Ensuring Rapid Mixing and Low Bias for Asynchronous Gibbs Sampling

Christopher De Sa CDESA@STANFORD.EDU
Department of Electrical Engineering, Stanford University, Stanford, CA 94309

Kunle Olukotun KUNLE@STANFORD.EDU
Department of Electrical Engineering, Stanford University, Stanford, CA 94309

Christopher Ré CHRISMRE@STANFORD.EDU
Department of Computer Science, Stanford University, Stanford, CA 94309

Abstract

Gibbs sampling is a Markov chain Monte Carlo technique commonly used for estimating marginal distributions. To speed up Gibbs sampling, there has recently been interest in parallelizing it by executing asynchronously. While empirical results suggest that many models can be efficiently sampled asynchronously, traditional Markov chain analysis does not apply to the asynchronous case, and thus asynchronous Gibbs sampling is poorly understood. In this paper, we derive a better understanding of the two main challenges of asynchronous Gibbs: bias and mixing time. We show experimentally that our theoretical results match practical outcomes.

1. Introduction

Gibbs sampling is one of the most common Markov chain Monte Carlo methods used with graphical models (Koller & Friedman, 2009). In this setting, Gibbs sampling (Algorithm 1) operates iteratively by choosing at random a variable from the model at each timestep, and updating it by sampling from its conditional distribution given the other variables in the model. Often, it is applied to inference problems, in which we are trying to estimate the marginal probabilities of some query events in a given distribution.

For sparse graphical models, to which Gibbs sampling is often applied, each of these updates needs to read the values of only a small subset of the variables; therefore each update can be computed very quickly on modern hardware. Because of this and other useful properties of Gibbs sampling, many systems use Gibbs sampling to perform inference on big data (Lunn et al., 2009; McCallum et al., 2009; Newman et al., 2007; Smola & Narayanamurthy, 2010; Theis et al., 2012; Zhang & Ré, 2014).

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

Algorithm 1 Gibbs sampling

Require: Variables x_i for $1 \leq i \leq n$, and distribution π.
  for $t = 1$ **to** T **do**
    Sample s uniformly from $\{1, \ldots, n\}$.
    Re-sample x_s uniformly from $\mathbf{P}_\pi(X_s | X_{\{1,\ldots,n\}\setminus\{s\}})$.
  end for

Since Gibbs sampling is such a ubiquitous algorithm, it is important to try to optimize its execution speed on modern hardware. Unfortunately, while modern computer hardware has been trending towards more parallel architectures (Sutter, 2005), traditional Gibbs sampling is an inherently sequential algorithm; that is, the loop in Algorithm 1 is not directly parallelizable. Furthermore, for sparse models, very little work happens within each iteration, meaning it is difficult to extract much parallelism from the body of this loop. Since traditional Gibbs sampling parallelizes so poorly, it is interesting to study variants of Gibbs sampling that can be parallelized. Several such variants have been proposed, including applications to latent Dirichlet allocation (Newman et al., 2007; Smola & Narayanamurthy, 2010) and distributed constraint optimization problems (Nguyen et al., 2013).

In one popular variant, multiple threads run the Gibbs sampling update rule in parallel without locks, a strategy called *asynchronous* or HOGWILD! execution—in this paper, we use these two terms interchangeably. This idea was proposed, but not analyzed theoretically, in Smola & Narayanamurthy (2010), and has been shown to give empirically better results on many models (Zhang & Ré, 2014). But when can we be sure that HOGWILD! Gibbs sampling will produce accurate results? Except for the case of Gaussian random variables (Johnson et al., 2013), there

is no existing analysis by which we can ensure that asynchronous Gibbs sampling will be appropriate for a particular application. Even the problems posed by HOGWILD!-Gibbs are poorly understood, and their solutions more so.

As we will show in the following sections, there are two main issues when analyzing asynchronous Gibbs sampling. Firstly, we will show by example that, surprisingly, HOGWILD!-Gibbs can be *biased*—unlike sequential Gibbs, it does not always produce samples that are arbitrarily close to the target distribution. Secondly, we will show that the *mixing time* (the time for the chain to become close to its stationary distribution) of asynchronous Gibbs sampling can be up to exponentially greater than that of the corresponding sequential chain.

To address the issue of bias, we need some way to describe the distance between the target distribution π and the distribution of the samples produced by HOGWILD!-Gibbs. The standard notion to use here is the *total variation distance*, but for the task of computing marginal probabilities, it gives an overestimate on the error caused by bias. To better describe the bias, we introduce a new notion of statistical distance, the *sparse variation distance*. While this relaxed notion of statistical distance is interesting in its own right, its main benefit here is that it uses a more local view of the chain to more tightly measure the effect of bias.

Our main goal is to identify conditions under which the bias and mixing time of asynchronous Gibbs can be bounded. One parameter that has been used to great effect in the analysis of Gibbs sampling is the *total influence* α of a model. The total influence measures the degree to which the marginal distribution of a variable can depend on the values of the other variables in the model—this parameter has appeared as part of a celebrated line of work on *Dobrushin's condition* ($\alpha < 1$), which ensures the rapid mixing of spin statistics systems (Dobrushin, 1956; Dyer et al., 2006; Hayes, 2006). It turns out that we can use this parameter to bound both the bias and mixing time of HOGWILD!-Gibbs, and so we make the following contributions:

- We describe a way to statistically model the asynchronicity in HOGWILD!-Gibbs sampling.
- To bound the bias, we prove that for classes of models with bounded total influence $\alpha = O(1)$, if sequential Gibbs sampling achieves small sparse variation distance to π in $O(n)$ steps, where n is the number of variables, then HOGWILD!-Gibbs samples achieve the same distance in at most $O(1)$ more steps.
- For models that satisfy Dobrushin's condition (that is, $\alpha < 1$), we show that the mixing time bounds of sequential and HOGWILD!-Gibbs sampling differ only by a factor of $1 + O(n^{-1})$.
- We validate our results experimentally and show that, by using asynchronous execution, we can achieve wall-clock speedups of up to $2.8\times$ on real problems.

2. Related Work

Much work has been done on the analysis of parallel Gibbs samplers. One simple way to parallelize Gibbs sampling is to run multiple chains independently in parallel: this heuristic uses parallelism to produce more samples overall, but does not produce accurate samples more quickly. Additionally, this strategy is sometimes worse than other strategies on a systems level (Smola & Narayanamurthy, 2010; Zhang & Ré, 2014), typically because it requires additional memory to maintain multiple models of the chain. Another strategy for parallelizing Gibbs sampling involves taking advantage of the structure of the underlying factor graph to run in parallel while still maintaining an execution pattern to which the standard sequential Gibbs sampling analysis can be applied (Gonzalez et al., 2011). Much further work has focused on parallelizing sampling for specific problems, such as LDA (Newman et al., 2007; Smola & Narayanamurthy, 2010) and others (Nguyen et al., 2013).

Our approach follows on the paper of Johnson et al. (2013), which named the HOGWILD!-Gibbs sampling algorithm and analyzed it for Gaussian models. Their main contribution is an analysis framework that includes a sufficient condition under which HOGWILD! Gaussian Gibbs samples are guaranteed to have the correct asymptotic mean. Recent work (Terenin et al., 2015) has analyzed a similar algorithm under even stronger regularity conditions. Here, we seek to give more general results for the analysis of HOGWILD!-Gibbs sampling on discrete-valued factor graphs.

The HOGWILD!-Gibbs sampling algorithm was inspired by a line of work on parallelizing stochastic gradient descent (SGD) by running it asynchronously. HOGWILD! SGD was first proposed by Niu et al. (2011), who proved that while running without locks causes race conditions, they do not significantly impede the convergence of the algorithm. The asynchronous execution strategy has been applied to many problems—such as PageRank approximations (Mitliagkas et al., 2015), deep learning (Noel & Osindero, 2014) and recommender systems (Yu et al., 2012)—so it is not surprising that it has been proposed for use with Gibbs sampling. Our goal in this paper is to combine analysis ideas that have been applied to Gibbs sampling and HOGWILD!, in order to characterize the behavior of asynchronous Gibbs. In particular, we are motivated by some recent work on the analysis of HOGWILD! for SGD (De Sa et al., 2015b; Liu & Wright, 2015; Liu et al., 2015; Mania et al., 2015). Several of these results suggest modeling the race conditions inherent in HOGWILD! SGD as noise in a stochastic process; this lets them bring a trove of statistical techniques to bear on the analysis of HOGWILD! SGD.

Therefore, in this paper, we will apply a similar stochastic process model to Gibbs sampling.

Several recent papers have focused on the mixing time of Gibbs sampling based on the structural properties of the model. Gotovos et al. (2015) and De Sa et al. (2015a) each show that Gibbs sampling mixes in polynomial time for a class of distributions bounded by some parameter. Unfortunately, these results both depend on *spectral methods* (that try to bound the spectral gap of the Markov transition matrix), which are difficult to apply to HOGWILD! Gibbs sampling for two reasons. First, spectral methods don't let us represent the sampler as a stochastic process, which limits the range of techniques we can use to model the noise. Secondly, while most spectral methods only apply to *reversible* Markov chains—and sequential Gibbs sampling is always a reversible chain—for HOGWILD!-Gibbs sampling the asynchronicity and parallelism make the chain non-reversible. Because of this, we were unable to use these spectral results in our asynchronous setting. We are forced to rely on the other method (Guruswami, 2000) for analyzing Markov processes, *coupling*—the type of analysis used with the Dobrushin condition—which we will describe in the following sections.

3. Modeling Asynchronicity

In this section, we describe a statistical model for asynchronous Gibbs sampling by adapting the hardware model outlined in De Sa et al. (2015b). Because we are motivated by the factor graph inference problem, we will focus on the case where the distribution π that we want to sample comes from a sparse, discrete graphical model.

Any HOGWILD!-Gibbs implementation involves some number of threads each repeatedly executing the Gibbs update rule on a single copy of the model (typically stored in RAM). We assume that this model serializes all writes, such that we can speak of the state of the system after t writes have occurred. We call this time t, and we will model the HOGWILD! system as a stochastic process adapted to the natural filtration $\mathcal{F}_t$. Here, $\mathcal{F}_t$ contains all events that have occurred up to time t, and we say an event is $\mathcal{F}_t$ *measurable* if it is known deterministically by time t.

We begin our construction by letting $x_{i,t}$ denote the ($\mathcal{F}_t$ measurable) value of variable i at time t, and letting $\tilde{I}_t$ be the ($\mathcal{F}_{t+1}$ measurable) index of the variable that we choose to sample at time t. For Gibbs sampling, we have

$$\forall i \in \{1, \ldots, n\},\ \mathbf{P}\left(\tilde{I}_t = i \middle| \mathcal{F}_t\right) = \frac{1}{n};$$

this represents the fact that we have an equal probability of sampling each variable.

Now that we have defined which variables are to be sampled, we proceed to describe how they are sampled. For HOGWILD!-Gibbs sampling, we must model the fact that the sampler does not get to use exactly the values of $x_{i,t}$; rather it has access to a cache containing potentially *stale* values. To do this, we define ($\mathcal{F}_{t+1}$ measurable) $\tilde{v}_{i,t} = x_{i,t-\tilde{\tau}_{i,t}}$, where $\tilde{\tau}_{i,t} \geq 0$ is a *delay parameter* ($\mathcal{F}_{t+1}$ measurable and independent of $\tilde{I}_t$) that represents how old the currently-cached value for variable i could be. A variable resampled using this stale data would have distribution

$$\mathbf{P}\left(\tilde{z}_{i,t} = z | \mathcal{F}_t\right) \propto \pi(\tilde{v}_{1,t}, \ldots, \tilde{v}_{i-1,t}, z, \tilde{v}_{i+1,t}, \ldots, \tilde{v}_{n,t}).$$

Using this, we can relate the values of the variables across time with

$$x_{i,t+1} = \begin{cases} \tilde{z}_{i,t} & \text{if } i = \tilde{I}_t \\ x_{i,t} & \text{otherwise.} \end{cases}$$

So far, our model is incompletely specified, because we have not described the distribution of the delays $\tilde{\tau}_{i,t}$. Unfortunately, since these delays depend on the number of threads and the specifics of the hardware (Niu et al., 2011), their distribution is difficult to measure. Instead of specifying a particular distribution, we require only a bound on the expected delay, $\mathbf{E}\left[\tilde{\tau}_{i,t} | \mathcal{F}_t\right] \leq \tau$. In this model, the τ parameter represents everything that is relevant about the hardware; representing the hardware in this way has been successful for the analysis of asynchronous SGD (Niu et al., 2011), so it is reasonable to use it for Gibbs sampling. In addition to this, we will need a similar parameter that bounds the tails of $\tilde{\tau}_{i,t}$ slightly more aggressively. We require that for some parameter τ^*, and for all i and t,

$$\mathbf{E}\left[\exp\left(n^{-1}\tilde{\tau}_{i,t}\right) \middle| \mathcal{F}_t\right] \leq 1 + n^{-1}\tau^*.$$

This parameter is typically very close to the expected value bound τ; in particular, as n approaches infinity, τ^* approaches τ.

4. The First Challenge: Bias

Perhaps the most basic result about sequential Gibbs sampling is the fact that, in the limit of large numbers of samples, it is unbiased. In order to measure convergence of Markov chains to their stationary distribution, it is standard to use the total variation distance.

Definition 1 (Total Variation Distance)**.** The *total variation distance* (Levin et al., 2009, p. 48) between two probability measures μ and ν on probability space Ω is defined as

$$\|\mu - \nu\|_{\text{TV}} = \max_{A \subset \Omega} |\mu(A) - \nu(A)|,$$

that is, the maximum difference between the probabilities that μ and ν assign to a single event A.

It is a well-known result that, for Gibbs sampling on a strictly-positive target distribution π, it will hold that

$$\lim_{t\to\infty} \left\| P^{(t)}\mu_0 - \pi \right\|_{\mathrm{TV}} = 0, \tag{1}$$

where $P^{(t)}\mu_0$ denotes the distribution of the t-th sample.

One of the difficulties that arises when applying HOGWILD! to Gibbs sampling is that the race conditions from the asynchronous execution add bias to the samples — Equation 1 no longer holds. To understand why, we can consider a simple example.

4.1. Bias Example

Consider a simple model with two variables X_1 and X_2 each taking on values in $\{0, 1\}$, and having distribution

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

Sequential Gibbs sampling on this model will produce unbiased samples from the target distribution. Unfortunately, this is not the case if we run HOGWILD!-Gibbs sampling on this model. Assume that the state is currently $(1, 1)$ and two threads, T_1 and T_2, simultaneously update X_1 and X_2 respectively. Since T_1 reads state $(1, 1)$ it will update X_1 to 0 or 1 each with probability 0.5; the same will be true for T_2 and X_2. Therefore, after this happens, every state will have probability 0.25; this includes the state $(0, 0)$ which should never occur! Over time, this race condition will produce samples with value $(0, 0)$ with some non-zero frequency; this is an example of *bias* introduced by the HOGWILD! sampling. Worse, this bias is not just theoretical: Figure 1 illustrates how the measured distribution for this model is affected by two-thread asynchronous execution. In particular, we observe that almost 5% of the mass is erroneously measured to be in the state $(0, 0)$, which has no mass at all in the true distribution. The total variation distance to the target distribution is quite large at 9.8%, and, unlike in the sequential case, this bias doesn't disappear as the number of samples goes to infinity.

4.2. Bounding the Bias

The previous example has shown that asynchronous Gibbs sampling will not necessarily produce a sequence of samples arbitrarily close to the target distribution. Instead, the samples may approach some other distribution, which we hope is sufficiently similar for some practical purpose. Often, the purpose of Gibbs sampling is to estimate the marginal distributions of individual variables or of events that each depend on only a small number of variables in the model. To characterize the accuracy of these estimates, the total variation distance is *too conservative*: it depends on the difference over all the events in the space, when most

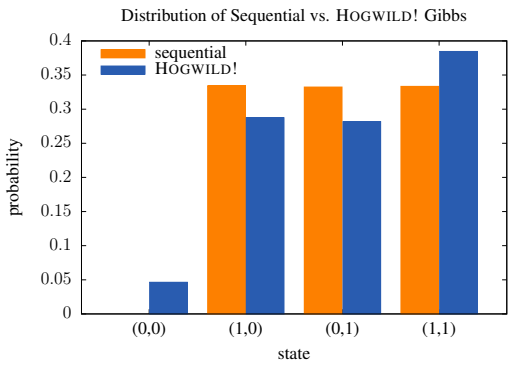

Figure 1. Bias introduced by HOGWILD!-Gibbs (10^6 samples).

of these are events that we do not care about. To address this, we introduce the following definition.

Definition 2 (Sparse Variation Distance)**.** For any event A in a probability space Ω over a set of variables V, let $|A|$ denote the number of variables upon which A depends. Then, for any two distributions μ and ν over Ω, we define the *ω-sparse variation distance* to be

$$\|\mu - \nu\|_{\mathrm{SV}(\omega)} = \max_{|A|\leq\omega} |\mu(A) - \nu(A)| .$$

For the wide variety of applications that use sampling for marginal estimation, the sparse variation distance measures the quantity we actually care about: the maximum possible bias in the marginal distribution of the samples. As we will show, asynchronous execution seems to have less effect on the sparse variation distance than the total variation distance, because sparse variation distance uses a more localized view of the chain. For example, in Figure 1, the total variation distance between the sequential and HOGWILD! distributions is 9.8%, while the 1-sparse variation distance is only 0.4%. That is, while HOGWILD! execution does introduce great bias into the distribution, it still estimates marginals of the individual variables accurately.

This definition suggests the question: how long do we have to run before our samples have low sparse variation distance from the target distribution? To answer this question, we introduce the following definition.

Definition 3 (Sparse Estimation Time)**.** The *ω-sparse estimation time* of a stochastic sampler with distribution $P^{(t)}\mu_0$ at time t and target distribution π is the first time t at which, for any initial distribution μ_0, the estimated distribution is within sparse variation distance ϵ of π,

$$t_{\mathrm{SE}(\omega)}(\epsilon) = \min\{t \in \mathbb{N} \mid \forall\mu_0,\ \|P^{(t)}\mu_0 - \pi\|_{\mathrm{SV}(\omega)} \leq \epsilon\}.$$

In many practical systems (Neubig, 2014; Shin et al., 2015), Gibbs sampling is used without a proof that it works;

instead, it is naively run for some fixed number of passes through the dataset. This naive strategy works for models for which accurate marginal estimates can be achieved after $O(n)$ samples. This $O(n)$ runtime is necessary for Gibbs sampling to be feasible on big data, meaning roughly that these are the models which it is interesting to try to speed up using asynchronous execution. Therefore, for the rest of this section, we will focus on the bias of the HOGWILD! chain for this class of models. When analyzing Gibbs sampling, we can bound the bias within the context of a coupling argument using a parameter called the *total influence*. While we arrived at this condition independently, it has been studied before, especially in the context of *Dobrushin's condition*, which ensures rapid mixing of Gibbs sampling.

Definition 4 (Total Influence)**.** 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, define α, the total influence of π, as

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

We say the model satisfies Dobrushin's condition if $\alpha < 1$.

One way to think of total influence for factor graphs is as a generalization of maximum degree; indeed, if a factor graph has maximum degree Δ, it can easily be shown that $\alpha \leq \Delta$. It turns out that if we can bound both this parameter and the sparse estimation time of sequential Gibbs sampling, we can give a simple bound on the sparse estimation time for asynchronous Gibbs sampling.

Claim 1. *Assume that we have a class of distributions with bounded total influence $\alpha = O(1)$. For each distribution π in the class, let $\bar{t}_{\mathrm{SE-seq}(\omega)}(\pi, \epsilon)$ be an upper bound on the ω-sparse estimation time of its sequential Gibbs sampler, and assume that it is a convex, decreasing function of ϵ. Further assume that, for any ϵ, across all models,*

$$\bar{t}_{\mathrm{SE-seq}(\omega)}(\pi, \epsilon) = O(n),$$

where n is the number of variables in the model. Then, for any ϵ, the sparse estimation time of HOGWILD!*-Gibbs across all models is bounded by*

$$t_{\mathrm{SE-hog}(\omega)}(\pi, \epsilon) \leq \bar{t}_{\mathrm{SE-seq}(\omega)}(\pi, \epsilon) + O(1).$$

Roughly, this means that HOGWILD!-Gibbs sampling "works" on all problems for which we know marginal estimation is "fast" and the total influence is bounded. Since the sparse estimation times here are measured in iterations, and the asynchronous sampler is able, due to parallelism, to run many more iterations in the same amount of wall clock time, this result implies that HOGWILD!-Gibbs can be much faster than sequential Gibbs for producing estimates of similar quality. To prove Claim 1, and more explicitly bound the bias, we use the following lemma.

Lemma 1. *Assume that we run* HOGWILD!*-Gibbs sampling on a distribution π with total influence α. Let $P^{(t)}_{\mathrm{hog}}$ denote the transition matrix of* HOGWILD!*-Gibbs and $P^{(t)}_{\mathrm{seq}}$ denote the transition matrix of sequential Gibbs. Then for any initial distribution μ_0 and for any t,*

$$\left\|P^{(t)}_{\mathrm{hog}}\mu_0 - P^{(t)}_{\mathrm{seq}}\mu_0\right\|_{\mathrm{SV}(\omega)} \leq \frac{\omega\alpha\tau t}{n^2}\exp\left(\frac{(\alpha-1)_+}{n}t\right),$$

where $(x)_+$ denotes x if $x > 0$ and 0 otherwise.

This lemma bounds the distance between the distributions of asynchronous and sequential Gibbs; if we let t be the sparse estimation time of sequential Gibbs, we can interpret this distance as an upper bound on the bias. When $t = O(n)$, this bias is $O(n^{-1})$, which has an intuitive explanation: for HOGWILD! execution, race conditions occur about once every $\Theta(n)$ iterations, so the bias is roughly proportional to the frequency of race conditions. This gives us a relationship between the statistical error of the algorithm and a more traditional notion of computational error.

Up until now, we have been assuming that we have a class for which the sparse estimation time is $O(n)$. Using the total influence α, we can identify a class of models known to meet this criterion.

Theorem 1. *For any distribution that satisfies Dobrushin's condition, $\alpha < 1$, the ω-sparse estimation time of the sequential Gibbs sampling process will be bounded by*

$$t_{\mathrm{SE-seq}(\omega)}(\epsilon) \leq \left\lceil \frac{n}{1-\alpha}\log\left(\frac{\omega}{\epsilon}\right)\right\rceil.$$

This surprising result says that, in order to produce good marginal estimates for any model that satisfies Dobrushin's condition, we need only $O(n)$ samples! While we could now use Lemma 1 to bound the sparse estimation time for HOGWILD!-Gibbs, a more direct analysis produces a slightly better result, which we present here.

Theorem 2. *For any distribution that satisfies Dobrushin's condition, $\alpha < 1$, and for any ϵ that satisfies*

$$\epsilon \geq 2\omega\alpha\tau(1-\alpha)^{-1}n^{-1},$$

the ω-sparse estimation time of the HOGWILD! *Gibbs sampling process will be bounded by*

$$t_{\mathrm{SE-hog}(\omega)}(\epsilon) \leq \left\lceil \frac{n}{1-\alpha}\log\left(\frac{\omega}{\epsilon}\right) + \frac{2\omega\alpha\tau}{(1-\alpha)^2\epsilon}\right\rceil.$$

This result gives us a definite class of models for which HOGWILD!-Gibbs sampling is guaranteed to produce accurate marginal estimates quickly.

5. The Second Challenge: Mixing Times

Even though the HOGWILD!-Gibbs sampler produces biased estimates, it is still interesting to analyze how long we need to run it before the samples it produces are independent of its initial conditions. To measure the efficiency of a Markov chain, it is standard to use the *mixing time*.

Definition 5 (Mixing Time)**.** The *mixing time* (Levin et al., 2009, p. 55) of a stochastic process with transition matrix $P^{(t)}$ at time t and target distribution π is the first time t at which, for any initial distribution μ_0, the estimated distribution is within TV-distance ϵ of $P^{(t)}\pi$. That is,

$$t_{\text{mix}}(\epsilon) = \min\left\{t \middle| \forall \mu_0, \left\| P^{(t)}\mu_0 - P^{(t)}\pi \right\|_{\text{TV}} \leq \epsilon \right\}.$$

5.1. Mixing Time Example

As we did with bias, here we construct an example model for which asynchronous execution disastrously increases the mixing time. The model we will construct is rather extreme; we choose this model because simpler, practical models do not seem to exhibit this type of catastrophic increase in the mixing time. We start, for some odd constant N, with N variables $X_1, \ldots, X_N$ all in $\{-1, 1\}$, and one factor with energy

$$\phi_X(X) = -M_1 \left| \mathbf{1}^T X \right|,$$

for some very large energy parameter M_1. The resulting distribution will be almost uniform over all states with $\mathbf{1}^T X \in \{-1, 1\}$. To this model, we add another bank of variables $Y_1, \ldots, Y_N$ all in $\{-1, 1\}$. These variables also have a single associated factor with energy

$$\phi_Y(X, Y) = \begin{cases} \frac{\beta}{N}\left(\mathbf{1}^T Y\right)^2 & \text{if } \left|\mathbf{1}^T X\right| = 1 \\ M_2\left(\mathbf{1}^T Y\right)^2 & \text{if } \left|\mathbf{1}^T X\right| > 1 \end{cases},$$

for parameters β and M_2. Combining these two factors gives us the overall distribution for our model,

$$\pi(X, Y) = \frac{1}{Z}\exp\left(\phi_X(X) + \phi_Y(X, Y)\right),$$

where Z is the constant necessary for this to be a distribution. Roughly, the X dynamics are constructed to regularly "generate" race conditions, while the Y dynamics are chosen to "detect" these race conditions and mix very slowly as a result. This model is illustrated in Figure 2.

We simulated two-thread HOGWILD!-Gibbs on this model, measuring the marginal probability that $\mathbf{1}^T Y > 0$; by symmetry, this event has probability 0.5 in the stationary distribution for both the sequential and asynchronous samplers. Our results, for a model with $N = 2001$, $\beta = 0.3$, $M_1 = 10^{10}$, and $M_2 = 100$, and initial state $X = Y = \mathbf{1}$, are plotted in Figure 3. Notice that, while the sequential sampler achieves the correct marginal probability relatively quickly, the asynchronous samplers take a much longer time to achieve the correct result, even for a relatively small expected delay ($\tau = 0.5$). These results suggest that something catastrophic is happening to the mixing time when we switch from sequential to asynchronous execution — and in fact we can prove this is the case.

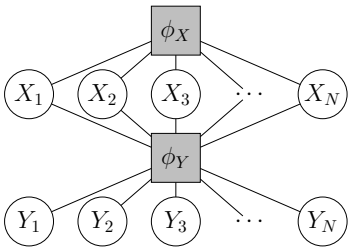

Figure 2. Factor graph model for mixing time example.

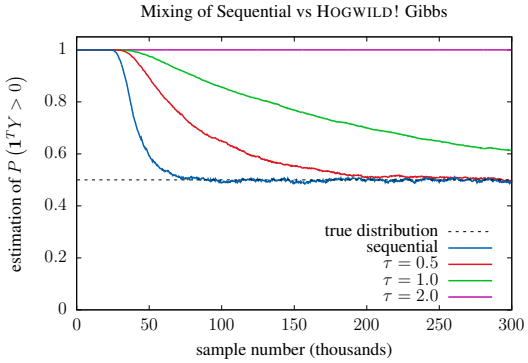

Figure 3. Example wherein asynchronous sampling greatly increases in mixing time. Marginals computed over 10^4 trials.

Statement 1. *For the example model described above, there exist parameters M_1, M_2, and β (as a function of N) such that the mixing time of sequential Gibbs sampling is $O(N \log N)$ but the mixing time of* HOGWILD!*-Gibbs sampling, even with $\tau = O(1)$, can be $\exp(\Omega(N))$.*

The intuition behind this statement is that for sequential Gibbs, the dynamics of the X part of the chain quickly causes it to have $\left|\mathbf{1}^T X\right| = 1$, and then remain there for the remainder of the simulation with high probability. This in turn causes the energy of the ϕ_Y factor to be essentially $\frac{\beta}{N}(\mathbf{1}^T Y)^2$, a model which is known to be fast-mixing because it satisfies Dobrushin's condition. On the other hand, for HOGWILD! Gibbs, due to race conditions we will see $\left|\mathbf{1}^T X\right| \neq 1$ with constant probability; this will cause the effective energy of the ϕ_Y factor to be dominated by the $M_2(\mathbf{1}^T Y)^2$ term, a model that is known to take exponential time to mix.

5.2. Bounding the Mixing Time

This example shows that fast mixing of the sequential sampler alone is not sufficient to guarantee fast mixing of the HOGWILD! chain. Consequently, we look for classes of models for which we can say something about the mixing time of both sequential and HOGWILD!-Gibbs. Dobrushin's condition is well known to imply rapid mixing of sequential Gibbs, and it turns out that we can leverage it again here to bound the mixing time of HOGWILD!-Gibbs.

Theorem 3. *Assume that we run Gibbs sampling on a distribution that satisfies Dobrushin's condition, $\alpha < 1$. Then the mixing time of sequential Gibbs will be bounded by*

$$t_{\mathrm{mix-seq}}(\epsilon) \leq \frac{n}{1-\alpha} \log\left(\frac{n}{\epsilon}\right).$$

Under the same conditions, the mixing time of HOGWILD!*-Gibbs will be bounded by*

$$t_{\mathrm{mix-hog}}(\epsilon) \leq \frac{n + \alpha\tau^*}{1-\alpha} \log\left(\frac{n}{\epsilon}\right).$$

The above example does not contradict this result since it does not satisfy Dobrushin's condition; in fact its total influence is very large and scales with n. We can compare these two mixing time results as

$$t_{\mathrm{mix-hog}}(\epsilon) \approx \left(1 + \alpha\tau^* n^{-1}\right) t_{\mathrm{mix-seq}}(\epsilon); \qquad (2)$$

the bounds on the mixing times differ by a negligible factor of $1 + O(n^{-1})$. This result shows that, for problems that satisfy Dobrusin's condition, HOGWILD!-Gibbs sampling mixes in about the same time as sequential Gibbs sampling, and is therefore a practical choice for generating samples.

5.3. A Positive Example: Ising Model

To gain intuition here, we consider a simple example. The Ising model (Ising, 1925) on a graph $G = (V, E)$ is a model over probability space $\{-1, 1\}^V$, and has distribution

$$p(\sigma) = \frac{1}{Z} \exp\left(\beta \sum_{(x,y)\in E} \sigma(x)\sigma(y) + \sum_{x\in V} B_x \sigma(x)\right),$$

where β is a parameter that is called the *inverse temperature*, the B_x are parameters that encode a *prior* on the variables, and Z is the normalization constant necessary for this to be a distribution. For graphs of maximum degree Δ and sufficiently small β, a bound on the mixing time of Gibbs sampling is known when $\Delta \tanh \beta \leq 1$. It turns out that the total influence of the Ising model can be bounded by $\alpha \leq \Delta \tanh \beta$, and so this condition is simply another way of writing Dobrushin's condition. We can therefore apply Theorem 3 to bound the mixing time of HOGWILD!-Gibbs with

$$t_{\mathrm{mix}}(\epsilon) \leq \frac{n + \tau^* \Delta \tanh \beta}{1 - \Delta \tanh \beta} \log\left(\frac{n}{\epsilon}\right).$$

This illustrates that the class of graphs we are considering includes some common, well-studied models.

5.4. Proof Outline

Here, we briefly describe the technique used to prove Theorem 3; for ease of presentation, we focus on the case where every variable takes on values in $\{-1, 1\}$. We start by introducing the idea of a coupling-based argument (Levin et al., 2009, p. 64), which starts by constructing two copies of the same Markov chain, X and $\bar{X}$, starting from different states but running together in the same probability space (i.e. using the same sources of randomness). For analyzing HOGWILD!-Gibbs sampling, we share randomness by having both chains sample the same variable at each iteration and sample it such that the resulting values are maximally correlated—additionally both chains are subject to the same HOGWILD! delays $\tilde{\tau}_{i,t}$.

At some random time, called the *coupling time* T_{c}, the chains will become equal—regardless of their initial conditions. Using this, we can bound the mixing time with

$$t_{\mathrm{mix}}(\epsilon) \leq \min\{t \mid \mathbf{P}(T_{\mathrm{c}} > t) \leq \epsilon\}.$$

In order to bound the probability that the chains are not equal at a particular time t, we focus on the quantity

$$\phi_t = \max_i P\left(X_{i,t} \neq \bar{X}_{i,t}\right). \qquad (3)$$

Under the conditions of Theorem 3, we are able to bound this using the total influence parameter. From here, we notice that by the union bound, $\mathbf{P}(T_{\mathrm{c}} > t) \leq n\phi_t$. Combining this with Equation 3 and reducing the subsequent expression lets us bound the mixing time, producing the result of Theorem 3.

6. Experiments

Now that we have derived a theoretical characterization of the behavior of HOGWILD!-Gibbs sampling, we examine whether this characterization holds up under experimental evaluation. First, we examine the mixing time claims we made in Section 5. Specifically, we want to check whether increasing the expected delay parameter τ^* actually increases the mixing time as predicted by Equation 2.

To do this, we simulated HOGWILD!-Gibbs sampling running on a random synthetic Ising model graph of order $n = 1000$, degree $\Delta = 3$, inverse temperature $\beta = 0.2$, and prior weights $E_x = 0$. This model has total influence $\alpha \leq 0.6$, and Theorem 3 guarantees that it will mix rapidly. Unfortunately, the mixing time of a chain is difficult to calculate experimentally. While techniques such as coupling from the past (Propp & Wilson, 1996) exist for estimating the mixing time, using these techniques in order to expose

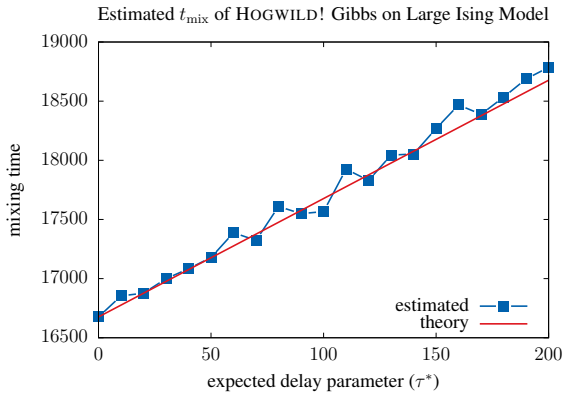

Figure 4. Comparison of estimated mixing time and theory-predicted (by Equation 2) mixing time as τ increases for a synthetic Ising model graph ($n = 1000$, $\Delta = 3$).

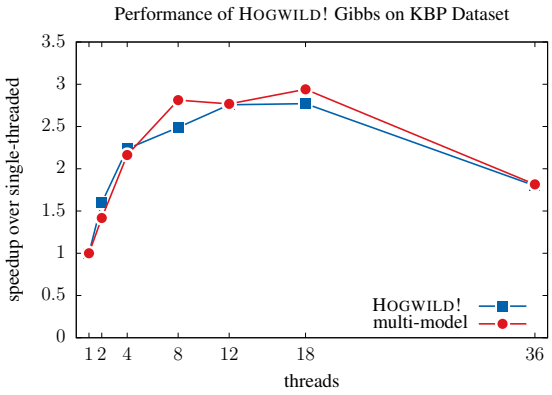

Figure 5. Speedup of HOGWILD! and multi-model Gibbs sampling on large KBP dataset (11 GB).

the (relatively small) dependence of the mixing time on τ proved to be computationally intractable.

Instead, we use a technique called coupling to the future. We initialize two chains, X and Y, by setting all the variables in X_0 to 1 and all the variables in Y_0 to -1. We proceed by simulating a coupling between the two chains, and return the coupling time T_{c}. Our estimate of the mixing time will then be $\hat{t}(\epsilon)$, where $\mathbf{P}(T_{\text{c}} \geq \hat{t}(\epsilon)) = \epsilon$.

Statement 2. *This experimental estimate is an upper bound for the mixing time. That is,* $\hat{t}(\epsilon) \geq t_{\text{mix}}(\epsilon)$.

To estimate $\hat{t}(\epsilon)$, we ran 10000 instances of the coupling experiment, and returned the sample estimate of $\hat{t}(1/4)$. To compare across a range of τ^*, we selected the $\tilde{\tau}_{i,t}$ to be independent and identically distributed according to the maximum-entropy distribution supported on $\{0, 1, \ldots, 200\}$ consistent with a particular assignment of τ^*. The resulting estimates are plotted as the blue series in Figure 4. The red line represents the mixing time that would be predicted by naively applying Equation 2 using the estimate of the sequential mixing time as a starting point — we can see that it is a very good match for the experimental results. This experiment shows that, at least for one archetypal model, our theory accurately characterizes the behavior of HOGWILD! Gibbs sampling as the delay parameter τ^* is changed, and that using HOGWILD!-Gibbs doesn't cause the model to catastrophically fail to mix.

Of course, in order for HOGWILD!-Gibbs to be useful, it must also speed up the execution of Gibbs sampling on some practical models. It is already known that this is the case, as these types of algorithms been widely implemented in practice (Smola & Narayanamurthy, 2010; Smyth et al., 2009). To further test this, we ran HOGWILD!-Gibbs sampling on a real-world 11 GB Knowledge Base Population dataset (derived from the TAC-KBP challenge) using a machine with a single-socket, 18-core Xeon E7-8890 CPU and 1 TB RAM. As a comparison, we also ran a "multi-model" Gibbs sampler: this consists of multiple threads with a single execution of Gibbs sampling running independently in each thread. This sampler will produce the same number of samples as HOGWILD!-Gibbs, but will require more memory to store multiple copies of the model.

Figure 5 reports the speedup, in terms of wall-clock time, achieved by HOGWILD!-Gibbs on this dataset. On this machine, we get speedups of up to $2.8\times$, although the program becomes memory-bandwidth bound at around 8 threads, and we see no significant speedup beyond this. With any number of workers, the run time of HOGWILD!-Gibbs is close to that of multi-model Gibbs, which illustrates that the additional cache contention caused by the HOGWILD! updates has little effect on the algorithm's performance.

7. Conclusion

We analyzed HOGWILD!-Gibbs sampling, a heuristic for parallelized MCMC sampling, on discrete-valued graphical models. First, we constructed a statistical model for HOGWILD!-Gibbs by adapting a model already used for the analysis of asynchronous SGD. Next, we illustrated a major issue with HOGWILD!-Gibbs sampling: that it produces biased samples. To address this, we proved that if for some class of models with bounded total influence, only $O(n)$ sequential Gibbs samples are necessary to produce good marginal estimates, then HOGWILD!-Gibbs sampling produces equally good estimates after only $O(1)$ additional steps. Additionally, for models that satisfy Dobrushin's condition ($\alpha < 1$), we proved mixing time bounds for sequential and asynchronous Gibbs sampling that differ by only a factor of $1 + O(n^{-1})$. Finally, we showed that our theory matches experimental results, and that HOGWILD!-Gibbs produces speedups up to $2.8\times$ on a real dataset.

Acknowledgments

The authors acknowledge the support of: DARPA FA8750-12-2-0335; NSF IIS-1247701; NSF CCF-1111943; DOE 108845; NSF CCF-1337375; DARPA FA8750-13-2-0039; NSF IIS-1353606; ONR N000141210041 and N000141310129; NIH U54EB020405; Oracle; NVIDIA; Huawei; SAP Labs; Sloan Research Fellowship; Moore Foundation; American Family Insurance; Google; and Toshiba.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, NSF, ONR, NIH, or the U.S. Government."

References

De Sa, Christopher, Zhang, Ce, Olukotun, Kunle, and Ré, Christopher. Rapidly mixing gibbs sampling for a class of factor graphs using hierarchy width. In *NIPS*. NIPS Foundation, 2015a.

De Sa, Christopher, Zhang, Ce, Olukotun, Kunle, and Ré, Christopher. Taming the wild: A unified analysis of HOGWILD!-style algorithms. In *NIPS*. NIPS Foundation, 2015b.

Dobrushin, RL. Central limit theorem for nonstationary markov chains. i. *Theory of Probability & Its Applications*, 1(4):329–383, 1956.

Dyer, Martin, Goldberg, Leslie Ann, and Jerrum, Mark. Dobrushin conditions and systematic scan. In *in Proc. 10th International Workshop on Randomization and Computation, Lecture Notes in Computer Science 4110*, pp. 327–338. Springer, 2006.

Gonzalez, Joseph, Low, Yucheng, Gretton, Arthur, and Guestrin, Carlos. Parallel gibbs sampling: From colored fields to thin junction trees. In *AISTATS*, pp. 324–332, 2011.

Gotovos, Alkis, Hassani, Hamed, and Krause, Andreas. Sampling from probabilistic submodular models. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 1936–1944. Curran Associates, Inc., 2015.

Guruswami, Venkatesan. Rapidly mixing markov chains: A comparison of techniques. *Available: cs. washington. edu/homes/venkat/pubs/papers. html*, 2000.

Hayes, Thomas P. A simple condition implying rapid mixing of single-site dynamics on spin systems. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pp. 39–46. IEEE, 2006.

Ising, Ernst. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

Johnson, Matthew, Saunderson, James, and Willsky, Alan. Analyzing hogwild parallel gaussian gibbs sampling. In *NIPS*, pp. 2715–2723, 2013.

Koller, Daphne and Friedman, Nir. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Levin, David Asher, Peres, Yuval, and Wilmer, Elizabeth Lee. *Markov chains and mixing times*. American Mathematical Soc., 2009.

Liu, Ji and Wright, Stephen J. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIOPT*, 25(1):351–376, 2015.

Liu, Ji, Wright, Stephen J, Ré, Christopher, Bittorf, Victor, and Sridhar, Srikrishna. An asynchronous parallel stochastic coordinate descent algorithm. *JMLR*, 16:285–322, 2015.

Lunn, David, Spiegelhalter, David, Thomas, Andrew, and Best, Nicky. The BUGS project: evolution, critique and future directions. *Statistics in medicine*, (25):3049–3067, 2009.

Mania, Horia, Pan, Xinghao, Papailiopoulos, Dimitris, Recht, Benjamin, Ramchandran, Kannan, and Jordan, Michael I. Perturbed iterate analysis for asynchronous stochastic optimization. *arXiv preprint arXiv:1507.06970*, 2015.

McCallum, Andrew, Schultz, Karl, and Singh, Sameer. Factorie: Probabilistic programming via imperatively defined factor graphs. In *NIPS*, pp. 1249–1257, 2009.

Mitliagkas, Ioannis, Borokhovich, Michael, Dimakis, Alexandros G., and Caramanis, Constantine. Frogwild!: Fast pagerank approximations on graph engines. *PVLDB*, 2015.

Neubig, Graham. Simple, correct parallelization for blocked gibbs sampling. Technical report, Nara Institute of Science and Technology, 2014.

Newman, David, Smyth, Padhraic, Welling, Max, and Asuncion, Arthur U. Distributed inference for latent dirichlet allocation. In *NIPS*, pp. 1081–1088, 2007.

Nguyen, Duc Thien, Yeoh, William, and Lau, Hoong Chuin. Distributed gibbs: A memory-bounded sampling-based dcop algorithm. In *Proceedings of the 2013 international conference on Autonomous agents*

and multi-agent systems, pp. 167–174. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

Niu, Feng, Recht, Benjamin, Re, Christopher, and Wright, Stephen. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, pp. 693–701, 2011.

Noel, Cyprien and Osindero, Simon. Dogwild!–Distributed Hogwild for CPU & GPU. 2014.

Propp, James Gary and Wilson, David Bruce. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9(1-2):223–252, 1996.

Shin, Jaeho, Wu, Sen, Wang, Feiran, De Sa, Christopher, Zhang, Ce, Wang, Feiran, and Ré, Christopher. Incremental knowledge base construction using deepdive. *PVLDB*, 2015.

Smola, Alexander and Narayanamurthy, Shravan. An architecture for parallel topic models. *PVLDB*, 2010.

Smyth, Padhraic, Welling, Max, and Asuncion, Arthur U. Asynchronous distributed learning of topic models. In *Advances in Neural Information Processing Systems*, pp. 81–88, 2009.

Sutter, Herb. The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software. *Dr. Dobb's Journal*, 30(3), 2005.

Terenin, Alexander, Simpson, Daniel, and Draper, David. Asynchronous distributed gibbs sampling. *arXiv preprint arXiv:1509.08999*, 2015.

Theis, Lucas, Sohl-dickstein, Jascha, and Bethge, Matthias. Training sparse natural image models with a fast gibbs sampler of an extended state space. In *NIPS*, pp. 1124–1132. 2012.

Yu, Hsiang-Fu, Hsieh, Cho-Jui, Si, Si, and Dhillon, Inderjit S. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *ICDM*, pp. 765–774, 2012.

Zhang, Ce and Ré, Christopher. DimmWitted: A study of main-memory statistical analytics. *PVLDB*, 2014.