Ensuring Rapid Mixing and Low Bias for Asynchronous Gibbs Sampling

Christopher De Sa, Kunle Olukotun, and Christopher Ré
Stanford University, Stanford, California
cdesa@stanford.edu, kunle@stanford.edu, chrismre@stanford.edu

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 and 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.

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

Gibbs sampling is often applied to sparse graphical models, where 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 [Newman *et al.*, 2007; Lunn *et al.*, 2009; McCallum *et al.*, 2009; Smola and Narayanamurthy, 2010; Theis *et al.*, 2012; Zhang and Ré, 2014].

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 and 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 previously proposed [Smola and Narayanamurthy, 2010], but not analyzed theoretically, and has been shown to give empirically better results on many models [Zhang and 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 useful for a particular application. Even the problems posed by HOGWILD!-Gibbs are poorly understood, and their solutions more so.

In the following sections, we will identify 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 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 $1 + O(n^{-1})$.

1.1 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 and Narayanamurthy, 2010; Zhang and Ré, 2014] because it requires additional memory.

The HOGWILD!-Gibbs sampling algorithm was inspired by a line of work on parallelizing stochastic gradient descent (SGD) by running it asynchronously [Niu *et al.*, 2011; Liu *et al.*, 2015; De Sa *et al.*, 2015; Mania *et al.*, 2015; Liu and Wright, 2015]. Several of these results suggest modeling the race conditions inherent in HOGWILD! SGD as noise in a stochastic process; in this paper, we will apply a similar stochastic process model to Gibbs sampling.

2 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.*, 2015]. We will focus on the case where our target distribution π ranges over a discrete space.

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 indexed by t. Let $x_{i,t}$ denote the value of variable i at time t. For HOGWILD!-Gibbs sampling, 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 model this, we define $\tilde{v}_{i,t} = x_{i,t-\tilde{\tau}_{i,t}}$ to be the potentially stale model state used to compute the update at time t, where $\tilde{\tau}_{i,t} \geq 0$ is a *delay parameter* that represents how old the currently-cached value for variable i could be. The distribution of these delays $\tilde{\tau}_{i,t}$ depends on the number of threads and the specifics of the hardware [Niu *et al.*, 2011]. To make our results general, we will not assume any particular distribution on the delays. Instead, we require only a bound on the expected delay, $\mathbf{E}\left[\tilde{\tau}_{i,t}|\mathcal{F}_t\right] \leq \tau$ for a hardware-dependent constant τ, and a very weak bound on its magnitude $\tilde{\tau}_{i,t} \leq n$, where n is the number of variables in the model.

3 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. Unfortunately, this is not the case for HOGWILD! Gibbs sampling, because the race conditions from the asynchrony add bias to the samples. To understand why, we can look at a simple example. Consider running HOGWILD! Gibbs on a model with two variables X_1 and X_2 each taking on values in $\{0, 1\}$, and having distribution

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

Suppose 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. To measure the amount of this bias, it is standard to use the *total variation distance*.

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

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

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

Here, we observe a total variation distance of 9.8% between the measured and the true distribution. Unlike in the sequential case, this bias doesn't disappear as the number of samples goes to infinity. This example shows that asynchronous Gibbs sampling will not necessarily samples that approach the target distribution. Instead, the samples may

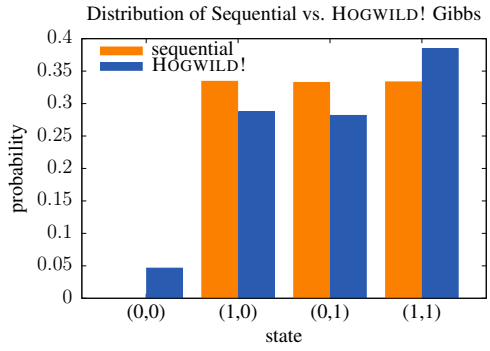

Figure 1: Bias introduced by HOGWILD! (10^6 samples).

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. 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 of these are events that we do not care about. To address this, we introduce a new metric.

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%.

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,

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

When analyzing Gibbs sampling, we can bound the sparse estimation time (and bias) using a parameter called the *total influence*. While we arrived at this condition independently, it has been studied before in the context of *Dobrushin's condition*, which ensures rapid mixing of Gibbs sampling.

Definition 4 (Total Influence)**.** Let π be a 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 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 prove a simple bound on the asymptotic sparse estimation time for asynchronous Gibbs sampling.

Theorem 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 fixed ϵ, across all distributions,*

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

where n is the number of variables in the model.[1] *Then, for any fixed 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" (in the sense of producing output with arbitrarily small bias ϵ) asymptotically on all classes of 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.

4 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] 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,

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

[1] In many practical systems [Neubig, 2014; Shin *et al.*, 2015], Gibbs sampling is naively run a fixed number of passes through the dataset, which works for exactly those models for which accurate marginal estimates can be achieved after $O(n)$ samples. This suggests that models with sparse estimation time $O(n)$ are practical and commonly occurring.

Unfortunately, as was the case with bias, the mixing time can also be greatly affected by running asynchronously. We can prove that there are classes of models for which asynchronous execution disastrously increases the mixing time.

Lemma 1. *There exist classes of distributions for which the mixing time of sequential Gibbs sampling is* $O(n \log n)$ *but the mixing time of* HOGWILD!*-Gibbs sampling, even with delay* $\tau = O(1)$*, can be* $\exp(\Omega(n))$.

This Lemma 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 2. *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*

$$t_{\mathrm{mix-hog}}(\epsilon) \leq \frac{n + 2\alpha\tau}{1-\alpha} \log\left(\frac{n}{\epsilon}\right).$$

Dobrushin's condition holds for a wide variety of problems, including the Ising model at high temperatures. We can compare these two mixing time results as

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

the bounds on the mixing times differ by a negligible factor of $1 + O(n^{-1})$. This result shows that, for problems that satisfy Dobrushin'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 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. Specifically, we want to check whether increasing the expected delay parameter τ actually increases the mixing time as predicted by Equation 1.

To do this, we simulated HOGWILD!-Gibbs sampling running on a random synthetic Ising model graph of order $n = 1000$, degree $\Delta = 3$, and inverse temperature $\beta = 0.2$. This model has total influence $\alpha \leq 0.6$, and Theorem 2 guarantees that it will mix rapidly. To estimate the mixing time, which is difficult to calculate experimentally, we use a technique called *coupling to the future*, which produces asymptotically tight upper-bound estimates of the mixing time. The blue series in Figure 2 displays the estimate of $t_{\mathrm{mix}}(1/4)$ across a range of τ[2]. The red line in Figure 2 shows the mixing time that would be predicted by naively applying Equation 1 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.

[2]We sampled the delays $\tilde{\tau}_{i,t}$ to be i.i.d. according to the maximum-entropy distribution supported on $\{0, 1, \ldots, 200\}$ consistent with a particular assignment of τ.

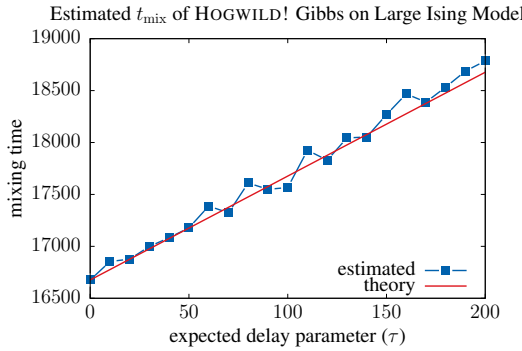

Figure 2: Comparison of estimated mixing time and theory-predicted (by Equation 1) mixing time as τ increases for a synthetic Ising model graph (10000 trials).

6 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 $1 + O(n^{-1})$.

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

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

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

[Hayes, 2006] Thomas P Hayes. 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*, pages 39–46. IEEE, 2006.

[Johnson *et al.*, 2013] Matthew Johnson, James Saunderson, and Alan Willsky. Analyzing hogwild parallel gaussian Gibbs sampling. In *NIPS*, pages 2715–2723, 2013.

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

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

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

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

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

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

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

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

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

[Nguyen *et al.*, 2013] Duc Thien Nguyen, William Yeoh, and Hoong Chuin Lau. Distributed Gibbs: A memory-bounded sampling-based DCOP algorithm. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 167–174. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

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

[Shin *et al.*, 2015] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, Feiran Wang, and Christopher Ré. Incremental knowledge base construction using DeepDive. *PVLDB*, 2015.

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

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

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

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