in ISS more generally contain highly weighted points by sampling any exponentially large subset of points and extracting statistics of interest. For example ISS allows the use of order-statistics (MAP/MPE estimation) which can often be com-

puted efficiently (e.g. using graphcuts, Viterbi), although the search method of GIS is another approach. The methods are orthogonal and future work can investigate incorporating explicit search or other techniques such as analytic marginalization within ISS to further reduce variance. Another key generalization of ISS is that sets of points can overlap, which grants additional freedom in selecting set-proposal distributions. For example when sets are defined by parity constraints, set-proposal distributions implement a strongly universal hash function, providing strong theoretical guarantees on the accuracy of the estimates.

4 IMPORTANCE SAMPLING OVER SETS

We propose a generalized importance sampling procedure, in which instead of randomly selecting a single configuration $\mathbf{x}$ we randomly select a (large) subset of configurations $S \subseteq \mathcal{X}$. Let $P(\mathcal{X})$ denote the power set of $\mathcal{X} = \{0,1\}^n$, i.e. the set of all subsets of $\mathcal{X}$. We define a probability distribution q over $P(\mathcal{X})$ as a **set-proposal distribution**. A set-proposal distribution induces the following function $\gamma : \mathcal{X} \to [0,1]$

$$\gamma(\mathbf{x}, q) = \sum_{S \in P(\mathcal{X})} 1(\mathbf{x} \in S) q(S) \tag{1}$$

Intuitively, $\gamma(\mathbf{x}, q)$ is the probability of $\mathbf{x}$ being contained in a set S sampled from q. We omit the dependency on q when the set-proposal distribution used is clear from the context. Standard proposal distributions used in Importance Sampling are a special case of set-proposal distributions, assigning zero probability to all subsets $S \subseteq \mathcal{X}$ such that $|S| \neq 1$. The following results generalizes the standard importance sampling result to our more general case.

Proposition 1. *Let q be any set-proposal distribution such that $w(\boldsymbol{x}) > 0$ implies $\gamma(\boldsymbol{x}, q) > 0$. Let $S \sim q$ denote a random sample from the set-proposal distribution q. Then $\sum_{x \in S} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x}, q)}$ is an unbiased estimator for the partition function Z.*

Proof.

$$\begin{aligned} Z &= \sum_{\mathbf{x} \in \mathcal{X}} w(\mathbf{x}) = \sum_{\mathbf{x} \in \mathcal{X}} w(\mathbf{x}) \frac{\gamma(\mathbf{x})}{\gamma(\mathbf{x})} \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})} \sum_{S \in P(\mathcal{X})} 1(\mathbf{x} \in S) q(S) \\ &= \sum_{S \in P(\mathcal{X})} q(S) \sum_{\mathbf{x} \in \mathcal{X}} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})} 1(\mathbf{x} \in S) \\ &= \sum_{S \in P(\mathcal{X})} q(S) \sum_{\mathbf{x} \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})} = E_{S \sim q}\left[\sum_{x \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})}\right] \end{aligned}$$

□

Note that when the set-proposal distribution q is a standard proposal distribution (over singletons), one recovers the standard importance sampling result. There are three main aspects to consider for the practical usability of Proposition 1. We need to 1) sample a subset S from q efficiently, 2) evaluate the importance weight $\gamma(\mathbf{x})$ tractably, 3) when S is (exponentially) large, represent S compactly and evaluate the summation. The first two considerations apply to traditional importance sampling as well. The third one is new. For example, if q deterministically chooses $S = \mathcal{X}$, then evaluating the estimator is just as hard as computing the partition function. As this extreme example suggests, the advantage is that by considering larger sets, one can significantly reduce the variance. The following corollary is very useful,

Corollary 1. *Let q be any set-proposal distribution such that $w(\boldsymbol{x}) > 0$ implies $\gamma(\boldsymbol{x}) > 0$. Let $S \sim q$ denote a random sample from the set-proposal distribution q. Then $E_{S \sim q}\left[\max_{\boldsymbol{x} \in S} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x})}\right]$ is a lower bound for the partition function Z.*

Proof. Since the weights are non-negative $w(\mathbf{x}) \geq 0$, it follows that $\max_{\mathbf{x} \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})} \leq \sum_{x \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})}$ and the claim follows from Proposition 1 by linearity of expectation. □

Notice that if q is a standard proposal distribution, i.e. $q(S) = 0$ if $|S| \neq 1$, the estimators $\sum_{x \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})}$ and $\max_{\mathbf{x} \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})}$ coincide. In general, the value of $\max_{\mathbf{x} \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})}$ can be exponentially far from $\sum_{x \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x})}$, for example in the case of a constant (uniform) weight function $w(\cdot)$. The upside is that the max statistic, i.e. computing the mode of the distribution, is often more tractable. For example, there are numerous classes of probabilistic models, such as attractive Ising models, where one can find the mode of the distribution (MAP/MPE query) in polynomial time, while computing the partition function is NP-hard [11, 14].

4.1 EXAMPLES OF SET-PROPOSAL DISTRIBUTIONS

In both examples below, let $m \leq n$, let $v_m(\mathbf{x}) = \{v_i(x_i), i = 1, \cdots, m\}$ be a family of marginal distributions over individual variables. Let b_i be independent samples from $v_i(x_i)$ for $i = 1, \cdots, m$.

4.1.1 Constraining Variables

We can define a set-proposal distribution q where to sample a set we define $S = \{\mathbf{x} \in \mathcal{X} : x_i = b_i, \forall i \in \{1, \cdots, m\}\}$. Note that $\gamma(\mathbf{x}) = \prod_{i=1}^m q(x_i) = \prod_{i=1}^m v_i(x_i)^{b_i}(1 - v_i(x_i))^{1-b_i}$. The set can be represented compactly using m equations (equivalently, additional factors to be added

to the graphical model that clamp some variables to certain values). Intuitively, this approach samples a set S where $|S| = 2^{n-m}$ by constraining or "clamping" variables $x_1, \cdots, x_m$ to fixed binary values.

4.1.2 Parity Constraints

As a second example of a set-proposal distribution, let $A \in \{0,1\}^{m \times n}$ be a binary matrix with rows a_i. We define a set-proposal distribution q according to the following generative process. To sample a set S from q, we define $S = \{\mathbf{x} \in \mathcal{X} : a_i \mathbf{x} = b_i \mod 2, \forall i \in \{1, \cdots, m\}\}$. It can be seen that given any $\mathbf{x} \in \mathcal{X}$, the probability that $\mathbf{x}$ belongs to a randomly chosen $S \sim q$ is again $\gamma(\mathbf{x}) = \prod_{i=1}^{m} q(x_i)$. This is the probability that $\mathbf{x}$ satisfies m parity equations with randomly chosen right-hand side coefficients. The set can be represented compactly using m linear equations modulo 2. Parity constraints can be represented compactly as a product of factors, using a linear number of extra variables [7].

5 MULTIPLE PROPOSAL DISTRIBUTIONS

As noted earlier, the lower bound obtained with Corollary 1 given by $L(S,q) = \max_{\mathbf{x} \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q)}$ might be loose compared to $A(S,q) = \sum_{x \in S} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q)}$, which is an unbiased estimator of Z by Proposition 1. Here we are making explicit the dependence of $\gamma(\mathbf{x}, q)$ on the set-proposal distribution q. Intuitively, this approximation is accurate when the weight distribution over S is peaked, i.e. the mode is a good approximation of the "total area". On the other hand, the lower bound is loose when there are "many" configurations in S that have a weight comparable to the one of the heaviest assignment. If that is the case, it is intuitively possible to randomly subsample the set S and obtain a smaller set $S' \subseteq S$ such that $\max_{\mathbf{x} \in S} w(\mathbf{x}) \approx \max_{\mathbf{x} \in S'} w(\mathbf{x})$. Since $|S'| < |S|$, the gap between the approximation introduced by considering only the mode of the weight distribution on S' yields a smaller error. This suggests the use of another set-proposal distribution q' that is more likely to propose smaller sets S' compared to q. Because we do not know a priori if the bound is tight or not, this discussion motivates a more general scheme that relies on multiple set-proposal distributions. By letting the typical size of a sampled set change, e.g. from 1 to $|\mathcal{X}| = 2^n$, we can sample from a wide variety of configurations and accurately predict $\log Z$ for distributions which are peaked to various degrees.

Proposition 2. *Let $\mathcal{Q} = (q_1, \cdots, q_k)$ be a family of set-proposal distributions. Let $S_1 \sim q_1, S_2 \sim q_2, \cdots, S_k \sim q_k$. Suppose that for all i, $w(\boldsymbol{x}) > 0$ implies $\gamma(\boldsymbol{x}, q_i) > 0$. Then $\frac{1}{k} \sum_{i=1}^{k} \sum_{x \in S_i} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x},q_i)}$ is an unbiased estimator for the partition function Z.*

Proof.

$$E_{S_1 \sim q_1, \cdots, S_k \sim q_k} \left[\frac{1}{k} \sum_{i=1}^{k} \sum_{x \in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x}, q_i)} \right] = \frac{1}{k} \sum_{i=1}^{k} E_{S_i \sim q_i} \left[\sum_{x \in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x}, q_i)} \right] = Z$$

where the last step follows from Proposition 1. □

The following corollary follows, and is implemented by Algorithm 1 with input $T = 1$.

Corollary 2. *Let $\mathcal{Q} = (q_1, \cdots, q_k)$ be a family of set-proposal distributions. Let $S_1 \sim q_1, S_2 \sim q_2, \cdots, S_k \sim q_k$. Suppose that for all i, $w(\boldsymbol{x}) > 0$ implies $\gamma(\boldsymbol{x}, q_i) > 0$. Then $\frac{1}{k} \sum_{i=1}^{k} \max_{\boldsymbol{x} \in S_i} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x},q_i)}$ is in expectation a lower bound for the partition function Z.*

Proof. Follows immediately from Corollary 1. □

Input : $w : \mathcal{X} \to \mathbb{R}^+$, T, k, q_i for $i = 1, \ldots, k$
Output: Estimate of $\log Z = \log \sum_{x \in \mathcal{X}} w(\mathbf{x})$

for $i = 1, \ldots, k$ **do**
  Sample $S_i^1 \ldots S_i^T$ according to q_i
  for $t = 1, \ldots, T$ **do**
    $\mathbf{m}_i^t = \max_{\mathbf{x} \in S_i^t} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q_i)}$
  end
  $M_i \leftarrow \text{Median}(\mathbf{m}_i^1 \ldots \mathbf{m}_i^T)$
end
Return $\frac{1}{k} \sum_{i=1}^{k} M_i$

Algorithm 1: Set importance sampling

If we now let the typical size of a sampled set change, e.g. from $|S| = 1$ to $|S| = |\mathcal{X}| = 2^n$, the magnitude of the various $\gamma(\mathbf{x}, q_i)$ varies exponentially. Therefore it is often the case that many terms in the sum $\sum_{i=1}^{k} \max_{\mathbf{x} \in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q_i)}$ will not contribute significantly to the overall estimate of Z. In practice, a sufficiently accurate lower bound is obtained by T samples of only the highest weighted sets,

$$\max_{i=1}^{k} \max_{\mathbf{x} \in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x}, q_i)} \tag{2}$$

The main advantage of this approach is computational, as we have now reduced the inference procedure to a *single* optimization problem. The following result shows that this still provides a valid lower bound:

Corollary 3. *Let $\mathcal{Q} = (q_1, \cdots, q_k)$ be a family of set-proposal distributions. Suppose that for all i, $w(\boldsymbol{x}) > 0$ implies $\gamma(\boldsymbol{x}, q_i) > 0$. Let $S_i^1, \cdots, S_i^T \sim q_i$ be i.i.d samples from the i-th set-proposal distribution q_i. Then $\max_{i=1}^{k} \text{Median} \left(\max_{\boldsymbol{x} \in S_i^1} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x},q_i)}, \cdots, \max_{\boldsymbol{x} \in S_i^T} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x},q_i)} \right)$ is with high probability an approximate lower bound for the partition function Z.*

Proof. For every i, let us denote $L(S_i) = \max_{\mathbf{x}\in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q_i)}$. Then by Proposition 1, $E_{S_i\sim q_i}[L(S_i)] \leq Z$. Therefore by Markov's inequality, $P\left[\max_{\mathbf{x}\in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q_i)} \geq 4Z\right] \leq \frac{1}{4}$. Let $S_i^1, \cdots, S_i^T$ be samples from q_i. It follows from Chernoff's inequality that

$$P[\text{Median}(L(S_i^1), \cdots, L(S_i^T)) \geq 4Z] \leq \exp\left(-\frac{T}{24}\right)$$

therefore from the union bound

$$P[\cup_{i=1}^k \text{Median}(L(S_i^1), \cdots, L(S_i^T)) \geq 4Z] \leq k\exp\left(-\frac{T}{24}\right)$$

Therefore

$$P[\max_{i=1}^k \text{Median}(L(S_i^1), \cdots, L(S_i^T)) \leq 4Z] \geq 1-k\exp\left(-\frac{T}{24}\right)$$

□

Corollary 3 is implemented in Algorithm 1 with line 8 changed to return the maximum M_i instead of the mean.

To make the procedure of Corollary 3 clear, consider again a family of set-proposal distributions q_i constructed by constraining variables as in the example of section 4.1.1. This can be implemented in Algorithm 1 by setting $k = n + 1$ and selecting q_i to constrain the first $i - 1$ variables: The outer-loop (line 1) searches for the q_i which contributes most towards the estimate of $\log Z$ by, in each iteration, enforcing i "hard" variable constraints which limit the set of possible configurations by defining sets S_i^t where $|S_i^t| = 2^{n+1-i}$. Notice under this setup that if the maximum iteration is $i = 1$ (no variables constrained, $|S| = |\mathcal{X}| = 2^n$), then this is equivalent to approximating $\log Z$ by the MAP/MPE configuration. Conversely, if the maximum is at $i = n + 1$, then this is equivalent to naive importance sampling based on proposal distribution q_{n+1} (all variables constrained, $|S| = 1$). If the heaviest weighted set is not one of these two special cases, then the set importance sampling method will produce a more accurate estimate of $\log Z$. Also note that since empirically most of the iterations do not contribute significantly to the overall estimate of $\log Z$, the outer loop in Algorithm 1 over all n variables can search with a larger granularity or logarithmically for the heaviest i. In practice, fixing the number of iterations in the outer loop to 10 (sampling sets with a granularity of $\frac{n}{10}$ constrained variables) is both accurate and fast to run. In fact, this can be taken a step further by skipping the loop altogether and searching for the heaviest weighted set as a *single* optimization problem: rather than a loop which incrementally adds "hard" variable constraints, we can add all the variable constraints at once as "soft" constraints which an optimization oracle may choose to satisfy. The reward for satisfying these constraints matches the scaling $\frac{1}{\gamma(\mathbf{x},q_i)}$. Formulating the estimate of $\log Z$ as a single optimization problem is useful for learning, see section 7.

5.1 RELATIONSHIP WITH RANDOMIZED HASHING

The advantage of using multiple proposal distributions is that one might be able to reduce the variance of the estimator, in accordance with the intuitive motivation presented earlier. In fact, it can be shown that there exists set-proposal distributions (based on universal hash functions) such that a polynomial number of samples is sufficient to obtain concentration of the estimate around the mean. The surprising result is that these proposal distributions are "universal", in that they are guaranteed to give accurate estimates (constant factor approximations) for *any* weight function $w(\cdot)$, i.e., any underlying graphical model.

Let $\mathcal{S} \subseteq P(\mathcal{X})$ be a family of sets defined as $\mathcal{S} = \{\{\mathbf{x} \in \mathcal{X} : A\mathbf{x} = \mathbf{b} \bmod 2\}, A \in \{0,1\}^{i\times n}, b \in \{0,1\}^i\}$. Let q_i be a set-proposal distribution where to sample from q_i we randomly choose each entry of A, b uniformly at random (independently). Then it can be shown that $\gamma(\mathbf{x}) = 2^{-i}$. For a state space $\mathcal{X} = \{0,1\}^n$, let us consider a family of n proposal distributions $\mathcal{Q}_P = (q_0, \cdots, q_n)$. These set-proposal distributions can be interpreted as implementing a strongly universal hash function, where each element $\mathbf{x} \in \mathcal{X}$ is sampled by the i-th proposal distribution q_i with probability 2^{-i}, and elements are sampled pairwise independently [9, 3, 12, 7, 8, 13]. As noted above, we can sample from q_i tractably, and represent sets $S \in \mathcal{S}$ in a compact way. Theorem 1 from [8] implies the follwing remarkable result

Corollary 4. *Let $\mathcal{Q}_P = (q_0, \cdots, q_n)$ be defined as above. Let $L(S_i) = \max_{\boldsymbol{x}\in S_i} \frac{w(\boldsymbol{x})}{\gamma(\boldsymbol{x},q_i)}$ for every i. Then for any weight function $w(\cdot)$ and $1 > \delta > 0$, $\sum_{i=1}^n \text{Median}(L(S_i^1), \cdots, L(S_i^T))$ is with probability at least $1 - \delta$ a constant factor approximation of Z when $T = \Theta(n \ln n/\delta)$.*

Reinterpreted in our set-proposal distribution framework, Corollary 4 is important because it shows that there exists a family of universal set-proposal distributions that are guaranteed to work well for any underlying target probability distribution p.

Although sets $S \in \mathcal{S}$ defined by parity constraints can be represented compactly, the resulting optimization problems $\max_{\mathbf{x}\in S_i} \frac{w(\mathbf{x})}{\gamma(\mathbf{x},q_i)}$ are generally difficult to solve, even when $w(\cdot)$ can be tractably optimized, i.e., $\max_{\mathbf{x}\in\mathcal{X}} w(\mathbf{x})$ can be solved efficiently. Rather than take a worst-case approach and consider a proposal distribution that is guaranteed to work for any weight function w as in [9, 3, 12, 7, 8, 13], in this paper we consider a more general class of set-proposal distributions. In particular, we construct proposal distributions that are tailored to particular probabilistic models (and weight function w). The main advantage of this approach is that we can leverage the structure of the original problem, and the corresponding optimization problems

will be easier to solve, leading to massive improvements in scalability. This is similar in spirit to traditional importance sampling, where one typically uses some prior knowledge on the underlying probabilistic model to construct good proposal distributions.

6 ADAPTIVE SET IMPORTANCE SAMPLING SCHEME

Similarly to standard adaptive importance sampling schemes (e.g. [22]), we propose an adaptive procedure where set-proposal distributions are adapted based on the samples collected. This is an enhancement of Algorithm 1 in that its iterative procedure can be exploited to adaptively improve the input set-proposal distributions.

Recall from section 4.1.1 that a set-proposal distribution can be defined in which sets S_i are sampled by constraining variables $x_1, \cdots, x_i$. For a fixed i there are 2^i such sets S_i (where $|S_i| = 2^{n-i}$), and as in the previous section for each i we sample T such sets S_i^t. Next, let $\mathbf{e}_i^t = \arg\max_{\mathbf{x} \in S_i^t} \frac{w(\mathbf{x})}{\gamma(\mathbf{x}, q_i)}$. We define empirical marginal distributions $\widehat{v}_{i+1}(\mathbf{x})$ based on the fraction of samples $\mathbf{e}_i^t$ that have variables x_1 to x_{i+1} set to one (with Laplace smoothing). Intuitively, the adaptive set importance algorithm performs the same iteration as Algorithm 1, sampling sets by incrementally constraining variables x_1 to x_i, except that it interpolates the *input* proposal distribution with the empirical marginal distributions from the previous iteration to define a new set-proposal distribution for the current iteration. The full details of the algorithm are shown in Algorithm 2. During each iteration, as in Algorithm 1, generate T sets $S_i^1 \ldots S_i^T$ where each represents a set of configurations $\mathbf{x}$ in which the first i binary variables are "clamped" to 0 or 1. The solutions $\mathbf{e}_i^t = \arg\max_{\mathbf{x} \in S_i^t} \frac{w(\mathbf{x})}{\gamma(\mathbf{x}, q_i)}$ produce T samples which define the empirical marginal distribution $\widehat{v}_{i+1}(\mathbf{x})$, and this empirical marginal distribution is used to sample sets in iteration $i+1$. For the first iteration, the variable x_1 is sampled according to any proposal distribution (uniformly by default). Note also that when obtaining the MAP configuration of many variables is intractable, the iteration can begin with any number of variables constrained according to any proposal distribution (not just x_1). ISS is still guaranteed to perform at least as well (and often much better) as IS by selecting subsets small enough that calculating the mode by brute force enumeration is tractable.

7 LEARNING

Because set importance sampling provides fast and scalable partition function estimates, it can be used for learning. We consider the standard problem of maximum likelihood learning of the parameters of an undirected graphical model. Given samples $\mathbf{x}_1, \cdots, \mathbf{x}_M$ from a parameterized probability distribution $p_\theta(\mathbf{x}) = \frac{1}{Z(\theta)} \exp(\theta \phi(\mathbf{x}))$,

Input : $w : \mathcal{X} \to \mathbb{R}^+, T$
Output: Estimate of $\log Z = \log \sum_{\mathbf{x} \in \mathcal{X}} w(\mathbf{x})$

$M_0 \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} w(\mathbf{x})$
Define $\widehat{v}_1(\mathbf{x})$ as uniform marginal over x_1
for $i = 1, \ldots, n$ **do**
  Sample $S_i^1 \ldots S_i^T$ using marginals $\widehat{v}_i(\mathbf{x})$ as in 4.1.1
  for $t = 1, \ldots, T$ **do**
    $\mathbf{m}_i^t, \mathbf{e}_i^t = \max, \arg\max_{\mathbf{x} \in S_i^t} \frac{w(\mathbf{x})}{\gamma(\mathbf{x}, q_i)}$
  end
  $M_i \leftarrow \text{Median}(\mathbf{m}_i^1 \ldots \mathbf{m}_i^T)$
  Compute $\widehat{v}_{i+1}(\mathbf{x})$ based on $\mathbf{e}_i^1 \ldots \mathbf{e}_i^T$ (with Laplace smoothing). This is the fraction of argmax results $\mathbf{e}_i^t$ with $x_j = 1$, for $j \in \{1, \cdots, i+1\}$.
end
Return $\frac{1}{n+1} \sum_{i=0}^{n} M_i$

Algorithm 2: Adaptive set importance sampling

find maximum likelihood estimate of the parameters

$$\max_\theta \sum_{i=1}^{M} \log p_\theta(x_i) \tag{3}$$

which can be equivalently written as $\max_\theta \theta \frac{1}{M} \sum_{i=1}^{M} \phi(x_i) - \log Z(\theta)$. It is well known that solving this parameter learning problem is very challenging because it requires inference to evaluate the partition function (or its gradient). In this section we show how our importance sampling scheme can be used to approximate the value of the partition function, leading to a new learning algorithm. The algorithm we obtain is similar to structured prediction learning algorithms and cutting plane techniques[30, 28, 26], used for example in training structured support vector machines. A key difference is that our approach is used to (approximately) optimize the likelihood (in a generative setting), rather than minimizing a loss function in a discriminative setting.

7.1 LEARNING ALGORITHM

Structured support vector machines (SSVM) [30] and other structured prediction learning methods [18, 5] are trained by solving a convex optimization problem in which the number of constraints is exponential. The problems can be solved tractably by iteratively constructing a sufficient subset of constraints and employing an optimization oracle to find the next constraint to include. In this way the subset of the constraints is enlarged iteratively and provides a lower bound on the optimization objective.

The learning approach based on set importance sampling is similar, but using the set importance sampling technique as an optimization oracle. Beginning from the logarithm of equation (3), $\max_\theta \theta \frac{1}{M} \sum_{i=1}^{M} \phi(\mathbf{x}_i) - \log Z(\theta)$, we can introduce a variable α which takes the place of $\log Z$ and cast the optimization as follows

$$\begin{aligned} \underset{\theta,\alpha}{\text{maximize}} \quad & \theta \frac{1}{M} \sum_{i=1}^{M} \phi(\mathbf{x}_i) - \alpha \\ \text{subject to} \quad & \alpha \geq \log Z(\theta) \end{aligned}$$

We then express $Z(\theta)$ using the approximation given by equation (2) as

$$\alpha \geq \log Z(\theta) \geq \max_{i=1}^{k} \max_{\mathbf{x} \in S_i} \theta\phi(\mathbf{x}) - \log \gamma(\mathbf{x}, q_i) \qquad (4)$$

Note this is an exponentially large set of *linear* constraints in θ and α, and therefore corresponds to a linear program (LP). Because the number of constraints is exponential in the number of variables, as in structured learning we consider only a subset C of constrained configurations $\mathbf{x}$. The reduced LP becomes,

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & \theta \frac{1}{M} \sum_{i=1}^{M} \phi(\mathbf{x}_i) - \alpha \\ \text{subject to} \quad & \alpha \geq \theta\phi(\mathbf{x}) + \beta(\mathbf{x}) \; \forall \; \mathbf{x} \in C \end{aligned} \qquad (5)$$

where $\beta(\mathbf{x}) = \max_{i|\mathbf{x} \in S_i}(-\log \gamma(\mathbf{x}, q_i)))$ and intuitively is the maximum importance weight for a given $\mathbf{x}$ under all set-proposal distributions q_i (see Appendix A). C is initially set to the training data set $\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$, and is enlarged during each learning iteration by searching for the most violated constraint, i.e. $\max_x \theta\phi(\mathbf{x}) + \beta(\mathbf{x})$.

The full learning procedure is described in Algorithm 3. The input to the algorithm are the M training examples, the vector of sufficient statistics ϕ and the (optional) choice of set-proposal distributions $q_1 \cdots q_k$ (for example uniform, based on the training examples, or adaptive if no q_i are provided). Each iteration of learning begins by finding the optimal weights for the LP in equation (5). Following the solution of the LP, we obtain the pair (θ_i, α_i). Then, using these learned weights θ_i, importance sampling over sets is used to approximate $\log Z$ for various set-proposal distributions and importance weights, and each of these samples (modes) is added to constraint set C. Note that Algorithm 3 takes advantage of an optimization oracle to solve the LP in equation (5). Another optimization oracle is used within our importance sampling over sets procedure to estimate $\log Z$ (by optimizing equation (4) using the current parameter estimate θ_i). Intuitively, the value of the partition function is not just approximated using the MAP assignment, but thanks to the importance weights, we are able to obtain better estimates. For example, if using the current weight vector θ_i the distribution is close to uniform, an approximation based on the MAP assignment would be poor, but we can still get good approximation to the partition function thanks to the importance weights.

Because at each iteration a (convex) linear program is solved to obtain the weights θ, at each iteration the weights obtained for the constraints C are guaranteed to be globally optimal for the approximate likelihood objective. This is in contrast to gradient-based learning algorithms. Moreover, as more constraints are added to C at each iteration, C approaches $\mathcal{X}$ and the LP objective is guaranteed to decrease monotonically towards the optimal approximate log likelihood of the training data.

Input : $\mathbf{x}_m, m = 1, \ldots, M$, $\phi(\mathbf{x})$, T, $q_i, i = 1, \ldots, k$
Output: Learned weight parameters θ

converged $\leftarrow$ False
i $\leftarrow 0$
$C \leftarrow \{\mathbf{x}_m, m = 1, \ldots, M\}$
while *not converged* **do**
  $i \leftarrow i + 1$
  θ_i, α_i = solve LP (5) subject to C
  for $t = 1, \ldots, T$ **do**
    $\mathbf{x}^*_{i,t}$, $\log Z_{est,t}$ = Run ISS with θ_i, $\{q_1, \ldots q_k\}$
  end
  $\log Z_{est} \leftarrow \underset{t=1,\ldots,T}{\text{median}} \log Z_{est,t}$
  if $\alpha_i \geq logZ_{est}$ **then**
    converged $\leftarrow$ True
  else
    $C \leftarrow C \cup \left\{\mathbf{x}^*_{i,t}, t = 1, \ldots, T\right\}$
  end
end
Return θ_i

Algorithm 3: Iterative learning algorithm

8 EXPERIMENTAL RESULTS

8.1 PARTITION FUNCTION

One application of importance sampling over sets is for problems in which computing $\max w(\mathbf{x})$ is tractable, but $\sum_x w(\mathbf{x})$ is intractable. An example are functions which are log-supermodular. For such problems we can leverage fast optimization (for example using graph cuts), as long as $w(\mathbf{x})/\gamma(\mathbf{x})$ stays tractable. For example, it is sufficient that $\log(1/\gamma(\mathbf{x}))$ is supermodular.

We evaluated importance sampling over sets (ISS) against standard inference algorithms: junction tree (EXACT), belief propagation (BP), mean-field (MF), the MAP configuration (MAP), and naive importance sampling (IS). For junction tree, belief propagation and mean-field, the libDAI library was used [20].

First, we evaluated importance sampling over sets for functions which are not log-supermodular, to demonstrate the effectiveness of the method on general models. Table 1 shows estimates of the log partition function of the Ising Models from the 2011 Probabilistic Inference Challenge (PIC2011) [1]. These models have "mixed" interactions and therefore are not log-supermodular. For general functions

[1] www.cs.huji.ac.il/project/PASCAL/showNet.php

which are not log-supermodular, each $\arg\max$ in ISS was solved as an integer quadratic program (IQP) in CPLEX, with a search granularity of $\frac{n}{10}$ to find the heaviest sets as discussed in section 5. The log partition function estimates in Table 1 use a uniform proposal distribution to show that even in the absence of a proposal distribution the ISS method performs better than both naive importance sampling and the MAP configuration (which are both special-cases of ISS).

Our second experiment validated ISS for functions which are log-supermodular, and thus for which ISS can be run in polynomial time. Fig. 1 evaluates the importance sampling over sets method on attractive (log-supermodular) Ising models with varying coupling strengths. Models have a field factor of 2.0, although we observed that a range of field factors gave almost identical results. For these log-supermodular potentials, optimization was performed with graph cuts using the OpenGM [2] library. For these experiments, importance sampling algorithms (IS and ISS) use the adaptive importance sampling scheme as a proposal distribution, where first ISS was run and the final empirical marginals in iteration n were also used for IS.

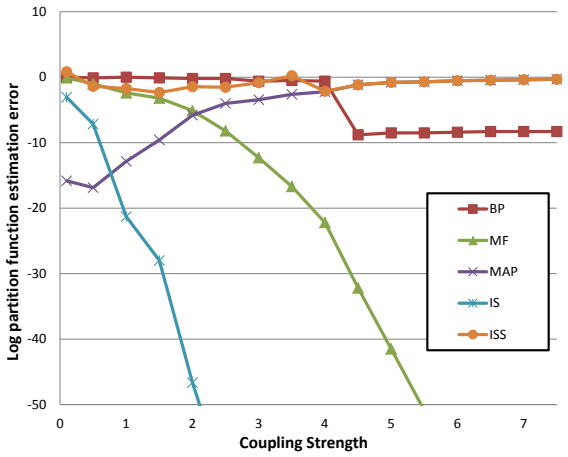

Figure 1: Log partition function error of various inference algorithms for 10x10 Ising grids with attractive (log-supermodular) interactions, a field factor of 2.0, and various coupling strengths. Importance sampling (IS) and importance sampling over sets (ISS) use adaptive set importance sampling.

Across a range of attractive Ising models, the set importance sampling technique provides very accurate estimates of the log partition function. Moreover, due to the log-supermodularity of the potentials, the ISS technique scales to much larger models, providing accurate estimates in polynomial time while other algorithms fail to converge. Tables 2 and 3 show the log partition function estimates and run-times for various Ising grid sizes, ranging from 10x10 to 1000x1000. Notably, for the 300x300 models mean-field did not converge, but was still run for 10 minutes to give a solution. Similarly for 1000x1000 models, belief propagation did not converge but gave a solution after 10 minutes. For 1000x1000 models mean-field did not complete a single iteration within 10 minutes.

Table 2: Log partition function estimates for various Ising model sizes. "–" indicates that no solution was obtained after 10 minutes. As in Table 1, ISS estimates are between MF (which tends to under-estimate) and BP (which tends to over-estimate)

	BP	**MF**	**MAP**	**IS**	**ISS**
10	207.6	202.7	202.0	161.2	206.4
20	840.3	817.7	825.3	593.5	832.3
50	5195	4940	5110	3704	5125
100	20930	19910	20679	14670	20690
300	1.91E5	1.82E5	1.88E5	1.35E5	1.88E5
1000	2.11E6	–	2.09E6	1.48E6	2.09E6

Table 3: Time (in seconds) to estimate logZ for Ising model sizes. "–" indicates that the algorithm did not converge within 10 minutes.

	EXACT	**BP**	**MF**	**MAP**	**IS**	**ISS**
10	1	1	1	1	1	1
20	–	1	1	1	1	1
50	–	5	8	1	1	5
100	–	15	112	1	1	3
300	–	119	–	8	1	27
1000	–	–	–	105	15	300

Finally, we extend the evaluation beyond Ising models by analyzing restricted Boltzmann machines (RBMs). Table 4 shows log partition function estimates for the largest RBM in [25] (784 visible units, 500 hidden units, trained on the MNIST dataset). AIS is the Annealed Importance Sampling technique described in that work. BP failed to converge. MF converged quickly but was less accurate than AIS. The quick convergence of mean-field was also noted by [24]. AIS was run in two modes, "no data" which estimated logZ from the model alone, and "data" which additionally used the training data to initialize the algorithm. In a similar spirit, due to the quick convergence of MF, and further demonstrating the flexibility of ISS to use any choice of proposal distribution, we ran mean-field to obtain marginals and used these as the proposal distribution for both IS and ISS. By leveraging MF as a proposal distribution ISS matches the accuracy of AIS with data. The ISS approach is valid even when no data is available.

8.2 LEARNING

In this final section we present preliminary analysis and empirical justification for the learning algorithm. We gen-

Table 1: Comparison of methods estimating the log partition function for Ising Models. The Importance Sampling (IS) and Importance Sampling over Sets (ISS) methods uses a uniform proposal distribution run over 5 random seeds, with the median presented. Shown in brackets next to ISS is the median number of constrained variables in the heaviest weighted set. The best estimate for each model is shown in bold.

	EXACT	**BP**	**MF**	**MAP**	**IS**	**ISS (c)**
grid10x10.f10	697.9	738.2	601.6	695.8	20.4	**697.8** (3)
grid10x10.f10.wrap	767.5	837	695.4	766.5	65.85	**767.9** (2)
grid10x10.f15.wrap	1146	1247	1036	1145.2	65.2	**1146.6** (2)
grid10x10.f5.wrap	390.1	419.7	355.1	387.8	66.4	**389.2** (2)
grid20x20.f10	3021	3234	2592	3015.7	299.1	**3017.1** (2)
grid20x20.f15	4520	4756	3947	4517.3	309.3	**4518.7** (2)
grid20x20.f2	671.7	**677.9**	621.6	635.7	282.9	637.8 (21)
grid20x20.f5	1531	1674	1391	1521.6	289.0	**1522.4** (1)

Table 4: Log partition function estimates for a restricted Boltzmann machine (RBM) trained on the MNIST dataset. Annealed importance sampling (AIS) was run with and without MNIST data for initialization. BP did not converge. IS and ISS were initialized with mean-field marginals as a proposal distribution and require no data.

Algorithm	$\log Z$
AIS (no data)	446.2
AIS (data)	451.1
BP	–
MF	437.5
MAP	71.6
IS	447.2
ISS	450.2

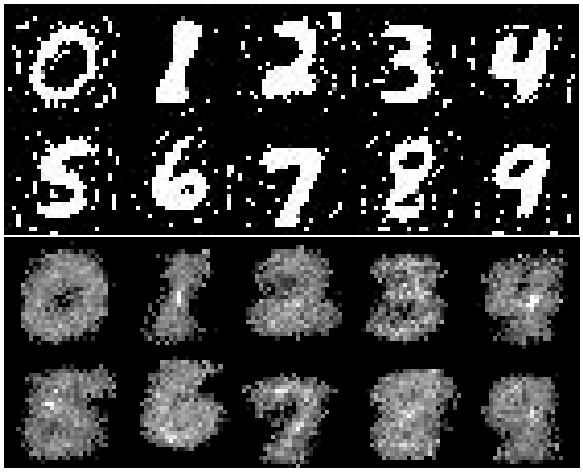

Figure 2: Visualization of learned weights of an undirected naive Bayes model trained generatively on the MNIST dataset. The top image shows weights after 5 iterations of training while the bottom image shows weights after 1000 iterations. Large positive weights are shown in white and large negative weights are shown in black.

eratively trained a naive Bayes model represented as an MRF on the MNIST handwritten digit dataset (size 28x28 images) and observed the algorithm's capability to learn weights which accurately modeled the data. The learned model contained 794 variables and 7840 parameters. Examples were used as described in Algorithm 3. Fig. 2 shows a visualization of the learned weights as training progresses. The top image in Fig. 2 shows weights after 5 iterations of training, while the bottom image shows weights after 1000 iterations. Early in training the model captures with high confidence the most common patterns in the digits, but also noise. As training progresses, the model learns to generalize and differentiate between random noise and statistical variations in the data. Adaptive ISS was used as a proposal distribution, and similar results were obtained using marginals defined by the training data.

A straightforward extension of this work is extending the learning to latent variable models such as restricted Boltzmann machines, which the cutting-plane technique may be well-suited to given the accuracy of ISS in estimating RBM partition functions. We leave learning larger and more complex models to future work as the current contribution focuses on the importance sampling over sets technique.

9 CONCLUSIONS

We introduced a novel probabilistic inference algorithm called importance sampling over sets, based on randomly selecting (exponentially large) subsets of states rather than individual ones as in traditional importance sampling. By solving MAP inference queries over the sampled sets we obtain estimates of the partition function of undirected graphical models. This idea was incorporated into a novel maximum likelihood learning algorithm where the optimization oracle was used to obtain cutting planes. We demonstrated empirically that our scheme provides accurate answers on a range of benchmark instances and scales to very large problems with up to a million variables.

References

[1] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[2] T. Beier B. Andres and J. H. Kappes. OpenGM: A C++ library for discrete graphical models. *ArXiv e-prints*, 2012.

[3] S. Chakraborty, K. Meel, and M. Vardi. A scalable and nearly uniform generator of SAT witnesses. In *Proc. of the 25th International Conference on Computer Aided Verification (CAV)*, 2013.

[4] Barry A Cipra. The best of the 20th century: editors name top 10 algorithms. *SIAM news*, 33(4):1–2, 2000.

[5] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

[6] Josip Djolonga and Andreas Krause. From MAP to marginals: Variational inference in Bayesian submodular models. In *Neural Information Processing Systems (NIPS)*, 2014.

[7] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Optimization with parity constraints: From binary codes to discrete integration. In *Proc. of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

[8] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *Proc. of the 30th International Conference on Machine Learning (ICML)*, 2013.

[9] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Low-density parity constraints for hashing-based discrete integration. In *Proc. of the 31st International Conference on Machine Learning (ICML)*, pages 271–279, 2014.

[10] V. Gogate and R. Dechter. SampleSearch: Importance sampling in presence of determinism. *Artificial Intelligence*, 175(2):694–729, 2011.

[11] Leslie Ann Goldberg and Mark Jerrum. The complexity of ferromagnetic ising with local fields. *Combinatorics, Probability and Computing*, 16(01):43–61, 2007.

[12] Carla P. Gomes, A. Sabharwal, and B. Selman. Model counting: A new strategy for obtaining good bounds. In *Proc. of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 54–61, 2006.

[13] Carla P. Gomes, Willem Jan van Hoeve, Ashish Sabharwal, and Bart Selman. Counting CSP solutions using generalized XOR constraints. In *Proc. of the 22nd National Conference on Artificial Intelligence (AAAI)*, 2007.

[14] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on computing*, 22(5):1087–1116, 1993.

[15] Mark Jerrum and Alistair Sinclair. The markov chain monte carlo method: An approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems*, pages 482–520. PWS Publishing, Boston, MA, 1997.

[16] Michael I. Jordan, Z. Ghahramani, Tommi Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[17] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[18] Alex Kulesza and Fernando Pereira. Structured learning with approximate inference. In *Advances in neural information processing systems*, pages 785–792, 2007.

[19] N.N. Madras. *Lectures on Monte Carlo Methods*. American Mathematical Society, 2002.

[20] Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, August 2010.

[21] Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

[22] Man-Suk Oh and James O. Berger. Adaptive importance sampling in monte carlo integration. *Journal of Statistical Computation and Simulation*, 41:143–168, 1992.

[23] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1):273–302, 1996.

[24] Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep Boltzmann machines. In *Proc. of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[25] Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proc. of the 25th International Conference on Machine Learning (ICML)*, 2008.

[26] Sunita Sarawagi and Rahul Gupta. Accurate max-margin training for structured output spaces. In *Proceedings of the 25th international conference on Machine learning*, pages 888–895. ACM, 2008.

[27] Dale Schuurmans. Greedy importance sampling. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

[28] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. In *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[29] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[30] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1169–1176. ACM, 2009.

A DERIVATION OF $\beta(\mathbf{x})$

Equation 5 introduces $\beta(\mathbf{x})$ in order to allow the single optimization problem approximating $\log Z$,

$$\max_{i=1}^{k} \max_{\mathbf{x} \in S_i} \theta\phi(\mathbf{x}) - \log \gamma(\mathbf{x}, q_i)$$

to be written in the form

$$\theta\phi(\mathbf{x}) + \beta(\mathbf{x}) \; \forall \, \mathbf{x} \in C$$

This reformulation is needed because, more generally, the cutting-planes technique requires that the lower-bound of $\log Z$ be written in the form

$$\log Z(\theta) \geq \max_{\mathbf{x} \in C} f(\theta, g(\mathbf{x}))$$

i.e. as the maximization over a set C of variable configurations $\mathbf{x}$ of a term which is linear in the parameter vector θ and which contains some (possibly nonlinear) function of $\mathbf{x}$ (the term must be linear in θ in order for Equation 5 to be a linear program). If the lower bound is expressed in such a form, it can then be equivalently represented by linear constraints of the form

$$\alpha \geq f(\theta, g(\mathbf{x})) \; \forall \, \mathbf{x} \in C$$

This section completes the derivation of Equation 5 from Equation 4 by showing how Equation 4 can be written as a maximization over configurations $\mathbf{x}$.

Proposition 3. *There exists a set C and function $\beta(\boldsymbol{x})$ such that $\log Z \geq \theta\phi(\boldsymbol{x}) + \beta(\boldsymbol{x}) \; \forall \, \boldsymbol{x} \in C$.*

Proof. From Equation 4,

$$\log Z(\theta) \geq \max_{i=1}^{k} \max_{\mathbf{x} \in S_i} (\theta\phi(\mathbf{x}) - \log \gamma(\mathbf{x}, q_i))$$

$$= \max_{\mathbf{x} \in \bigcup_{i=1}^{k} S_i} (\theta\phi(\mathbf{x}) + \max_{i | \mathbf{x} \in S_i} (-\log \gamma(\mathbf{x}, q_i)))$$

$$= \max_{\mathbf{x} \in C} (\theta\phi(\mathbf{x}) + \beta(\mathbf{x}))$$

□

Where $\beta(\mathbf{x}) = \beta(\mathbf{x}, q_1, \ldots, q_k) = \max_{i | \mathbf{x} \in S_i} \log \gamma(\mathbf{x}, q_i)$ and C is the union of all $\mathbf{x}$ in each sampled set $S_i \sim q_i$. Intuitively, given any configuration of variables $\mathbf{x}$, $\beta(\mathbf{x})$ represents the maximum scale factor (importance weight) of $\mathbf{x}$ for all set-proposal distributions q_i. For multiple $S_i^t \sim q_i, t = 1, \ldots, T$, it is necessary once again that

$$\log Z(\theta) \geq \max_{i=1}^{k} \operatorname*{median}_{t=1,\ldots,T} \max_{\mathbf{x} \in S_i^t} (\theta\phi(\mathbf{x}) - \log \gamma(\mathbf{x}, q_i))$$

be written in the form

$$\theta\phi(\mathbf{x}) + \beta(\mathbf{x}) \; \forall \, \mathbf{x} \in C$$

Taking the same approach,

$$\log Z(\theta) \geq \max_{i=1}^{k} \operatorname*{median}_{t=1,\ldots,T} \max_{\mathbf{x} \in S_i^t} (\theta\phi(\mathbf{x}) - \log \gamma(\mathbf{x}, q_i))$$

$$= \max_{\mathbf{x} \in \bigcup_{i=1}^{k} \bigcup_{t=1}^{T} S_i^t} (\theta\phi(\mathbf{x}) + \max_{i | \mathbf{x} \in \bigcup_{t=1}^{T} S_i^t} \operatorname*{median}_{t=1,\ldots,T} (-\log \gamma(\mathbf{x}, q_i)))$$

In practice it also works well to replace the median with the max, as Corollary 3 proves an approximate lower bound and the bound is made tighter by taking the max over T samples. Making this substitution,

$$\log Z(\theta) \geq \max_{i=1}^{k} \max_{t=1}^{T} \max_{\mathbf{x} \in S_i^t} (\theta\phi(\mathbf{x}) - \log \gamma(\mathbf{x}, q_i))$$

$$= \max_{\mathbf{x} \in \bigcup_{i=1}^{k} \bigcup_{t=1}^{T} S_i^t} (\theta\phi(\mathbf{x}) + \max_{i | \mathbf{x} \in \bigcup_{t=1}^{T} S_i^t} (-\log \gamma(\mathbf{x}, q_i)))$$

$$= \max_{\mathbf{x} \in C} (\theta\phi(\mathbf{x}) + \beta(\mathbf{x}))$$