Ensuring Rapid Mixing and Low Bias for Asynchronous Gibbs Sampling

Christopher De Sa, Kunle Olukotun, and Christopher Ré

cdesa@stanford.edu, kunle@stanford.edu, chrismre@cs.stanford.edu

Departments of Electrical Engineering and Computer Science, Stanford University

Overview

Everyone uses Gibbs sampling!

- De facto Markov Chain Monte Carlo (MCMC) method for inference.
- Used for learning in graphical models.
- Works very well in practice.
- Used by ***many systems*** such as Factorie, OpenBugs, PGibbs, and DeepDive — including competition-winners.

It's important for Gibbs sampling to run fast!

- Modern hardware (CPU, GPU, FPGA) is ***parallel***, with many computations running at the same time.

- Gibbs sampling is inherently ***sequential*** — the updates must happen one at a time.

HOGWILD!: Just parallelize asynchronously

- Run multiple threads in parallel ***without locks***.
- We call this ***asynchronous execution***.
- The idea comes from stochastic gradient descent (SGD) — Niu et al 2011.
- Very successful for SGD — with guarantees!

Asynchronous Gibbs Sampling

- Known to get good parallel scaling
- Actually ***used by practitioners***
- ...but ***no theoretical guarantees*** were known

How can we know asynchronous Gibbs works?

- Bound the ***sample bias*** — how far are the samples produced by the chain from the target distribution?
- Bound the ***mixing time*** — how long do we need to run the chain before we are independent of initial conditions?

Folklore says that both of these quantities are not affected too much by asynchronicity.

- Intuition borrowed from SGD, where asynchronicity provably has little effect.
- But is this actually true?

Our Contributions

The folklore *is not necessarily true*

- We show cases where running asynchronously can ***greatly affect*** sample bias and mixing time.

We provide *guaranteed bounds* on both the sample bias and the mixing time

- Using a reasonable restriction on the distribution
- Captures models encountered in practice

What is Gibbs Sampling?

Goal: produce samples from some distribution π

- Typically, it's ***too hard*** to compute π directly.
- It's easy to compute ***conditional distributions***.

Gibbs sampling: Sample from distribution π

Require: Initial state X_i for $i \in \{1, \ldots, n\}$
loop
  Select a variable i uniformly from $\{1, \ldots, n\}$.
  Re-sample X_i from its conditional distribution in π given the other variables $X_{\{1,\ldots,n\}\setminus\{i\}}$.
  Output sample X.
end loop

Modeling Asynchronicity

When we read a variable, it could be *stale*

- No locking → updates based on old values.
- This leads to ***race conditions***.
- Unbounded staleness → algorithm won't necessarily make progress.

Standard assumption: *bound the staleness*

- Define ***parameter*** τ: number of writes between when a variable was read and when it was used.
- τ models everything relevant about the hardware: number of threads, cache properties, etc.
- ***Standard technique*** to analyze HOGWILD! SGD.

Standard Metrics

Total variation distance.

- Used to measure convergence of MCMC.

Let μ and ν be two distributions on the same space. The ***total variation distance*** between them is

$$\|\mu - \nu\|_{TV} = \max_A |\mu(A) - \nu(A)|,$$

where A is any event in the space.

Total influence α of a model.

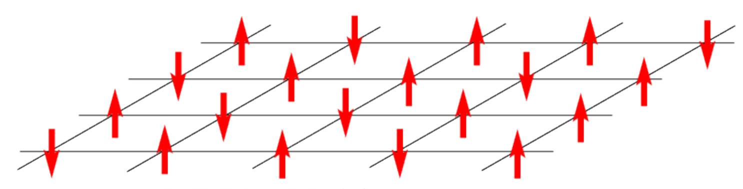
- Measures the degree to which one variable can depend on the other variables in the model.
- Maximum degree is an upper bound for α.

Let π be a probability distribution over some set of variables I. Let B_j be the set of state pairs (X, Y) which differ ***only at variable*** j. Let $\pi_i(\cdot|X_{I\setminus\{i\}})$ denote the ***conditional distribution*** in π of variable i given all the other variables in state X. Then α, the ***total influence*** of π, is

$$\alpha = \max_{i \in I} \sum_{j \in I} \max_{(X,Y) \in B_j} \left\|\pi_i(\cdot|X_{I\setminus\{i\}}) - \pi_i(\cdot|Y_{I\setminus\{i\}})\right\|_{TV}.$$

Model satisfies ***Dobrushin's condition*** if $\alpha < 1$.

- Condition that ensures the rapid mixing of spin statistics systems.



2–D Ising Model

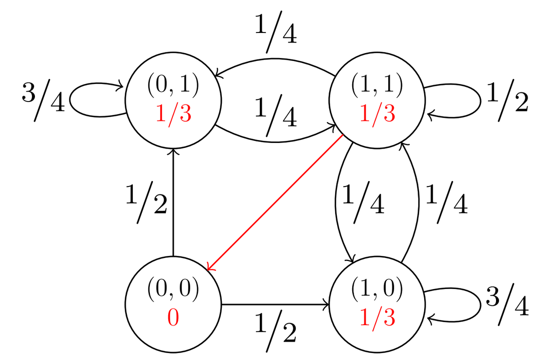
Bounding the Sample Bias

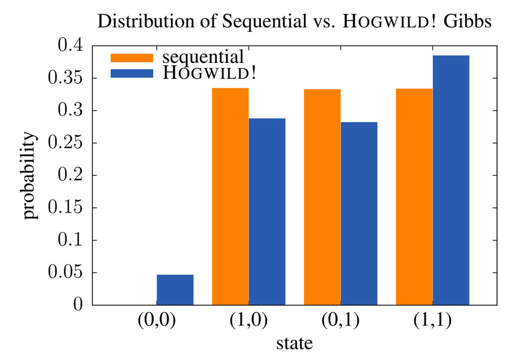
Known result: sequential Gibbs sampling always approaches the target distribution over time → ***no bias***.

Asynchronous Gibbs sampling can have ***asymptotic bias***!

- Consider example below, with two binary variables.
- We plot the results of 2-thread asynchronous Gibbs on this model — 9.8% of the mass is measured erroneously!

$$p(0,1) = p(1,0) = p(1,1) = \frac{1}{3} \qquad p(0,0) = 0.$$





Our contribution: bounds on sample bias.

- Measure with new metric: ***sparse variation distance***
- For ***marginal estimation***, sparse variation distance is what we really care about.

Let μ and ν be two distributions on the same space. The ***ω-sparse variation distance*** between them is

$$\|\mu - \nu\|_{SV(\omega)} = \max_{|A| \le \omega} |\mu(A) - \nu(A)|,$$

where $|A|$ is number of variables on which event A depends.

For a model which satisfies ***Dobrushin's condition*** ($\alpha < 1$), the ***asymptotic bias is bounded by***

$$\lim_{t \to \infty} \left\|P^{(t)}\mu_0 - \pi\right\|_{SV(\omega)} \le \frac{\alpha\tau\omega}{(1-\alpha)n}.$$

Even if $\alpha \ge 1$, as long as $\alpha = O(1)$ and only $O(n)$ steps of sequential Gibbs are required to get good marginal estimates, we can get the following ***asymptotic bound***.

$$\lim_{t \to \infty} \left\|P^{(t)}\mu_0 - \pi\right\|_{SV(\omega)} = O(\tau\omega/n).$$

- Roughly: if sequential Gibbs gets fast estimates, then asynchronous Gibbs has small bias.
- More details are in the paper.

Bounding the Mixing Time

The ***mixing time*** t_{mix} is the number of steps required to be close to independent of initial conditions.

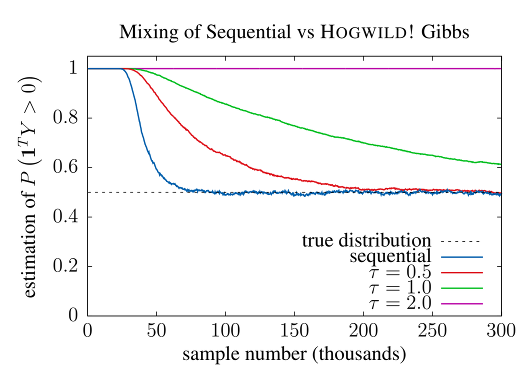
- We need t_{mix} small for Gibbs sampling to be tractable.

The ***mixing time*** of a process with distribution $P^{(t)}\mu_0$ at time t starting from from distribution μ_0 is

$$t_{mix} = \min\left\{ t \,\middle|\, \forall \mu_0 \left\|P^{(t)}\mu_0 - P^{(t)}\pi\right\|_{TV} \le \frac{1}{4} \right\}.$$

Asynchronicity can ***affect the mixing time***!

- See example to the right.
- Even models with $t_{mix} = \tilde{O}(n)$ for sequential Gibbs could have $t_{mix} = 2^{\Omega(n)}$ for asynchronous Gibbs!



Our contribution: bounds on the mixing time.

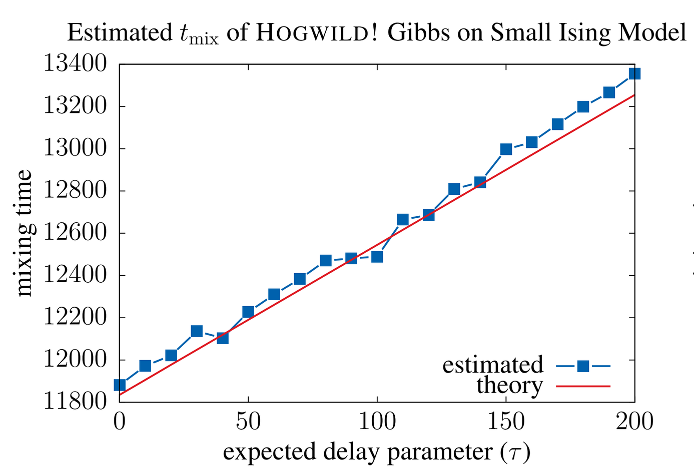
- If model satisfies ***Dobrushin's condition*** ($\alpha < 1$), there's a known bound on the mixing time of sequential Gibbs.
- We can also prove a bound for HOGWILD! Gibbs.

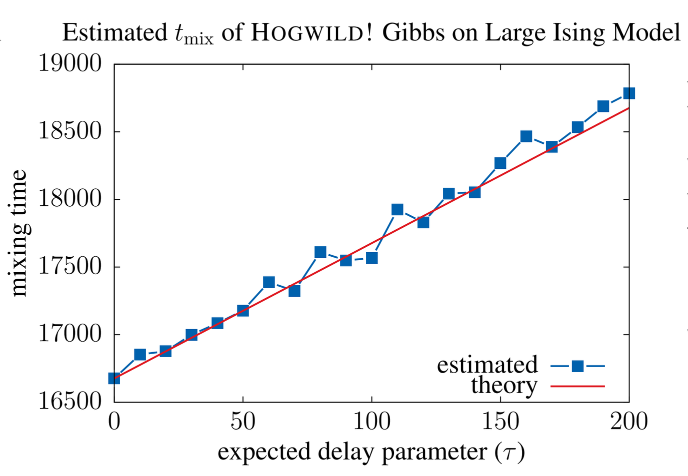
$$t_{mix-seq} \le \frac{n}{1-\alpha}\log(4n) \qquad t_{mix-hog} \le \frac{n+\alpha\tau}{1-\alpha}\log(4n).$$

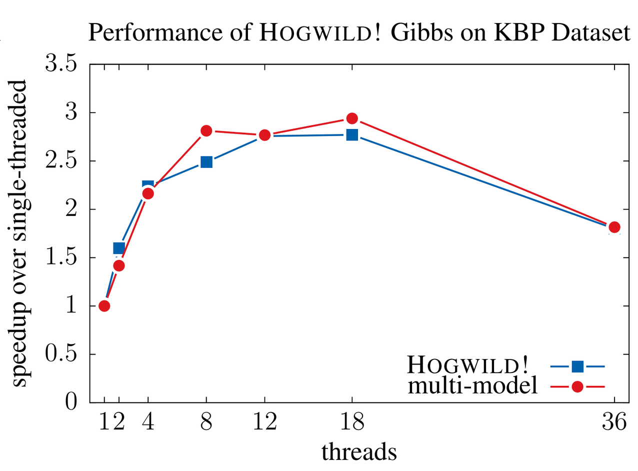
Mixing times are about the same!

- Predicted relationship: $\dfrac{t_{mix-hog}}{t_{mix-seq}} \approx 1 + \dfrac{\alpha\tau}{n}$.
- HOGWILD! ***runs much faster*** on hardware.

Experiments







The first two plots show that the experimentally observed mixing times of HOGWILD! Gibbs sampling on two different Ising model graphs ***match our theoretical predictions***. The third plot shows wall-clock performance of asynchronous Gibbs on a real KBP dataset, and compares it to another method, "multi-model" Gibbs, which has similar runtime but ***produces lower-quality samples***.