A Sampling Method Based on LDPC Codes

Xuhong Zhang[1], Gregory W. Wornell[1]

[1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

Massachusetts Institute of Technology

Introducton

Inference problems are ubiquitous but often hard ➡ Approximate inference methods needed!

Sampling is a popular approximate inference algorithm based on Law of Large Numbers.

A wish list for an ideal sampling method:

-Tractable computational costs: space and time
-Correct samples
-Independent samples
-Good scaling behaviour with data dimension
-Efficient use of randomness
-….

A sampling method satisfying all of these does NOT exist! Compromises need to be made somewhere.

Our Contribution: we proposed a novel sampling method based on LDPC codes that works for distributions over discrete alphabets. It offers a different trade-off among various desirable features. In particular, in addition to **good scaling behaviour**, the proposed scheme has **good control over independence properties** and **makes efficient utilization of randomness**.

Proposed Sampling Scheme

Inspiration 1: Typical Sequences

Let $P_x(\cdot)$ be the target distribution from which we want to sample. Assume $\boldsymbol{y} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^t)$ are t independent samples from $P_x(\cdot)$. We can define the set of typical sequences:

$$\mathcal{T}_\epsilon^t = \{\boldsymbol{y} \in \{0,1\}^{Nt} : |\frac{1}{t}\log_2 P_x(\boldsymbol{y}) + NH_0 \mid \le \epsilon\}$$

where $NH_0 = -\sum_{\boldsymbol{x}} P_x(\boldsymbol{x}) \log_2 P_x(\boldsymbol{x})$.

Asymptotic Equipartition Property (AEP) states that for large enough t, the sequences in $\mathcal{T}_\epsilon^t$:

- are roughly equiprobable with probability $\sim 2^{-tNH_0}$
- have combined probability close to 1 ➡ there are $\sim 2^{tNH_0}$ typical sequences

Idea: Samples for $P_x(\cdot)$ can be obtained (e.g. by marginalization) if we have typical sequences. But how can we get typical sequences?

Inspiration 2: Compression

Compression (of discrete source) can be thought of as associating with each typical sequence an index. Instead of the sequences, the index is stored.

$|\mathcal{T}_\epsilon^t| \approx 2^{tNH_0} \ll 2^{tN} =$ # of all sequences

➡ Bits to represent index ≪ bits in the sequence

- Compression scheme: given the typical sequence, come up with an index.
- We want the opposite: **given an index, can we recover a typical sequence?**

(binary) LDPC Codes:

$$\mathbb{H} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \end{pmatrix} \in \{0,1\}^{K \times Nt}, \boldsymbol{y} \in \{0,1\}^{Nt}$$

$\boldsymbol{z} = \mathbb{H}\boldsymbol{y}$ is the parity bits associated with $\boldsymbol{y}$

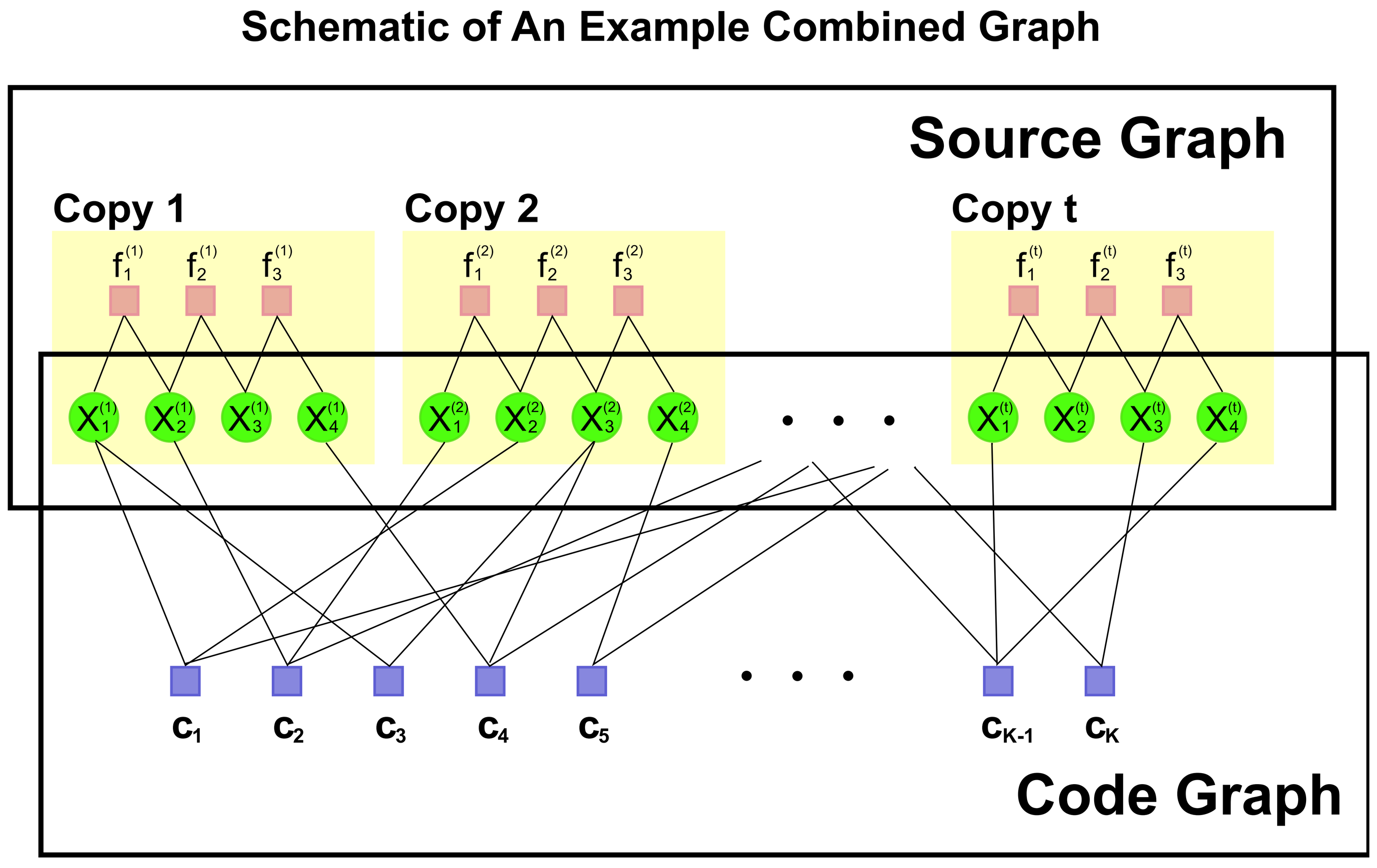
We essentially use the LDPC code as a hash function for the typical sequences!

Idea: LDPC codes have been used for compression, let us reverse the process to do sampling!

Construction of Combined Graph

1. The **source graph** consists of t independent copies of target distribution factor graph;
2. Attached to the source graph is an LDPC code, whose factor graph we call **code graph**;
3. Source graph and code graph share the variables, and together are called **combined graph**.

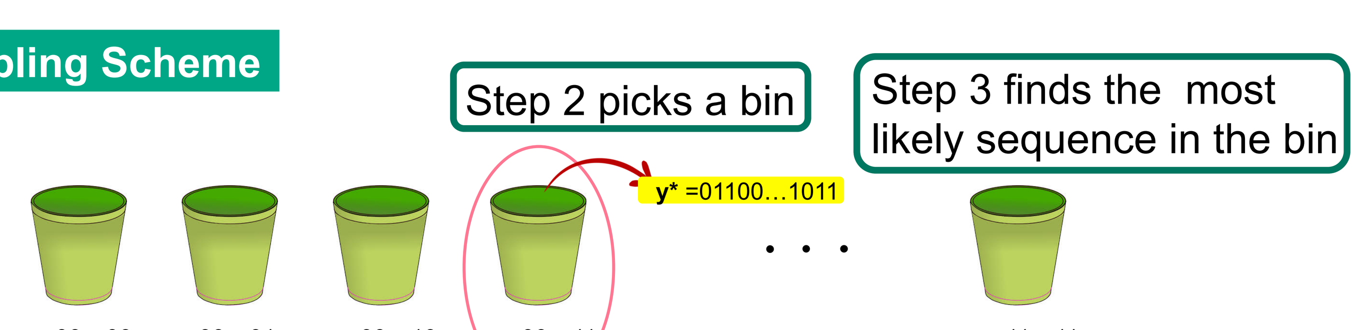
Schematic of An Example Combined Graph



Factor Nodes   Variable Nodes   Check Nodes

LDPC-Based Sampling Scheme

Each choice of parity bits corresponds to a bin containing sequences



1) Given target distribution $P_x(\cdot)$, construct the corresponding combined graph
2) Flip K independent fair coins and use the results as the values of parity bits
3) Find an LDPC codeword $(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^t)$ that maximizes $\prod_{i=1}^t P_x(\boldsymbol{x}^i)$, i.e. the most likely sequence among the ones correspond to the selected parity bits

Theoretical Correctness

Remember the goal is to uniformly sample from typical sequences. It turns out that if the code rate is matched to the entropy rate of $P_x(\cdot)$, a randomly chosen LDPC code is with high probability a good **hash function** for typical sequences: **almost all bins contain exactly one typical sequence and no sequence more likely than a typical one**. More precisely:

Theorem 1: Given $P_x(\cdot)$ over $\{0,1\}^N$, $\epsilon > 0, \delta > 0$, let t be large enough s.t. $\mathbb{P}(\mathcal{T}_\epsilon^t) \ge 1 - \frac{\delta}{2}$. Then there exists an LDPC code with parity check matrix $\mathbb{H} \in \{0,1\}^{K \times Nt}$ where $K \sim NtH_0$, for which the proposed sampling scheme produces a sequence in $\mathcal{T}_\epsilon^t$ with probability at least $1 - \delta$.

So far we assumed large enough t, but exactly how large?

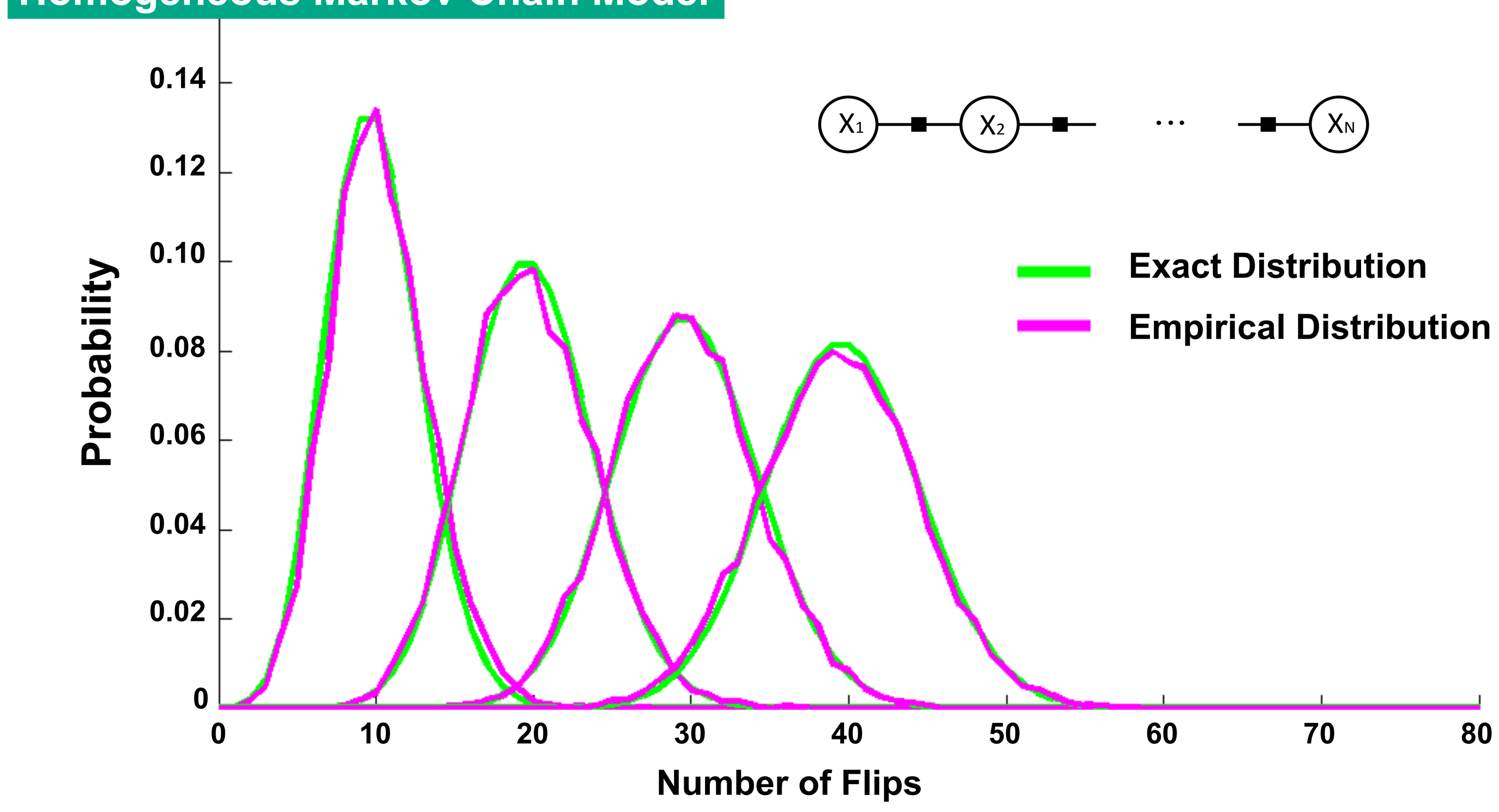
Theorem 2: If $H_0 < 1$, it is sufficient to have

$$t > \frac{1}{H_0 + \epsilon - \log_2(1 + H_0 \ln 2 + o\left(\frac{1}{N}\right))} \log_2 \frac{4}{\delta} \approx \frac{2}{2\epsilon + H_0^2 \ln 2} \log_2 \frac{4}{\delta}$$

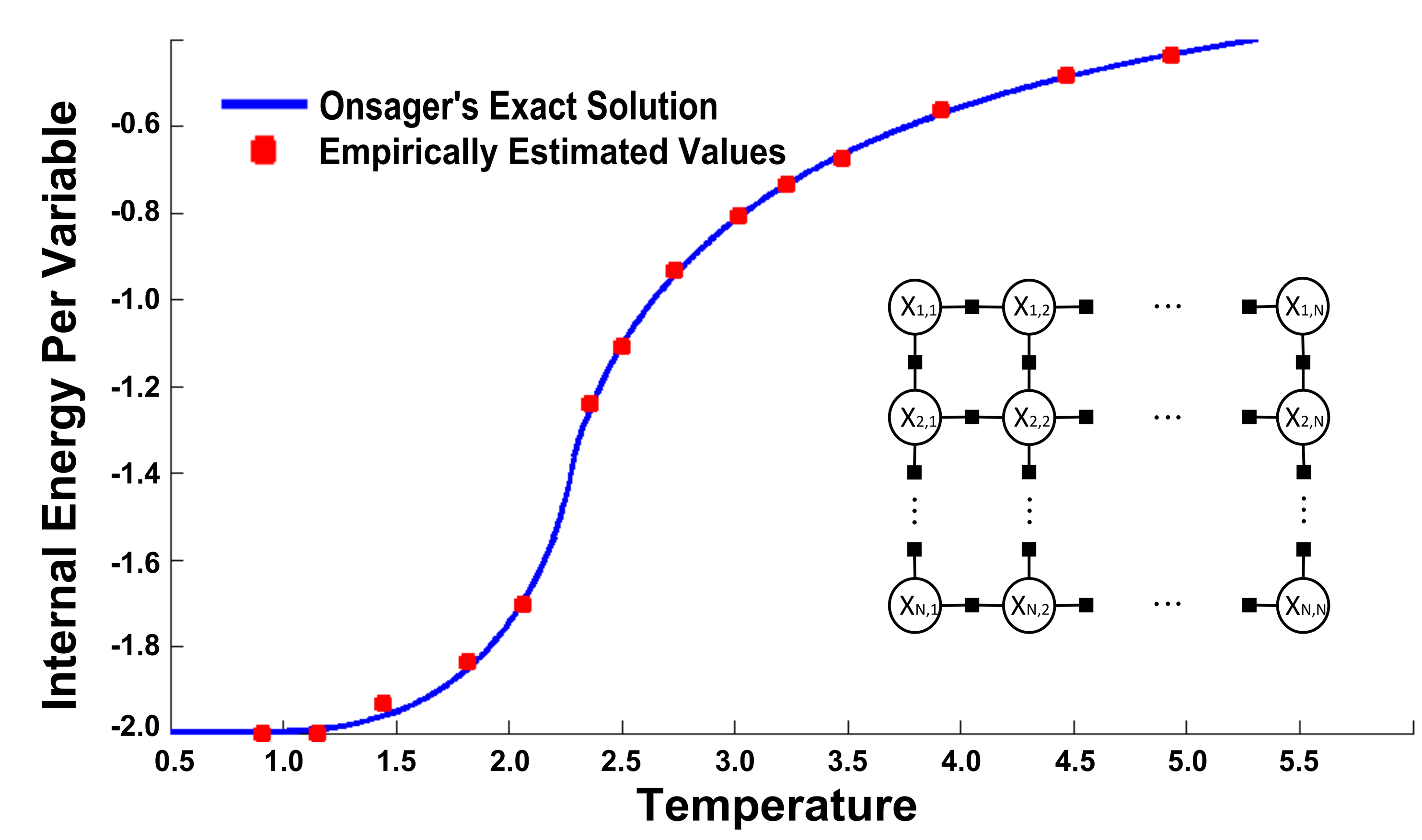
Experiment Results

Of course, step 3 in the proposed scheme (i.e. finding the most likely sequence corresponding to the selected parity bits) **is infeasible in practice**. Instead, **Belief Propagation** algorithm is implemented to **approximately** simulate the proposed scheme. The following experiments show that the approximate simulation of proposed scheme performs as we expect it to.

Homogeneous Markov Chain Model



Homogeneous 2D Ising Model



Discussion

The 'best' sampling method is application-dependent: different methods provide different trade-offs.

	Rejection Sampling	Importance Sampling	MCMC Methods	Proposed Scheme
Applicable to Continuous Variables?	Yes	Yes	Yes	No
Quality of Proposal Distribution Vital?	Yes	Yes	No	No
Applicable to High-Dimension Data?	No	No	Yes	Yes
Guaranteed Correctness?	Yes	No	Asymptotic	Approximate
Independent Samples?	Yes	N/A	No	Approximate

Highlights of some particularly desirable features of the proposed scheme:

a.Good scaling behaviour and applicability to high-dimensional data due to the distributed nature of message-passing.

b.Economical use of randomness: randomness per sample needed is close to the theoretical minimum.

c.Good control over independence properties of the produced samples, and thus seems less susceptible to the local minima problem.