Sparse State

Hee Ryang Choi

February 8, 2023

Sparse State

A single contraction of estimated tensor network produces sparse state, representing amplitudes of

- Randomly chosen L uncorrelated groups of bit strings, where $L = 2^{47}$, with 47 closed qubits
- Each group containing l correlated bit strings, where $l = 2^{6}$, with 6 open qubits
- In our reference, the qubit ids of open qubits are $11, 19, 28, 29, 37, 44$

Histogram of approximate bitstring probabilities

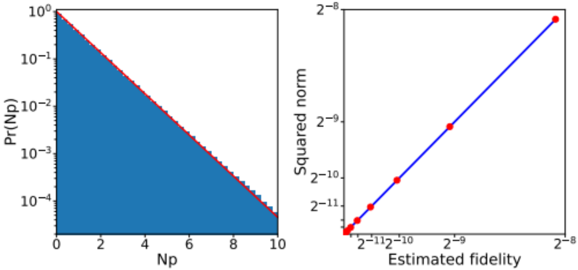


FIG. 3. (Left:) Histogram of approximate bitstring probabilities $p(\mathbf{s}) = |\widehat{\psi}(\mathbf{s})|^2/\mathcal{N}_s$ for 2^{26} bitstrings obtained from the Sycamore circuits with $n = 53$ qubits and $m = 20$ cycles. $\mathcal{N}_s$ is the norm factor and $N = 2^n$. The estimated fidelity $\widehat{\psi}(\mathbf{s})$ to the true final state $\psi(\mathbf{s})$ is $F \approx 0.0037$. The red line denotes the Porter Thomas distribution. (Right:) Comparison between the estimated fidelity (blue lines) and the norm factor of $\widehat{\psi}_L(\mathbf{s})$ obtained by summing over a fraction of paths.

Porter-Thomas Distribution

Porter-Thomas Distribution can be defined as

$$Prob(p) = De^{-Dp} \tag{1}$$

for measurement probabilities $p = p_s(q)$, with sample number D. Here, $p_s(q) = \langle q| U\rho_0 U^* |q\rangle$ is simulated ideal probability of q.

- Fraction of bitstrings with theoretical probability in $[p, p + dp]$ is $Prob(p)dp = De^{-Dp}dp$
- Total number of bitstrings is $N(p)dp = D^2 e^{-Dp}dp$
- Probability that a bitstring in $[p, p + dp]$ is sampled is $pN(p)dp = pD^2 e^{-Dp}dp = f(p)dp$

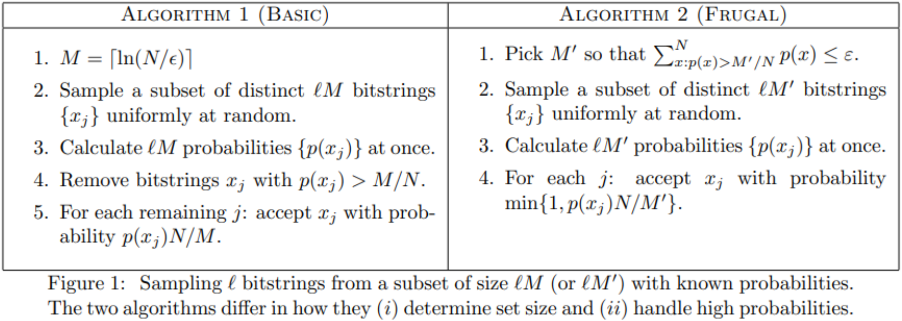
Porter-Thomas Distribution

Thus the average probability of a sampled bitstring will be

$$\begin{aligned}\langle P(q_i)\rangle &= \int_0^1 p f(p)\, dp = \int_0^1 p^2 D^2 e^{-Dp}\, dp \\ &= \frac{2}{D}\left(1 - e^{-D}\left(\frac{D^2}{2} + D + 1\right)\right) \approx \frac{2}{D}.\end{aligned}$$

Rejection Sampling Algorithm

The Rejection Sampling Algorithm works as follows. In our reference (I. L. Markov, “Quantum supremacy is both closer and farther than it appears”), they use $\epsilon = 10^{-3}$ and $n = 49$, $M = 41$, $N = 2^{41}$. Also we assumed that the distribution has Porter Thomas Distribution.



Algorithm 1 (Basic)	Algorithm 2 (Frugal)
1. $M = \lceil \ln(N/\epsilon) \rceil$ 2. Sample a subset of distinct ℓM bitstrings $\{x_j\}$ uniformly at random. 3. Calculate ℓM probabilities $\{p(x_j)\}$ at once. 4. Remove bitstrings x_j with $p(x_j) > M/N$. 5. For each remaining j: accept x_j with probability $p(x_j)N/M$.	1. Pick M' so that $\sum_{x:p(x)>M'/N}^{N} p(x) \leq \varepsilon$. 2. Sample a subset of distinct $\ell M'$ bitstrings $\{x_j\}$ uniformly at random. 3. Calculate $\ell M'$ probabilities $\{p(x_j)\}$ at once. 4. For each j: accept x_j with probability $\min\{1, p(x_j)N/M'\}$.

Figure 1: Sampling ℓ bitstrings from a subset of size ℓM (or $\ell M'$) with known probabilities. The two algorithms differ in how they (i) determine set size and (ii) handle high probabilities.

Basic Rejection Algorithm

If we follow basic rejection algorithm, we can estimate the expected number ϵ of probabilities $p(x) > M/N$ using Porter-Thomas distribution as

$$\mathbf{Exp}\Big[\sum_{p(x)\geq M/N} 1\Big] = \epsilon = N\int_{M}^{\infty} e^{-t}dt = Ne^{-M}$$

Frugal Rejection Algorithm

To reduce the number M of probabilities per output bistring to M' without significantly increasing the error, we can set $M'(\epsilon)$ to satisfying

$$\sum_{x:p(x)>M'/N}^{N} p(x) \leq \epsilon \tag{2}$$

Also the distribution $p'(x)$ from Basic Rejection Algorithm samples has statistical variational distance from the true distribution as

$$\frac{1}{2}_{x}\|p(x) - p'(x)| = \mathbf{Exp}\{\sum_{x:p(x)>M'/N}^{N} p(x)\} \leq \epsilon \tag{3}$$

Frugal Rejection Algorithm

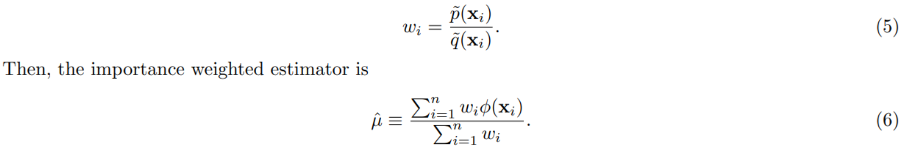
To get a tight bound on the number of probabilities needed per bitstring, can use Porter-Thomas statistics

$$\sum_{x:p(x)>M'/N}^{N} p(x) = \int_{M'}^{\infty} te^{-t} dt = e^{-M'}(1+M') \tag{4}$$

The statistical distance decrease exponentially with the number of probabilities M' per bitstring. For example, $M' = 10$ gives $\epsilon = 5 \times 10^{-4}$. In the case of the Frugal Rejection Algorithm, it can be used even if it is not a Porter-Thomas distribution, but the analytical error bound is unknown. After this algorithm, with lM' probabilities, number of samples are in $[l - \sqrt{lM'}, l + \sqrt{lM'}$

Importance Sampling

To use Importance Sampling, sample *n* items $x_1, \cdots x_n$ *q*, with easy distribution *q*. Our goal is to estimate average of real distribution *p*. Then, define



$$w_i = \frac{\tilde{p}(\mathbf{x}_i)}{\tilde{q}(\mathbf{x}_i)}. \tag{5}$$

Then, the importance weighted estimator is

$$\hat{\mu} \equiv \frac{\sum_{i=1}^{n} w_i \phi(\mathbf{x}_i)}{\sum_{i=1}^{n} w_i}. \tag{6}$$

Or, we can simply think as

$$E_{x \sim p}[f(x)] = \int p(x) f(x) dx = \int \frac{p(x)}{q(x)} q(x) f(x) dx = E_{x \sim q}\left[\frac{p(x)}{q(x)} f(x) \right]$$

Markov Chain Acceptance Rejection

Let ρ be a probability density on $\mathbb{R}^d$ called the *target density*, $Q = (q(\,\cdot\,|x))_{x\in\mathbb{R}^d}$ be a family of probability densities on $\mathbb{R}^d$ referred to as *proposal densities* and $\alpha\colon \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$ be a function, which may depend on ρ and Q, called *acceptance probability function*. Starting with some initial point $X_1 \in \mathbb{R}^d$, iterate for $k \in \mathbb{N}$:

(i) Draw a sample $Y_k \sim q(\,\cdot\,|X_k)$, independent of $X_1, \dots, X_{k-1}$ and $Y_1, \dots, Y_{k-1}$.

(ii) Compute the acceptance probability $\alpha_k = \alpha(X_k, Y_k) \in [0,1]$.

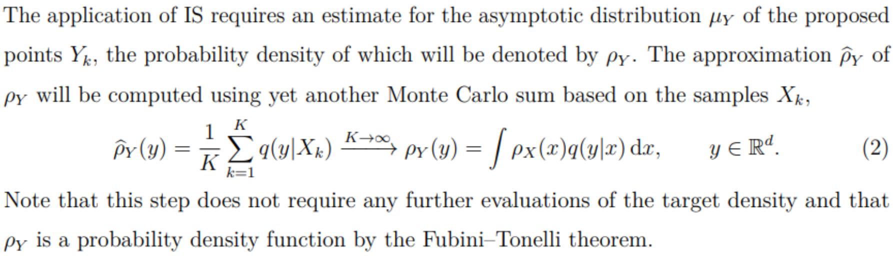
(iii) Set $X_{k+1} = Y_k$ with probability α_k (“accept”) and $X_{k+1} = X_k$ otherwise (“reject”).

Markov Chain Acceptance Rejection

The most prominent class of acceptance rejection methods are MH algorithms, where the choice of the acceptance probability function

$$\alpha(x,y) = \min\left\{1, \frac{q(x|y)\rho(y)}{q(y|x)\rho(x)}\right\}, \qquad x, y \in \mathbb{R}^d, \tag{1}$$

MCMC - IS

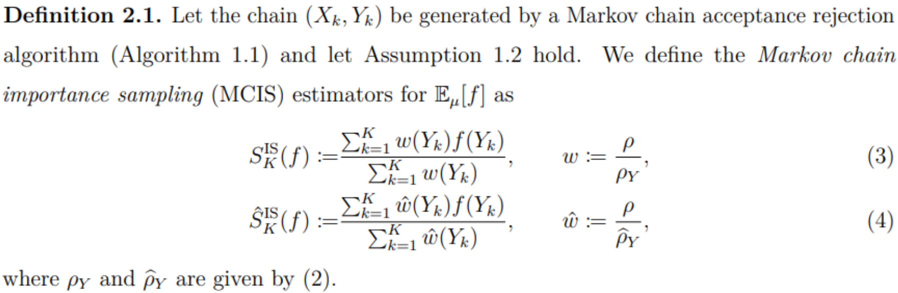


The application of IS requires an estimate for the asymptotic distribution μ_Y of the proposed points Y_k, the probability density of which will be denoted by ρ_Y. The approximation $\widehat{\rho}_Y$ of ρ_Y will be computed using yet another Monte Carlo sum based on the samples X_k,

$$\widehat{\rho}_Y(y) = \frac{1}{K}\sum_{k=1}^{K} q(y|X_k) \xrightarrow{K\to\infty} \rho_Y(y) = \int \rho_X(x)q(y|x)\,\mathrm{d}x, \qquad y \in \mathbb{R}^d. \tag{2}$$

Note that this step does not require any further evaluations of the target density and that ρ_Y is a probability density function by the Fubini–Tonelli theorem.

MCMC - IS



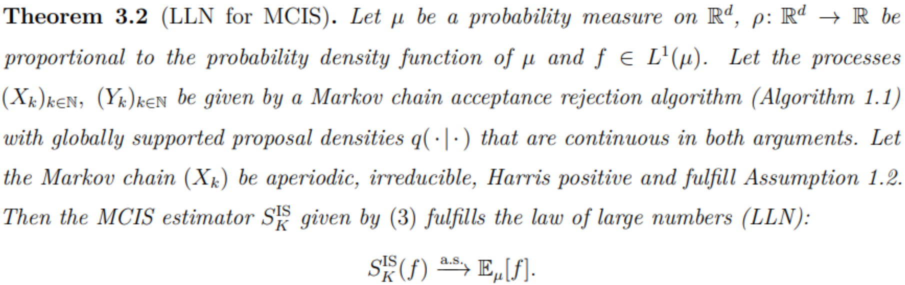
Definition 2.1. Let the chain (X_k, Y_k) be generated by a Markov chain acceptance rejection algorithm (Algorithm 1.1) and let Assumption 1.2 hold. We define the *Markov chain importance sampling* (MCIS) estimators for $\mathbb{E}_\mu[f]$ as

$$S_K^{\mathrm{IS}}(f) := \frac{\sum_{k=1}^K w(Y_k) f(Y_k)}{\sum_{k=1}^K w(Y_k)}, \qquad w := \frac{\rho}{\rho_Y}, \tag{3}$$

$$\hat{S}_K^{\mathrm{IS}}(f) := \frac{\sum_{k=1}^K \hat{w}(Y_k) f(Y_k)}{\sum_{k=1}^K \hat{w}(Y_k)}, \qquad \hat{w} := \frac{\rho}{\widehat{\rho}_Y}, \tag{4}$$

where ρ_Y and $\widehat{\rho}_Y$ are given by (2).

MCMC - IS



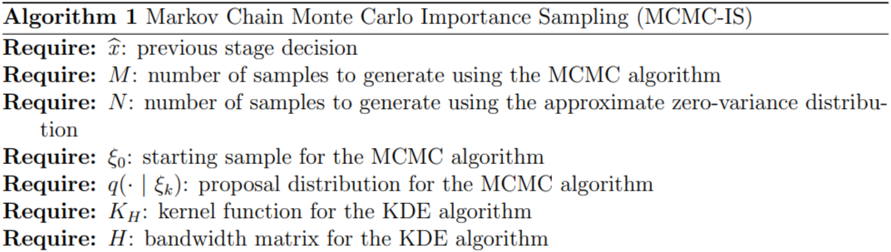
Theorem 3.2 (LLN for MCIS). *Let μ be a probability measure on $\mathbb{R}^d$, $\rho\colon \mathbb{R}^d \to \mathbb{R}$ be proportional to the probability density function of μ and $f \in L^1(\mu)$. Let the processes $(X_k)_{k\in\mathbb{N}}$, $(Y_k)_{k\in\mathbb{N}}$ be given by a Markov chain acceptance rejection algorithm (Algorithm 1.1) with globally supported proposal densities $q(\,\cdot\,|\,\cdot\,)$ that are continuous in both arguments. Let the Markov chain (X_k) be aperiodic, irreducible, Harris positive and fulfill Assumption 1.2. Then the MCIS estimator S_K^{IS} given by (3) fulfills the law of large numbers (LLN):*

$$S_K^{\mathrm{IS}}(f) \xrightarrow{\text{a.s.}} \mathbb{E}_\mu[f].$$

KDE

$$\widehat{f}_h(\mathrm{x}) = \frac{1}{n}\sum_{i=1}^{n} K_h(\mathrm{x} - x_i) = \frac{1}{nh} K\left(\frac{\mathrm{x} - x_i}{h}\right)$$

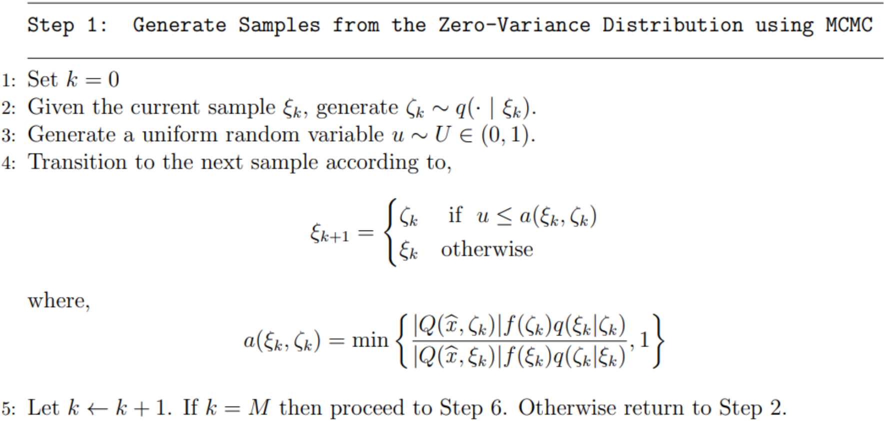
MCMC - IS



Algorithm 1 Markov Chain Monte Carlo Importance Sampling (MCMC-IS)

```
Require: $\widehat{x}$: previous stage decision
Require: $M$: number of samples to generate using the MCMC algorithm
Require: $N$: number of samples to generate using the approximate zero-variance distribution
Require: $\xi_0$: starting sample for the MCMC algorithm
Require: $q(\cdot \mid \xi_k)$: proposal distribution for the MCMC algorithm
Require: $K_H$: kernel function for the KDE algorithm
Require: $H$: bandwidth matrix for the KDE algorithm
```

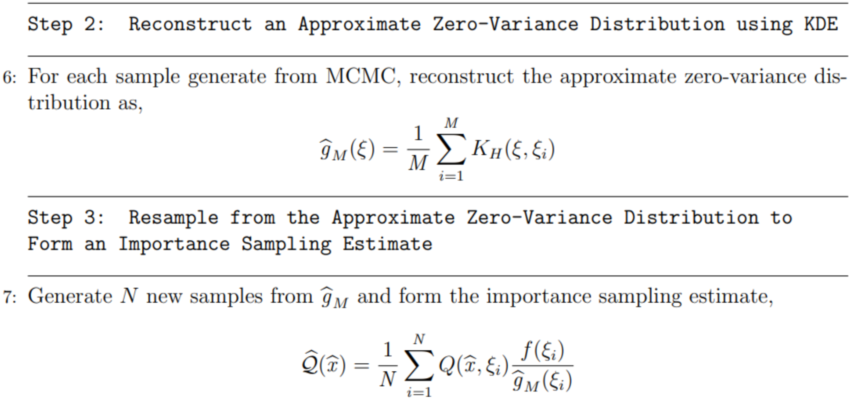
MCMC - IS



Step 1: Generate Samples from the Zero-Variance Distribution using MCMC

```
1: Set $k = 0$
2: Given the current sample $\xi_k$, generate $\zeta_k \sim q(\cdot \mid \xi_k)$.
3: Generate a uniform random variable $u \sim U \in (0, 1)$.
4: Transition to the next sample according to,

       $$\xi_{k+1} = \begin{cases} \zeta_k & \text{if } \ u \leq a(\xi_k, \zeta_k) \\ \xi_k & \text{otherwise} \end{cases}$$

   where,

       $$a(\xi_k, \zeta_k) = \min \left\{ \frac{|Q(\widehat{x}, \zeta_k)| f(\zeta_k) q(\xi_k|\zeta_k)}{|Q(\widehat{x}, \xi_k)| f(\xi_k) q(\zeta_k|\xi_k)}, 1 \right\}$$

5: Let $k \leftarrow k + 1$. If $k = M$ then proceed to Step 6. Otherwise return to Step 2.
```

MCMC - IS



Step 2: Reconstruct an Approximate Zero-Variance Distribution using KDE

6: For each sample generate from MCMC, reconstruct the approximate zero-variance distribution as,

$$\widehat{g}_M(\xi) = \frac{1}{M}\sum_{i=1}^{M} K_H(\xi, \xi_i)$$

Step 3: Resample from the Approximate Zero-Variance Distribution to Form an Importance Sampling Estimate

7: Generate N new samples from $\widehat{g}_M$ and form the importance sampling estimate,

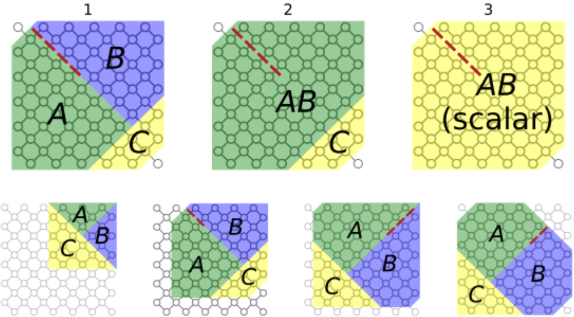
$$\widehat{Q}(\widehat{x}) = \frac{1}{N}\sum_{i=1}^{N} Q(\widehat{x}, \xi_i)\frac{f(\xi_i)}{\widehat{g}_M(\xi_i)}$$

Limitations

- In the case of MCMC-IS, there is no known advantage or convergence to the case of the Porter-Thomas distribution.
- In the case of Frugal Rejection Sampling, there is a part studied about the Porter-Thomas Distribution, and above all, it is a method adopted by the Google Sycamore Team for sampling.
- However, for frugal sampling, the overhead has many disadvantages compared to MCMC. To solve this, [Villalonga, B. et al, A flexible high-performance simulator for the verification and benchmarking of quantum circuits implemented on real hardware] was referred to.

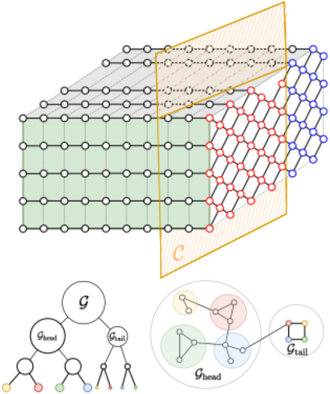
Modified Frugal Rejction Sampling

In AB, just sample 10^6 as is. But in C, we randomly choose N_C number of bitstrings. Then use each N_C strings as one batch. If we modifying acceptance process, acceptance probability will be $1-(1-1/M)^{N_c}$, instead of N_C. There will be optimal point of N_C.



Head Tail Method

Feng Pan and Pan Zhang, “Simulation of Quantum Circuits Using the Big-Batch Tensor Network Method,”


C
$\mathcal{G}$
$\mathcal{G}_{\text{head}}$
$\mathcal{G}_{\text{tail}}$
$\mathcal{G}_{\text{tail}}$
$\mathcal{G}_{\text{head}}$

Sparse State Method

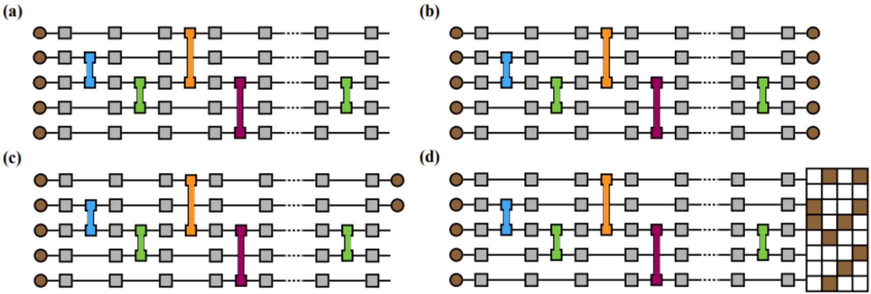


FIG. 4. Four different types of quantum circuit simulation strategies by contracting the corresponding tensor network. (a) Full amplitude simulation. (b) Single amplitude simulation. (c) Batch simulation. (d) Sparse-state simulation.

Sparse State Method

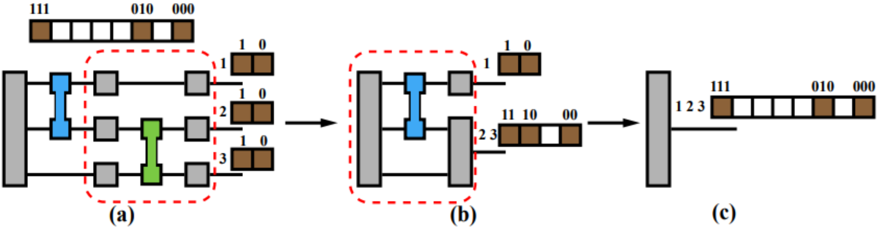


FIG. 5. An example to illustrate the sparse-state simulation.