Chapter 7

Analyzing Hogwild Parallel Gaussian Gibbs Sampling

7.1 Introduction

Scaling probabilistic inference algorithms to large datasets and parallel computing architectures is a challenge of great importance and considerable current research interest, and great strides have been made in designing parallelizeable algorithms. Along with the powerful and sometimes complex new algorithms, a very simple strategy has proven to be surprisingly useful in some situations: running local Gibbs sampling updates on multiple processors in parallel while only periodically communicating updated statistics (see Section 7.4 for details). We refer to this strategy as "Hogwild Gibbs sampling" in reference to recent work [84] in which sequential computations for computing gradient steps were applied in parallel (without global coordination) to great beneficial effect.

This Hogwild Gibbs sampling strategy is not new; indeed, Gonzalez et al. [42] attributes a version of it to the original Gibbs sampling paper (see Section 7.2 for a discussion), though it has mainly been used as a heuristic method or initialization procedure without theoretical analysis or guarantees. However, extensive empirical work on Approximate Distributed Latent Dirichlet Allocation (AD-LDA) [83, 82, 73, 7, 55], which applies the strategy to generate samples from a collapsed LDA model [12], has demonstrated its effectiveness in sampling LDA models with the same or better predictive performance as those generated by standard serial Gibbs [83, Figure 3]. The results are empirical and so it is difficult to understand how model properties and algorithm parameters might affect performance, or whether similar success can be expected for any other models. There have been recent advances in understanding some of the particular structure of AD-LDA [55], but a thorough theoretical explanation for the effectiveness and limitations of Hogwild Gibbs sampling is far from complete.

Sampling-based inference algorithms for complex Bayesian models have notoriously resisted theoretical analysis, so to begin an analysis of Hogwild Gibbs sampling we consider a restricted class of models that is especially tractable for analysis: Gaussians.

Gaussian distributions and algorithms are tractable because of their deep connection with linear algebra. Further, Gaussian sampling is of significant interest in its own right, and there is active research in developing effective Gaussian samplers [72, 89, 90, 29]. Gaussian Hogwild Gibbs sampling can be used in conjunction with those methods to allow greater parallelization and scalability, provided some understanding of its applicability and tradeoffs.

The main contribution of this chapter is a linear algebraic framework for analyzing the stability and errors in Gaussian Hogwild Gibbs sampling. Our framework yields several results, including a simple proof for a sufficient condition for all Gaussian Hogwild Gibbs sampling processes to be stable and yield the correct asymptotic mean no matter the allocation of variables to processors. Our framework also provides an analysis of errors introduced in the process covariance, which in one case of interest leads to an inexpensive correction for those errors.

In Section 7.2 we discuss some related work in greater detail. In Section 7.3 we overview known connections between Gaussian sampling and linear system solvers, connections on which we build to provide an analysis for Hogwild Gibbs sampling. In Section 7.4 we precisely define the parallel updating scheme. Finally, in Section 7.5 we present our analytical framework and main results on Gaussian models.

■ 7.2 Related work

There has been significant work on constructing parallel Gibbs sampling algorithms, and the contributions are too numerous to list here. One recent body of work [42] provides exact parallel Gibbs samplers which exploit particular graphical model structure for parallelism. The algorithms are supported by the standard Gibbs sampling analysis, and the authors point out that while heuristic parallel samplers such as the AD-LDA sampler offer easier implementation and often greater parallelism, they are currently not supported by much theoretical analysis. Gonzalez et al. [42] attribute one version (see Section 7.4) of Hogwild Gibbs to the original Gibbs sampling paper [39] and refer to it as Synchronous Gibbs, though the Gibbs sampling paper only directly discusses an asynchronous implementation of their exact Gibbs sampling scheme rather than a parallelized approximation [39, Section XI]. Gonzalez et al. [42] also gives a result on Synchronous Gibbs in the special case of two processors.

The parallel sampling work that is most relevant to the proposed Hogwild Gibbs sampling analysis is the thorough empirical demonstration of AD-LDA [83, 82, 73, 7, 55] and its extensions. The AD-LDA sampling algorithm is an instance of the strategy we have named Hogwild Gibbs, and Bekkerman et al. [7, Chapter 11] suggests applying the strategy to other latent variable models.

The work of Ihler and Newman [55] provides some understanding of the effective-

ness of a variant of AD-LDA by bounding in terms of run-time quantities the one-step error probability induced by proceeding with sampling steps in parallel, thereby allowing an AD-LDA user to inspect the computed error bound after inference [55, Section 4.2]. In experiments, the authors empirically demonstrate very small upper bounds on these one-step error probabilities, e.g. a value of their parameter $\varepsilon = 10^{-4}$ meaning that at least 99.99% of samples are expected to be drawn just as if they were sampled sequentially. However, this per-sample error does not necessarily provide a direct understanding of the effectiveness of the overall algorithm because errors might accumulate over sampling steps; indeed, understanding this potential error accumulation is of critical importance in iterative systems. Furthermore, the bound is in terms of empirical run-time quantities, and thus it does not provide guidance regarding on which other models the Hogwild strategy may be effective. Ihler and Newman [55, Section 4.3] also provides approximate scaling analysis by estimating the order of the one-step bound in terms of a Gaussian approximation and some distributional assumptions.

Finally, Niu et al. [84] provides both a motivation for Hogwild Gibbs sampling as well as the Hogwild name. The authors present "a lock-free approach to parallelizing stochastic gradient descent" (SGD) by providing analysis that shows, for certain common problem structures, that the locking and synchronization needed for a stochastic gradient descent algorithm to converge on a multicore architecture are unnecessary, and in fact the robustness of the SGD algorithm compensates for the uncertainty introduced by allowing processors to perform updates without locking their shared memory.

■ 7.3 Gaussian sampling background

In this section we fix notation for Gaussian distributions and describe known connections between Gaussian sampling and a class of stationary iterative linear system solvers which are useful in analyzing the behavior of Hogwild Gibbs sampling.

The density of a Gaussian distribution on n variables with mean vector μ and positive definite[1] covariance matrix $\Sigma \succ 0$ has the form

$$p(x) \propto \exp\left\{-\tfrac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\right\} \propto \exp\left\{-\tfrac{1}{2}x^{\mathsf{T}}Jx + h^{\mathsf{T}}x\right\} \tag{7.3.1}$$

where we have written the information parameters $J \triangleq \Sigma^{-1}$ and $h \triangleq J\mu$. The matrix J is often called the *precision matrix* or *information matrix*, and it has a natural interpretation in the context of Gaussian graphical models: its entries are the coefficients on pairwise log potentials and its sparsity pattern is exactly the sparsity pattern of a graphical model. Similarly h, also called the *potential vector*, encodes node potentials and evidence.

[1] We assume models are non-degenerate, i.e. that covariances are of full rank.

In many problems [113] one has access to the pair (J, h) and must compute or estimate the moment parameters μ and Σ (or just the diagonal) or generate samples from $\mathcal{N}(\mu, \Sigma)$. Sampling provides both a means for estimating the moment parameters and a subroutine for other algorithms. Computing μ from (J, h) is equivalent to solving the linear system $J\mu = h$ for μ.

One way to generate samples is via Gibbs sampling, in which one iterates sampling each x_i conditioned on all other variables to construct a Markov chain for which the invariant distribution is the target $\mathcal{N}(\mu, \Sigma)$. The conditional distributions for Gibbs sampling steps are

$$p(x_i | x_{\neg i} = \bar{x}_{\neg i}) \propto \exp\left\{-\frac{1}{2}\begin{pmatrix} x_i & \bar{x}_{\neg i}^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} J_{ii} & J_{i\neg i} \\ J_{\neg ii} & J_{\neg i\neg i} \end{pmatrix} \begin{pmatrix} x_i \\ \bar{x}_{\neg i} \end{pmatrix} + \begin{pmatrix} h_i & h_{\neg i}^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} x_i \\ \bar{x}_{\neg i} \end{pmatrix}\right\} \tag{7.3.2}$$

$$\propto \exp\left\{-\frac{1}{2} J_{ii} x_i^2 + (h_i - J_{i\neg i}\bar{x}_{\neg i}) x_i\right\} \tag{7.3.3}$$

where the indexing $x_{\neg i} \triangleq (x_j : j \neq i) \in \mathbb{R}^{n-1}$ denotes all the variables other than x_i and $J_{i\neg i} \triangleq (J_{ij} : j \neq i)$ denotes the ith row of J with its ith entry removed. That is, we update each x_i to be a scalar Gaussian sample with mean $\frac{1}{J_{ii}}(h_i - J_{i\neg i}\bar{x}_i)$ and variance $\frac{1}{J_{ii}}$ or, equivalently,

$$x_i \leftarrow \frac{1}{J_{ii}}(h_i - J_{i\neg i}\bar{x}_{\neg i}) + v_i \quad \text{where} \quad v_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \frac{1}{J_{ii}}). \tag{7.3.4}$$

Since each variable update is a linear function of other variables with added Gaussian noise, we can collect one scan for $i = 1, 2, \ldots, n$ into a matrix equation relating the sampler state vector at t and $t + 1$:

$$x^{(t+1)} = -D^{-1}Lx^{(t+1)} - D^{-1}L^{\mathsf{T}}x^{(t)} + D^{-1}h + D^{-\frac{1}{2}}\tilde{v}^{(t)} \tag{7.3.5}$$

$$\tilde{v}^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, I). \tag{7.3.6}$$

where we have split $J = L + D + L^{\mathsf{T}}$ into its strictly lower-triangular, diagonal, and strictly upper-triangular parts, respectively. Note that $x^{(t+1)}$ appears on both sides of the equation, and that the sparsity patterns of L and L^{T} ensure that the updated value $x_i^{(t+1)}$ depends only on $x_a^{(t)}$ and $x_b^{(t+1)}$ for all $a > i$ and $b < i$. We can rearrange the equation into an update expression:

$$(I + D^{-1}L)x^{(t+1)} = -D^{-1}L^{\mathsf{T}}x^{(t)} + D^{-1}h + D^{-\frac{1}{2}}v^{(t)} \tag{7.3.7}$$

$$x^{(t+1)} = -(D+L)^{-1}L^{\mathsf{T}}x^{(t)} + (D+L)^{-1}h + (D+L)^{-1}D^{\frac{1}{2}}v^{(t)} \tag{7.3.8}$$

$$= -(D+L)^{-1}L^{\mathsf{T}}x^{(t)} + (D+L)^{-1}h + (D+L)^{-1}\tilde{v}^{(t)} \tag{7.3.9}$$

$$\tilde{v}^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, D). \tag{7.3.10}$$

The expectation of this update is exactly the Gauss-Seidel iterative linear system solver update [9, Section 7.3] applied to $J\mu = h$, i.e. $x^{(t+1)} = -(D+L)^{-1}L^{\mathsf{T}}x^{(t)} + (D+L)^{-1}h$. Therefore a Gaussian Gibbs sampling process can be interpreted as Gauss-Seidel iterates on the system $J\mu = h$ with appropriately-shaped noise injected at each iteration.

Gauss-Seidel is one instance of a stationary iterative linear solver based on a *matrix splitting*. In general, one can construct a stationary iterative linear solver for any splitting $J = M - N$ where M is invertible, and similarly one can construct iterative Gaussian samplers via

$$x^{(t+1)} = (M^{-1}N)x^{(t)} + M^{-1}h + M^{-1}v^{(t)} \tag{7.3.11}$$

$$v^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, M^{\mathsf{T}} + N) \tag{7.3.12}$$

with the constraint that $M^{\mathsf{T}} + N \succeq 0$ (i.e. that the splitting is P-regular [77]). For a stationary iterative process like (7.3.11) to be *stable* or *convergent* for any initialization we require the eigenvalues of its update map to lie in the interior of the complex unit disk, i.e. $\rho(M^{-1}N) \triangleq \max_i |\lambda_i(M^{-1}N)| < 1$ [9, Lemma 7.3.6]. The Gauss-Seidel solver (and Gibbs sampling) correspond to choosing M to be the lower-triangular part of J and N to be the negative of the strict upper-triangle of J. $J \succ 0$ is a sufficient condition for Gauss-Seidel to be convergent [9, Theorem 7.5.41] [101], and the connection to Gibbs sampling provides an alternative proof.

For solving linear systems with splitting-based algorithms, the complexity of solving linear systems in M directly affects the computational cost per iteration. For the Gauss-Seidel splitting (and hence Gibbs sampling), M is chosen to be lower-triangular so that the corresponding linear system can be solved efficiently via back-substitution. In the sampling context, the per-iteration computational complexity is also determined by the covariance of the injected noise process $v^{(t)}$, because at each iteration one must sample from a Gaussian distribution with covariance $M^{\mathsf{T}} + N$.

We highlight one other standard stationary iterative linear solver that is relevant to analyzing Gaussian Hogwild Gibbs sampling: Jacobi iterations, in which one splits

$J = D - A$ where D is the diagonal part of J and A is the negative of the off-diagonal part. Due to the choice of a diagonal M, each coordinate update depends only on the previous sweep's output, and thus the Jacobi update sweep can be performed in parallel. A sufficient condition for the convergence of Jacobi iterates is for J to be a generalized diagonally dominant matrix (i.e. an H-matrix) [9, Definition 5.13]. A simple proof [2] due to Ruozzi and Tatikonda [96], is to consider Gauss-Seidel iterations on a *lifted* $2n \times 2n$ system:

$$\begin{pmatrix} D & -A \\ -A & D \end{pmatrix} \xrightarrow{\text{G-S update}} \begin{pmatrix} D^{-1} & \\ D^{-1}AD^{-1} & D^{-1} \end{pmatrix} \begin{pmatrix} & A \\ & \end{pmatrix} = \begin{pmatrix} & D^{-1}A \\ & (D^{-1}A)^2 \end{pmatrix} \tag{7.3.13}$$

where zero entries are left blank where dimensions can be inferred. Therefore one iteration of Gauss-Seidel on the lifted system corresponds to two iterations of the Jacobi update $D^{-1}A$ to the latter n entries in the lifted system, so Jacobi iterations converge if Gauss-Seidel on the lifted system converges. Furthermore, a sufficient condition for Gauss-Seidel to converge on the lifted system is for the lifted matrix to be positive definite, and by taking Schur complements we require $D - AD^{-1}A \succ 0$ or $I - (D^{-\frac{1}{2}}AD^{-\frac{1}{2}})(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}) \succ 0$, which is equivalent to requiring strict generalized diagonal dominance of J [9, Theorem 5.14].

■ 7.4 Hogwild Gibbs model

In this section, we define the Hogwild Gibbs computational model and fix some notation for the iterative process that we use for the remainder of the chapter.

As with standard Gibbs sampling, we assume we are given a collection of n random variables $\{x_i : i \in [n]\}$ where $[n] \triangleq \{1, 2, \ldots, n\}$ and that we can sample from the conditional distributions $x_i|x_{\neg i}$. Gibbs sampling is an iterative Markov process on the *state vector* $x^{(t)}$ for times $t = 1, 2, \ldots$ so that the stationary distribution is the joint distribution of $\{x_i : i \in [n]\}$.

For Hogwild Gibbs, we assume we are given a partition $\{\mathcal{I}_1, \mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $[n]$ that represents an allocation of the state vector to K processors, so that the kth processor updates the state values indexed by $\mathcal{I}_k$. We assume each partition element $\mathcal{I}_k$ is contiguous and ordered and we write $x_{\mathcal{I}_k} \triangleq (x_i : i \in \mathcal{I}_k)$ to denote the corresponding sub-vector of any vector x. We keep this partition fixed over time for the majority of this chapter, though we describe a generalization in Theorem 7.6.7.

The Hogwild Gibbs algorithm is shown in Algorithm 7.1. We define two iterations: *outer iterations*, which count the number of global synchronizations among the proces-

[2] When J is symmetric one can arrive at the same condition by applying a similarity transform as in Proposition 7.7.4. We use the lifting argument here because we extend the idea in our other proofs.

Algorithm 7.1 Hogwild Gibbs

Input: Joint distribution over $x = (x_1, \ldots, x_n)$, partition $\{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $\{1, 2, \ldots, n\}$

```
Initialize x̄^(1)
for t = 1, 2, ... do
    for k = 1, 2, ..., K in parallel do
        x̄_{I_k}^(t+1) ← LocalGibbs(x̄^(t), I_k, q(t, k))
function LocalGibbs(x̄, I, q)
    for j = 1, 2, ..., q do
        for i ∈ I in order do
            x̄_i ← sample x_i | x_{¬i} = x̄_{¬i}
    return x̄
```

sors, and *inner iterations*, which count processor-local Gibbs scans. That is, during outer iteration t (for each $t = 1, 2, \ldots$), processor k runs a number $q(t, k)$ of inner iterations, each of which consists of a systematic scan Gibbs update [94, Algorithm A.40] of its variables indexed by $\mathcal{I}_k$. During the inner iterations on each processor, the processors do not communicate; in particular, all inner iterations on processor k compute Gibbs updates using out-of-date values of x_i for $i \notin \mathcal{I}_k$. Processors synchronize values once per outer iteration, and we write $x^{(t)}$ for the globally shared value before the inner iterations of outer iteration t. For the majority of this chapter, we fix the number of inner iterations performed to be constant for all processors and for all outer iterations, so that $q(t, k) = q$, though we describe a generalization in Theorem 7.6.7.

There are several special cases of this general scheme that may be of interest. The Synchronous Gibbs scheme of Gonzalez et al. [42] corresponds to associating one variable to each processor, so that $|\mathcal{I}_k| = 1$ for each $k = 1, 2, \ldots, K$ (in which case we may take $q = 1$ since no local iterations are needed with a single variable). More generally, it is particularly interesting to consider the case where the partition is arbitrary and q is very large, in which case the local Gibbs iterations can mix and exact block samples are drawn on each processor using old statistics from other processors for each outer iteration. Finally, note that setting $K = 1$ and $q = 1$ reduces to standard Gibbs sampling on a single processor.

■ 7.5 Gaussian analysis setup

Given that Gibbs sampling iterations and Jacobi solver iterations can each be written as iterations of a stochastic linear dynamical system (LDS), it is not surprising that Gaussian Hogwild Gibbs sampling can also be expressed as an LDS by appropriately composing these ideas. In this section we describe the LDS corresponding to Gaussian Hogwild Gibbs sampling and provide convergence and error analysis, along with a

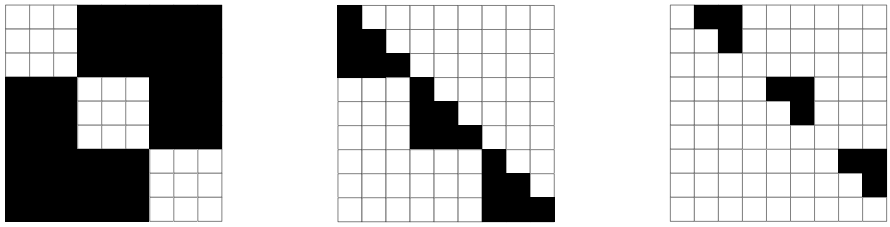


Figure 7.1: Support pattern (in black) of the Hogwild splitting $J = B - C - A$ with $n = 9$ and the processor partition $\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$.

connection to a class of linear solvers.

Given a joint Gaussian distribution of dimension n represented by a pair (J, h) as in (7.3.1), consider a block-Jacobi splitting of J into its block diagonal and off-block-diagonal components, $J = D_{\text{bd}} - A$, according to the partition. A includes the entries of J corresponding to cross-processor terms, and this block-Jacobi splitting will model the outer iterations in Algorithm 7.1. We further perform a Gauss-Seidel splitting on D_{bd} into (block-diagonal) lower-triangular and strictly upper-triangular parts, $D_{\text{bd}} = B - C$; these processor-local Gauss-Seidel splittings model the inner itertions in Algorithm 7.1. We refer to this splitting $J = B - C - A$ as the Hogwild splitting; see Figure 7.1 for an example.

For each outer iteration of the Hogwild Gibbs sampler we perform q processor-local Gibbs steps, effectively applying the block-diagonal update $B^{-1}C$ repeatedly using $Ax^{(t)} + h$ as a potential vector that includes out-of-date statistics from the other processors. The resulting update operator for one outer iteration of the Hogwild Gibbs sampling process is

$$x^{(t+1)} = (B^{-1}C)^q x^{(t)} + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1} \left(Ax^{(t)} + h + v^{(t,j)} \right) \tag{7.5.1}$$

$$v^{(t,j)} \overset{\text{iid}}{\sim} \mathcal{N}(0, D) \tag{7.5.2}$$

where D is the diagonal of J. Note that we shape the noise diagonally because in Hogwild Gibbs sampling we simply apply standard Gibbs updates in the inner iterations.

■ 7.6 Convergence and correctness of means

Because the Gaussian Hogwild Gibbs sampling iterates form a Gaussian linear dynamical system, the process is stable (i.e. its iterates converge in distribution) if and only

if [9, Lemma 7.3.6] the deterministic part of the update map (7.5.2) has spectral radius less than unity, i.e.

$$T \triangleq (B^{-1}C)^q + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1}A \tag{7.6.1}$$

$$= (B^{-1}C)^q + (I - (B^{-1}C)^q)(I - (B^{-1}C))^{-1}B^{-1}A \tag{7.6.2}$$

$$= (B^{-1}C)^q + (I - (B^{-1}C)^q)(B - C)^{-1}A \tag{7.6.3}$$

$$= T_{\text{ind}}^q + (I - T_{\text{ind}}^q)T_{\text{bl}}, \tag{7.6.4}$$

where

$$T_{\text{ind}} \triangleq (B^{-1}C) \qquad T_{\text{bl}} \triangleq (B - C)^{-1}A, \tag{7.6.5}$$

satisfies $\rho(T) < 1$. The term T_{ind} is the block Gauss-Seidel update when $A = 0$ and the processors' random variables are independent, while the term T_{bl} is the block Jacobi update, which corresponds to solving the processor-local linear systems exactly at each outer iteration. The update (7.6.4) falls into the class of two-stage splitting methods [77, 35, 34], and the next proposition is equivalent to such two-stage solvers having the correct fixed point.

Proposition 7.6.1. *If a Gaussian Hogwild Gibbs process is stable, then its mean is* $\mu = J^{-1}h$.

Proof. If the process is stable the mean process has a unique fixed point, from (7.5.2) and (7.6.4) and using the definitions of T_{ind} and T_{block} we can write the fixed-point equation for the process mean μ_{Hog} as

$$(I - T)\mu_{\text{Hog}} = (I - T_{\text{ind}})(I - T_{\text{block}})\mu_{\text{Hog}} = (I - T_{\text{ind}})(B - C)^{-1}h, \tag{7.6.6}$$

hence $(I - (B - C)^{-1}A)\mu_{\text{Hog}} = (B - C)^{-1}h$ and $\mu_{\text{Hog}} = (B - C - A)^{-1}h = J^{-1}h$. □

The behavior of the spectral radius of the update map can be very complicated. In Figure 7.2, we compare $\rho(T)$ for $q = 1$ and $q = \infty$ for models generated from a simple random ensemble. Each point corresponds to a sampled model $J = QQ^{\mathsf{T}} + nrI$ with $Q_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ and $r \overset{\text{iid}}{\sim} \text{Uniform}[0.5, 1]$, and the value of each point's vertical coordinate is the spectral radius of the Hogwild update T when $q = \infty$ (i.e. $T = T_{\text{block}}$) while the horizontal coordinate is the spectral radius of T when $q = 1$. Hogwild Gibbs sampling on the model is convergent with $q = 1$ when the point is to the left of the vertical red line, and it is convergent as $q = \infty$ when the point is below the horizontal line. The figure shows that, while convergence in the two cases shows a positive correlation, Hogwild Gibbs can be convergent when $q = 1$ and not when $q = \infty$ and it can be

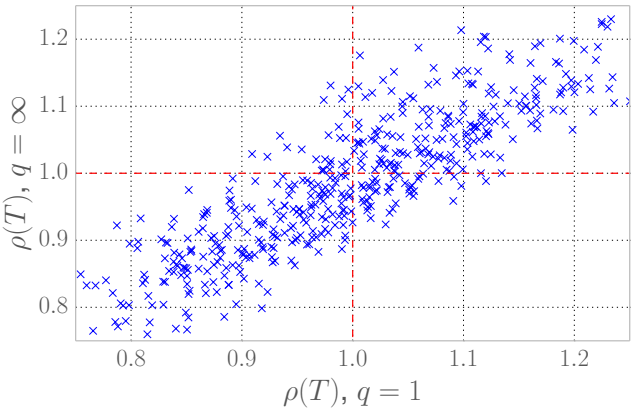


Figure 7.2: Comparing Hogwild stability on random models for extreme values of the inner iteration count q. Each point corresponds to a sampled model, where the horizontal coordinate is the spectral radius at $q = 1$ and the vertical coordinate is the spectral radius at $q = \infty$.

convergent when $q = \infty$ and not when $q = 1$. Therefore the behavior of the algorithm with varying q is difficult to understand in general.

Despite the complexity of the update map's stability, in the next subsection we give a simple argument that identifies its convergence with the convergence of Gauss-Seidel iterates on a larger, non-symmetric linear system. Given that relationship we then prove a condition on the entries of J that ensures the stability of the Gaussian Hogwild Gibbs sampling process.

■ 7.6.1 A lifting argument and sufficient condition

First observe that we can write multiple steps of Gauss-Seidel as a single step of Gauss-Seidel on a larger system: given $J = L - U$ where L is lower-triangular (including the diagonal, unlike the notation of Section 7.3) and U is strictly upper-triangular, consider applying Gauss-Seidel to a larger block $k \times k$ system:

$$\begin{pmatrix} L & & & -U \\ -U & L & & \\ & \ddots & \ddots & \\ & & -U & L \end{pmatrix} \xrightarrow{\text{G-S}} \begin{pmatrix} L^{-1} & & & \\ L^{-1}UL^{-1} & L^{-1} & & \\ \vdots & & \ddots & \\ (L^{-1}U)^{k-1}L^{-1} & \cdots & L^{-1}UL^{-1} & L^{-1} \end{pmatrix} \begin{pmatrix} & & & U \\ & & & \\ & & & \\ & & & \end{pmatrix} = \begin{pmatrix} & & & L^{-1}U \\ & & & \vdots \\ & & & (L^{-1}U)^k \end{pmatrix} \tag{7.6.7}$$

Therefore one step of Gauss-Seidel on the larger system corresponds to k applications of the Gauss-Seidel update $L^{-1}U$ from the original system to the last block element of the lifted state vector.

Now we provide a lifted linear system on which Gauss-Seidel iterations correspond to applying Gaussian Hogwild Gibbs iterations to a block component.

Proposition 7.6.2. *Two applications of the Hogwild update T of (7.6.4) are equivalent to the update to the last block element of the state vector in one Gauss-Seidel iteration on the $(2qn) \times (2qn)$ system*

$$\begin{pmatrix} E & -F \\ -F & E \end{pmatrix} \tilde{x} = \begin{pmatrix} h \\ \vdots \\ h \end{pmatrix} \text{ with } E = \begin{pmatrix} B & & \\ -C & B & \\ & \ddots & \ddots \\ & & -C \; B \end{pmatrix} \quad F = \begin{pmatrix} & & A+C \\ & & A \\ & & \vdots \\ & & A \end{pmatrix}. \tag{7.6.8}$$

That is, if $P = \begin{pmatrix} 0 & \cdots & 0 & I \end{pmatrix}$ is $n \times 2qn$ with an $n \times n$ identity as its last block entry, then

$$P \begin{pmatrix} E & \\ -F & E \end{pmatrix}^{-1} \begin{pmatrix} & F \end{pmatrix} P^{\mathsf{T}} = P \begin{pmatrix} & E^{-1}F \\ & (E^{-1}F)^2 \end{pmatrix} P^{\mathsf{T}} = T^2. \tag{7.6.9}$$

Proof. It suffices to consider $E^{-1}F$. Furthermore, since the claim concerns the last block entry, we need only consider the last block row of $E^{-1}F$. E is block lower-bidiagonal and hence E^{-1} has the same lower-triangular form as in (7.6.7),

$$E^{-1} = \begin{pmatrix} B^{-1} & & & \\ B^{-1}CB^{-1} & B^{-1} & & \\ \vdots & \ddots & \ddots & \\ (B^{-1}C)^{q-1}B^{-1} & \dots & B^{-1}CB^{-1} & B^{-1} \end{pmatrix}, \tag{7.6.10}$$

and the product of the last block row of E^{-1} with the last block column of F yields

$$\begin{aligned} &\begin{pmatrix} (B^{-1}C)^{q-1}B^{-1} & \dots & (B^{-1}C)B^{-1} & B^{-1} \end{pmatrix} \cdot \begin{pmatrix} A+C & A & \dots & A \end{pmatrix} && (7.6.11) \\ &= (B^{-1}C)^q + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1}A = T. && (7.6.12) \end{aligned}$$

□

Proposition 7.6.3. *Gaussian Hogwild Gibbs sampling is convergent if Gauss-Seidel converges on the system (7.6.8).*

To give a sufficient condition for the convergence of Gauss-Seidel on the lifted system and hence the Gaussian Hogwild Gibbs process, we first state a standard result for Gauss-Seidel and a simple corollary.

Lemma 7.6.4 (Theorem 6.2 [22])**.** *If J is strictly diagonally dominant then its Gauss-Seidel update matrix is a contraction in max norm; that is, if for every i we have $|J_{ii}| > \sum_{j \neq i} |J_{ij}|$ then letting $J = L - U$ where L is lower-triangular and U is strictly*

upper-triangular we have

$$||L^{-1}U||_\infty < 1 \quad \textit{where} \quad ||A||_\infty \triangleq \sup_{x\neq 0} \frac{||Ax||_\infty}{||x||_\infty} = \max_i \sum_{j=1}^n |A_{ij}| \tag{7.6.13}$$

and where $||x||_\infty \triangleq \max_i |x_i|$.

Note that the Gauss-Seidel update being a contraction in any induced matrix norm immediately implies it is convergent since the spectral radius is upper bounded by any induced norm; that is, $\rho(A) \leq ||A||$ for any induced matrix norm $||\cdot||$ because if v is the eigenvector corresponding to an eigenvalue of A that achieves its spectral radius then

$$||A|| = \sup_{x\neq 0} \frac{||Ax||}{||x||} \geq \frac{||Av||}{||v||} = \rho(A). \tag{7.6.14}$$

We can extend the lemma slightly by considering generalized diagonally dominant matrices and adapting the max norm accordingly.

Corollary 7.6.5. *If J is strictly generalized diagonally dominant then its Gauss-Seidel update matrix is a contraction in a weighted max norm; that is, if there exists an $r \in \mathbb{R}^n$ with $r > 0$ entrywise such that for every i we have $r_i|J_{ii}| > \sum_{j\neq i} r_j|J_{ij}|$, then letting $J = L - U$ where L is lower-triangular and U is strictly upper-triangular we have*

$$||L^{-1}U||_\infty^r < 1 \quad \textit{where} \quad ||A||_\infty^r \triangleq \sup_{x\neq 0} \frac{||Ax||_\infty^r}{||x||_\infty^r} = \max_i \frac{1}{r_i} \sum_{j=1}^n |A_{ij}|r_j \tag{7.6.15}$$

and where $||x||_\infty^r \triangleq \max_i \frac{1}{r_i}|x_i|$.

Proof. Let $R \triangleq \text{diag}(r)$ and note that JR is strictly diagonally dominant. Therefore by Lemma 7.6.4 we have that

$$1 > ||R^{-1}L^{-1}UR||_\infty = \max_{x\neq 0} \frac{||R^{-1}L^{-1}URx||_\infty}{||x||_\infty} \tag{7.6.16}$$

$$= \max_{x\neq 0} \frac{||L^{-1}URx||_\infty^r}{||x||_\infty} \tag{7.6.17}$$

$$= \max_{x\neq 0} \frac{||L^{-1}U||_\infty^r}{||x||_\infty^r} = ||L^{-1}U||_\infty^r \tag{7.6.18}$$

where we have used $||x||_\infty^r = ||R^{-1}x||_\infty$ and on the last line substituted $x \mapsto R^{-1}x$. □

With generalized diagonal dominance as a sufficient condition for Gauss-Seidel convergence, we can use the lifting construction of Proposition 7.6.2 to give a sufficient condition for the convergence of Gaussian Hogwild Gibbs.

Theorem 7.6.6. *If J is strictly generalized diagonally dominant, that is if there exists an $r \in \mathbb{R}^n$ with $r > 0$ entrywise such that*

$$r_i|J_{ii}| > \sum_{j \neq i} r_j|J_{ij}|, \tag{7.6.19}$$

then Gaussian Hogwild Gibbs sampling is convergent for any fixed variable partition and any fixed number of inner iterations. Further, we have $||T||_\infty^r < 1$.

Proof. Since each scalar row of the coefficient matrix in (7.6.8) contains only entries from one row of J and zeros, it is generalized diagonally dominant with a scaling vector that consists of $2q$ copies of r. Gauss-Seidel iterations on generalized diagonally dominant systems are convergent by Lemma 7.6.4 and so by Proposition 7.6.3 the corresponding Gaussian Hogwild Gibbs iterations are convergent.

To show the stronger result that the update is a contraction, first we define $\widetilde{T}$ to be the Gauss-Seidel update matrix for the system (7.6.8), i.e.

$$\widetilde{T} \triangleq \begin{pmatrix} E & \\ -F & E \end{pmatrix}^{-1} \begin{pmatrix} & F \\ & \end{pmatrix}, \tag{7.6.20}$$

and we define $\widetilde{r}$ to be $2q$ copies of r. For any $x \in \mathbb{R}^n$ we have

$$||x||_\infty^r = ||P^\mathsf{T} x||_\infty^{\widetilde{r}} > ||\widetilde{T} P^\mathsf{T} x||_\infty^{\widetilde{r}} \geq ||P\widetilde{T} P^\mathsf{T} x||_\infty^r = ||Tx||_\infty^r \tag{7.6.21}$$

where we have used the orthogonality of P and Corollary 7.6.5. □

Note that the lifting construction in (7.6.8) immediately generalizes to the case where the number of inner iterations varies from processor to processor. Furthermore, the proof of Theorem 7.6.6 shows that T is a contraction in $||\cdot||_\infty^r$ regardless of the partition or structure or inner iteration counts. Therefore we can immediately generalize the result to the non-stationary case, where the numbers of inner iterations and even the partition structure vary across outer iterations.

Theorem 7.6.7. *If J is strictly generalized diagonally dominant, then for any inner iteration schedule q with $1 \leq q(t,k) < q_{\max}$ (for $t = 1, 2, \ldots$, $k = 1, 2, \ldots, K$, and any $q_{\max} < \infty$) and any sequence of partitions $\mathcal{I}^{(t)} = \{\mathcal{I}_1^{(t)}, \ldots, \mathcal{I}_K^{(t)}\}$ Gaussian Hogwild Gibbs is convergent.*

Proof. We write $\mathcal{T}$ for the set of all possible update maps T, where $T^{(t)}$ is a function of both $q(t,k)$ and $\mathcal{I}^{(t)}$. The process is convergent if the joint spectral radius [95, 62] of $\mathcal{T}$ satisfies

$$\rho(\mathcal{T}) \triangleq \lim_{\ell\to\infty} \sup\{||T_1 \cdots T_\ell||^{\frac{1}{\ell}} : T_i \in \mathcal{T}\} < 1 \tag{7.6.22}$$

where $||\cdot||$ is any matrix norm. We use the matrix norm induced by the vector norm $||\cdot||_\infty^r$ defined in (7.6.15) and note that any induced norm is submultiplicative, so that for any matrices T_1 and T_2

$$||T_1T_2||_\infty^r \leq ||T_1||_\infty^r ||T_2||_\infty^r. \tag{7.6.23}$$

Then, using the submultiplicative property and the contraction property from Theorem 7.6.6, for any ℓ and any $T_1, T_2, \ldots, T_\ell \in \mathcal{T}$ we have

$$(||T_1 \cdots T_\ell||_\infty^r)^{\frac{1}{\ell}} \leq (||T_1||_\infty^r \cdots ||T_\ell||_\infty^r)^{\frac{1}{\ell}} \leq 1 - \epsilon \tag{7.6.24}$$

for some $\epsilon > 0$ using the fact that $\mathcal{T}$ is finite. Therefore $\rho(\mathcal{T}) < 1$ and the process is convergent. □

Generalized diagonally dominant matrices are also known as H-matrices [9, Definition 5.13]; see Berman and Plemmons [9, Theorem 5.14] for a long list of equivalent characterizations. For an H-matrix to be a valid precision matrix it must also be positive semidefinite (PSD). Such matrices can also be described as having factor-width two [13]; that is, a PSD H-matrix J can be factored as $J = GG^\mathsf{T}$ where G is a rectangular matrix in which each column has at most two nonzeros.

In terms of Gaussian graphical models, generalized diagonally dominant models include tree models and latent tree models (since H-matrices are closed under Schur complements), in which the density of the distribution can be written as a tree-structured set of pairwise potentials over the model variables and a set of latent variables. Latent tree models are useful in modeling data with hierarchical or multiscale relationships, and this connection to latent tree structure is evocative of many hierarchical Bayesian models. PSD H-matrices also include walk-summable matrices [75], for which the Gaussian Loopy Belief Propagation algorithm converges and yields correct mean estimates. More broadly, diagonally dominant systems are well-known for their tractability and applicability in many other settings [63], and Gaussian Hogwild Gibbs provides another example of their utility.

Because of the connection to linear system solvers known as two-stage multisplittings, these results can be identified with Theorem 2.3 of Frommer and Szyld [34], which shows that if the coefficient matrix is an H-matrix then the corresponding two-stage iterative solver is convergent. Indeed, by the connection between solvers and samplers

one can prove these convergence theorems as corollaries to Frommer and Szyld [34, Theorem 2.3] (or vice-versa), though our proof technique is much simpler. The other results on two-stage multisplittings [34, 77], including the results on asynchronous iterations, can also be applied immediately for results on the convergence of Gaussian Hogwild Gibbs sampling.

The sufficient conditions provided by Theorems 7.6.6 and 7.6.7 are coarse in that they provide convergence for any partition or update schedule. However, given the complexity of the processes, as exhibited in Figure 7.2, it is difficult to provide general conditions without taking into account some model structure.

■ 7.6.2 Exact local block samples

Convergence analysis simplifies greatly in the case where exact block samples are drawn at each processor because q is sufficiently large or because another exact sampler [90, 29] is used on each processor. This regime of Hogwild Gibbs sampling is particularly interesting because it minimizes communication between processors.

In (7.5.2), we see that as $q \to \infty$ we have $T \to T_{\text{block}}$; that is, the deterministic part of the update becomes the block Jacobi update map, which admits a natural sufficient condition for convergence:

Proposition 7.6.8. *If* $((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^2 \prec I$, *then block Gaussian Hogwild Gibbs sampling converges.*

Proof. Since similarity transformations preserve eigenvalues, with $\bar{A} \triangleq (B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}}$ we have $\rho(T_{\text{block}}) = \rho((B-C)^{\frac{1}{2}}(B-C)^{-1}A(B-C)^{-\frac{1}{2}}) = \rho(\bar{A})$ and since $\bar{A}$ is symmetric $\bar{A}^2 \prec I \Rightarrow \rho(\bar{A}) < 1 \Rightarrow \rho(T_{\text{block}}) < 1$. □

■ 7.7 Variances

Since we can analyze Gaussian Hogwild Gibbs sampling as a linear dynamical system, we can write an expression for the steady-state covariance Σ_{Hog} of the process when it is stable. For a general stable LDS of the form $x^{(t+1)} = Tx^{(t)} + v^{(t)}$ with $v^{(t)} \sim \mathcal{N}(0, \Sigma_{\text{inj}})$ where Σ_{inj} is the injected noise of the system, the steady-state covariance is given by the series $\sum_{t=0}^{\infty} T^t \Sigma_{\text{inj}} T^{t\mathsf{T}}$, which is the solution to the linear discrete-time Lyapunov equation $\Sigma = T\Sigma T^{\mathsf{T}} + \Sigma_{\text{inj}}$ in Σ [20, 105].

The injected noise Σ_{inj} for the the Hogwild iterations is determined by the inner iterations, which itself is a linear dynamical system with injected noise covariance D, the diagonal of J. For Hogwild sampling we have $\Sigma_{\text{inj}} = (I - T_{\text{ind}}^q)(B-C)^{-1}D(B-C)^{-1}(I - T_{\text{ind}}^q)^{\mathsf{T}}$. The target covariance is $J^{-1} = (B-C-A)^{-1}$.

Composing these expressions we see that the Hogwild covariance is complicated in general, but we can analyze some salient properties in at least two regimes of particular

Algorithm 7.2 Hogwild Gibbs with Symmetric Local Sweeps

Input: Joint distribution over $x = (x_1, \ldots, x_n)$, partition $\{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $\{1, 2, \ldots, n\}$

```
Initialize x̄^(1)
for t = 1, 2, ... do
    for k = 1, 2, ..., K in parallel do
        x̄_{I_k}^(t+1) ← LocalGibbs(x̄^(t), I_k, q(t, k))
function LocalGibbs(x̄, I, q)
    for j = 1, 2, ..., q do
        for i ∈ I in order do
            x̄_i ← sample x_i | x_{¬i} = x̄_{¬i}
        for i ∈ I in reverse order do
            x̄_i ← sample x_i | x_{¬i} = x̄_{¬i}
    return x̄
```

interest: first when A is small so that higher-order powers of A can be ignored, and second when local processors draw exact block samples (e.g. when $q \to \infty$).

■ 7.7.1 Low-order effects in A

Intuitively, the Hogwild strategy works best when cross-processor interactions are small, and so it is natural to analyze the case when A is small and we can discard terms that include powers of A beyond first or second order. To provide an analysis for the low-order regime, we first describe a variant of the Hogwild Gibbs algorithm that enables more detailed spectral analysis. We also fix notation for derivatives. The results in this subsection assume that the Hogwild process is convergent, which is guaranteed for small enough A by continuity of the spectral radius.

For the remainder of Section 7.7.1 we analyze a slight variant of the Hogwild Gibbs algorithm in which processor-local Gibbs update sweeps are performed once in order and once in reverse order for each local iteration, as shown in Algorithm 7.2. This variant is more amenable to spectral analysis because its corresponding inner splitting has more structure than the Gauss-Seidel inner splitting of Algorithm 7.1. To see the difficulty with the Gauss-Seidel inner splitting, consider the splitting

$$\begin{pmatrix} 1 & 0.7 & \\ 0.7 & 1 & 0.7 \\ & 0.7 & 1 \end{pmatrix} \xrightarrow{\text{G-S}} \begin{pmatrix} 1 & & \\ 0.7 & 1 & \\ & 0.7 & 1 \end{pmatrix}^{-1} \begin{pmatrix} & 0.7 & \\ & & 0.7 \\ & & \end{pmatrix} = \begin{pmatrix} 0 & 0.7 & \\ & 0.7^2 & 0.7 \\ & 0.7^3 & 0.7^2 \end{pmatrix}. \quad (7.7.1)$$

This Gauss-Seidel update is not diagonalizable; its Jordan form is

$$\begin{pmatrix} 0 & 0.7 & \\ & 0.7^2 & 0.7 \\ & 0.7^3 & 0.7^2 \end{pmatrix} = \begin{pmatrix} -0.35 & \frac{5}{14} & -\frac{5}{14} \\ 0 & 0.5 & 0.5 \\ 0 & 0.35 & -0.35 \end{pmatrix} \begin{pmatrix} 0 & 1 & \\ & 0 & \\ & & 0.98 \end{pmatrix} \begin{pmatrix} -0.35 & \frac{5}{14} & -\frac{5}{14} \\ 0 & 0.5 & 0.5 \\ 0 & 0.35 & -0.35 \end{pmatrix}^{-1} \tag{7.7.2}$$

and so there is no basis of eigenvectors for the invariant subspace with eigenvalue 0. In general, a Gauss-Seidel update matrix may not be diagonalizable, and little can be said about its eigenvalues.

The inner splitting update matrix for Algorithm 7.2 is that of symmetric Gauss-Seidel, or Symmetric Successive Over-Relaxation (SSOR) with unit relaxation parameter [22]. This update has much clearer spectral properties, as we show in the following lemma. This lemma extends slightly a standard result that the eigenvalues of the SSOR update are real [22, p. 299].

Lemma 7.7.1. *Let $J \succ 0$ and let $J = D - L - L^\mathsf{T}$, where D is the diagonal of J and L and L^T are its strictly lower- and upper-triangular parts, respectively. The symmetric Gauss-Seidel update matrix*

$$(D - L^\mathsf{T})^{-1} L (D - L)^{-1} L^\mathsf{T} \tag{7.7.3}$$

is diagonalizable, and furthermore its eigenvalues are real and in $[0, 1)$.

Proof. We first show (7.7.3) is similar to a positive semidefinite matrix whenever J is symmetric. By applying the similarity transformation $X \mapsto P^{-1} X P$ where $P \triangleq (D - L)^{-1} D^{\frac{1}{2}}$, we see (7.7.3) has the same eigenvalues as

$$D^{-\frac{1}{2}} L^\mathsf{T} (D - L^\mathsf{T})^{-1} D^{\frac{1}{2}} D^{-\frac{1}{2}} L (D - L)^{-1} D^{\frac{1}{2}} = YZ \tag{7.7.4}$$

where $Y \triangleq D^{-\frac{1}{2}} L^\mathsf{T} (D - L^\mathsf{T})^{-1} D^{\frac{1}{2}}$ and $Z \triangleq D^{-\frac{1}{2}} L (D - L)^{-1} D^{\frac{1}{2}}$. Note that

$$L^\mathsf{T} (D - L^\mathsf{T})^{-1} = (D - (D - L^\mathsf{T}))(D - L^\mathsf{T})^{-1} = D(D - L^\mathsf{T})^{-1} - I \tag{7.7.5}$$

and similarly $L(D - L)^{-1} = D(D - L)^{-1} - I$. Hence

$$Z = D^{-1/2} L (D - L)^{-1} D^{1/2} = D^{1/2} (D - L)^{-1} D^{1/2} - I \tag{7.7.6}$$

$$= \left[D^{1/2} (D - L^\mathsf{T})^{-1} D^{1/2} - I \right]^\mathsf{T} = \left[D^{-1/2} L^\mathsf{T} (D - L^\mathsf{T})^{-1} D^{1/2} \right]^\mathsf{T} = Y^\mathsf{T} \tag{7.7.7}$$

and so $YZ = YY^\mathsf{T}$ is positive semidefinite and has nonnegative (real) eigenvalues. Furthermore, when $J \succ 0$ the eigenvalues have absolute value less than unity because symmetric Gauss-Seidel is convergent on positive definite systems. □

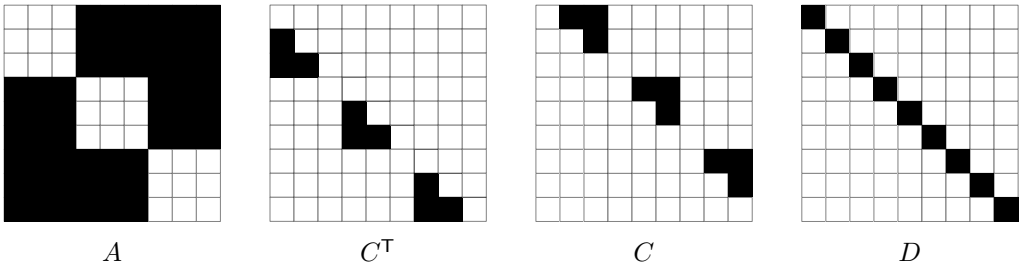


Figure 7.3: Support pattern (in black) of the splitting for Hogwild Gibbs with symmetric local sweeps, $J = D - C^\mathsf{T} - C - A$, with $n = 9$ and the processor partition $\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$.

To model Algorithm 7.2 with its symmetric Gauss-Seidel inner splitting, given a precision matrix J we split $J = D_{\text{bd}} - A$ into block diagonal and block off-diagonal parts as in Section 7.5, then further split $D_{\text{bd}} = D - C^\mathsf{T} - C$ into diagonal, strictly lower-triangular, and strictly upper-triangular parts. Note that $B = D - C^\mathsf{T}$, and so compared to the splitting presented in Section 7.5, we now split $J = D - C^\mathsf{T} - C - A$ instead of $J = B - C - A$. Additionally, though we have $B - C = D - C^\mathsf{T} - C$, in the equations in this section we continue to use $B - C$ for simplicity and consistency with other sections. See Figure 7.3 for an example of the sparsity pattern of A, C^T, C, and D, and compare to Figure 7.1.

The inner-splitting update matrix for Algorithm 7.2 is then the block-diagonal matrix $S \triangleq (D-C)^{-1}C^\mathsf{T}(D-C^\mathsf{T})^{-1}C$. Comparing to (7.6.4), the deterministic part of the update map becomes

$$T \triangleq S^q + (I - S^q)T_{\text{bl}} \tag{7.7.8}$$

for the same definition of T_{bl} as in (7.6.5), and the discrete-time Lyapunov equation for the Hogwild covariance remains

$$\Sigma_{\text{Hog}} = T\Sigma_{\text{Hog}}T^\mathsf{T} + \Sigma_{\text{inj}} \tag{7.7.9}$$

where now $\Sigma_{\text{inj}} = (I-S^q)(B-C)^{-1}D(B-C)^{-1}(I-S^q)^\mathsf{T}$. Similarly, in other expressions we can replace each occurrence of $T_{\text{ind}} = B^{-1}C$ with S. In the following analysis, we use the fact that S has a complete basis of eigenvectors and that its eigenvalues are real and lie in $[0,1)$.

Next, we fix notation for derivatives, following the notation used in Pressley [91]. Let $\mathbb{R}^{n\times n}$ denote the space of real $n \times n$ matrices. For a function $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, we write its derivative at $X \in \mathbb{R}^{n\times n}$ as the linear map $D_X f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined by

$$D_X f(Y) \triangleq \frac{d}{dt} f(X + tY)\bigg|_{t=0} \qquad \forall\, Y \in \mathbb{R}^{n\times n} \tag{7.7.10}$$

where the differentiation is performed element-wise. Similarly, we write the second derivative at $X \in \mathbb{R}^{n\times n}$ as the symmetric bilinear form $D_X^2 f : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined[3] by

$$D_X^2 f(Y,Y) = \frac{d^2}{dt^2} f(X+tY)\Big|_{t=0} \qquad \forall Y \in \mathbb{R}^{n\times n}. \tag{7.7.11}$$

Finally, we write the Taylor approximation for f around the point 0 as

$$f(X) = f(0) + D_0 f(X) + \frac{1}{2} D_0^2 f(X,X) + \mathcal{O}(||X||^3) \tag{7.7.12}$$

where $||\cdot||$ is any submultiplicative matrix norm.

To analyze the Hogwild covariance error to low order in A, we write both the exact covariance and the Hogwild covariance as functions of the symmetric matrix A, respectively $\Sigma(A)$ and $\Sigma_{\text{Hog}}(A)$. We write the Taylor expansion of Σ_{Hog} around 0 as

$$\Sigma_{\text{Hog}}(A) = \Sigma(0) + D_0\Sigma_{\text{Hog}}(A) + \frac{1}{2} D_0^2 \Sigma_{\text{Hog}}(A,A) + \mathcal{O}(||A||^3), \tag{7.7.13}$$

where $\Sigma(0) = (B-C)^{-1}$, and compare it to the exact series expansion for the target covariance $\Sigma = J^{-1}$ given by

$$\begin{aligned}
J^{-1} &= [B-C-A]^{-1} && (7.7.14)\\
&= (B-C)^{-\frac{1}{2}} \left[I - (B-C)^{-\frac{1}{2}} A (B-C)^{-\frac{1}{2}}\right]^{-1} (B-C)^{-\frac{1}{2}} && (7.7.15)\\
&= (B-C)^{-\frac{1}{2}} \Big[I + (B-C)^{-\frac{1}{2}} A (B-C)^{-\frac{1}{2}} \\
&\qquad + ((B-C)^{-\frac{1}{2}} A (B-C)^{-\frac{1}{2}})^2 + \cdots\Big] (B-C)^{-\frac{1}{2}} && (7.7.16)\\
&= \Sigma(0) + (B-C)^{-1} A (B-C)^{-1} \\
&\qquad + (B-C)^{-1} A (B-C)^{-1} A (B-C)^{-1} + \mathcal{O}(||A||^3). && (7.7.17)
\end{aligned}$$

In particular, to understand low-order effects in A, we compare the lowest-order terms that disagree in the two expansions.

We measure the total error as $||\Sigma_{\text{Hog}}(A) - \Sigma(A)||_{P,\text{Fro}}$, where

$$||X||_{P,\text{Fro}} \triangleq ||P^{-1} X P^{-\mathsf{T}}||_{\text{Fro}} \quad \text{and} \quad ||X||_{\text{Fro}} \triangleq \text{tr}(X^{\mathsf{T}} X) \tag{7.7.18}$$

and where $P \triangleq (D - C^{\mathsf{T}})^{-1} D^{\frac{1}{2}}$ is the similarity transformation used in the proof of

[3] A symmetric bilinear form R on a vector space V is defined by the quadratic form Q with $Q(u) = R(u,u)$ for all $u \in V$ via the polarization identity $4R(u,v) = Q(u+v) - Q(u-v)$. Thus to define the second derivative it suffices to define the corresponding quadratic form, as in (7.7.11). In our analysis based on Taylor series expansion, we only use the quadratic form.

Lemma 7.7.1. In the following, we analyze the error on the block-diagonal and the error off the block-diagonal separately, decomposing

$$||\Sigma_{\text{Hog}}(A) - \Sigma(A)||_{P,\text{Fro}} = \\ ||\Pi_{\text{bd}}(\Sigma_{\text{Hog}}(A) - \Sigma(A))||_{P,\text{Fro}} + ||\Pi_{\text{obd}}(\Sigma_{\text{Hog}}(A) - \Sigma(A))||_{P,\text{Fro}} \quad (7.7.19)$$

where Π_{bd} and Π_{obd} project to the block-diagonal and off-block-diagonal, respectively, and we have used the fact that P is itself block-diagonal.

Block-diagonal error

To analyze the low-order effects of A on the block-diagonal error, we first differentiate (7.7.9) to write an equation for $D_0\Sigma_{\text{Hog}}(A)$:

$$D_0\Sigma_{\text{Hog}}(A) - S^q D_0\Sigma_{\text{Hog}}(A) S^{q\mathsf{T}} = \widetilde{A}^{(1)} - S^q \widetilde{A}^{(1)} S^{q\mathsf{T}} - (I - S^q)\widetilde{A}^{(1)}(I - S^q)^{\mathsf{T}} \quad (7.7.20)$$

where $\widetilde{A}^{(1)} \triangleq (B-C)^{-1}A(B-C)^{-1} = D_0\Sigma(A)$ is the first-order term in the expansion for the exact covariance in (7.7.16). Note, however, that because A is zero on its block-diagonal, $\Pi_{\text{bd}}(D_0\Sigma_{\text{Hog}}(A)) = 0 = \Pi_{\text{bd}}(\widetilde{A}^{(1)})$ so the first-order terms in both expansions, (7.7.16) and (7.7.13), are identical on the block-diagonal.

To compare second-order terms on the block-diagonal, we differentiate (7.7.9) twice to write an equation for $D_0^2\Sigma_{\text{Hog}}(A)$:

$$\Pi_{\text{bd}}\left(D_0^2\Sigma_{\text{Hog}}(A,A) - S^q D_0^2\Sigma_{\text{Hog}}(A,A) S^{q\mathsf{T}}\right) = 2\Pi_{\text{bd}}\left((I - S^q)\widetilde{A}^{(2)}(I - S^q)^{\mathsf{T}}\right) \quad (7.7.21)$$

where $\widetilde{A}^{(2)} \triangleq (B-C)^{-1}A(B-C)^{-1}A(B-C)^{-1} = \frac{1}{2}D_0^2\Sigma(A,A)$ is the second-order term in the expansion for the exact covariance in (7.7.16). Using (7.7.21) and the fact that S has a complete set of eigenvectors, we can decompose the error in the second-order terms as

$$\left|\left|\Pi_{\text{bd}}\left(\frac{1}{2}D_0^2\Sigma_{\text{Hog}}(A,A) - \widetilde{A}^{(2)}\right)\right|\right|_{P,\text{Fro}}^2 = \sum_{k\in[K]}\sum_{(i,j)\in\mathcal{I}_k^2} |\widetilde{a}_{ij}^{(2)}|^2 f(\lambda_i^q, \lambda_j^q)^2 \quad (7.7.22)$$

where each (λ_i, λ_j) is a pair of eigenvalues of a block of S, and $\widetilde{a}_{ij}^{(2)} \triangleq (Q^{\mathsf{T}}P^{-1}\widetilde{A}^{(2)}P^{-\mathsf{T}}Q)_{ij}$, where Q is the orthogonal matrix such that $Q^{\mathsf{T}}P^{-1}SPQ$ is diagonal. The function $f : (-1,1)^2 \to \mathbb{R}_+$ is defined by

$$f(\lambda_i^q, \lambda_j^q) \triangleq \left|1 - \frac{(1-\lambda_i^q)(1-\lambda_j^q)}{1 - \lambda_i^q\lambda_j^q}\right|. \quad (7.7.23)$$

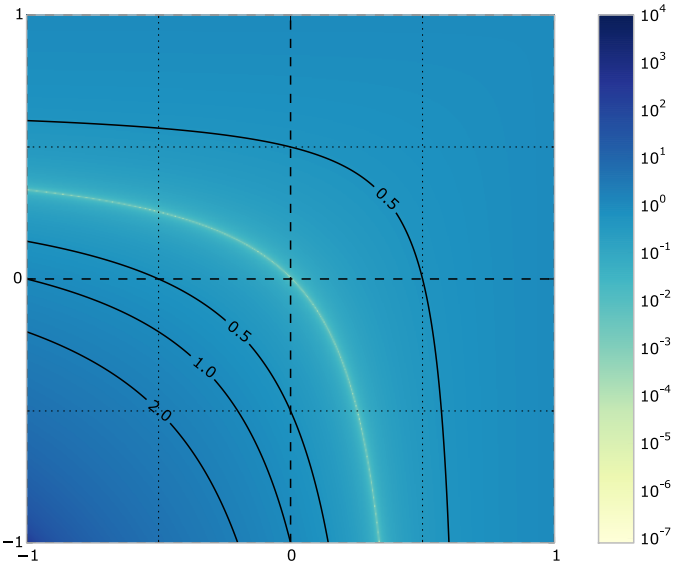


Figure 7.4: A plot of the function f defined in (7.7.23).

Hence we can understand the error by analyzing the values of $f(\lambda_i^q, \lambda_j^q)$ for all the pairs of eigenvalues of S. For a detailed derivation, see Appendix A.

We plot the function f in Figure 7.4. In the figure, the color corresponds to the value of f on a logarithmic scale, and we label some level sets of f. Note in particular that $f(0,0) = 0$ and that $0 \leq f(x,y) < 1$ for $(x,y) \in [0,1)^2$. Due to Lemma 7.7.1, we need only consider the nonnegative quadrant $[0,1)^2$, which corresponds to the upper-right of the figure. Using these properties of f and the decomposition (7.7.22) yields the following proposition.

Proposition 7.7.2. *Using* $||\cdot|| = ||\cdot||_{P,\mathrm{Fro}}$, *we have*

(1) For all $q \geq 1$, *the block-diagonal Hogwild covariance satisfies*

$$||\Pi_{\mathrm{bd}}\left(\Sigma_{\mathrm{Hog}}(A) - \Sigma(A)\right)|| < ||\Pi_{\mathrm{bd}}\left(\Sigma(0) - \Sigma(A)\right)|| + \mathcal{O}(||A||^3); \tag{7.7.24}$$

(2) The dominant (second-order) error term decreases with increasing q *in the sense that*

$$\left|\left|\Pi_{\mathrm{bd}}\left(D_0^2\Sigma_{\mathrm{Hog}}(A,A) - D_0^2\Sigma(A,A)\right)\right|\right| \to 0 \tag{7.7.25}$$

monotonically as $q \to \infty$.

Proof. (1) is immediate from the decomposition (7.7.22), Lemma 7.7.1, and the observation that $0 \leq f(x,y) < 1$ for $(x,y) \in [0,1)^2$. To show (2), first we note that since $\lim_{q\to\infty} \lambda_i^q = 0$ for each eigenvalue λ_i of S and because f is continuous at $(0,0)$, we have that $\lim_{q\to\infty} f(\lambda_i^q, \lambda_j^q) = f(0,0) = 0$ for every pair. Monotonicity follows from the fact that if for any $(x_0, y_0) \in [0,1)^2$ we define a path $\gamma(t) = (x_0^t, y_0^t)^\mathsf{T}$ for $t \in \mathbb{R}_+$, then we have

$$\frac{d}{dt}\gamma(t) = \begin{pmatrix} \ln(x_0)x_0^t & \ln(y_0)y_0^t \end{pmatrix}^\mathsf{T} < 0 \tag{7.7.26}$$

element-wise, and since

$$\nabla f(x,y) = \begin{pmatrix} \frac{(1-y)^2}{(1-xy)^2} & \frac{(1-x)^2}{(1-xy)^2} \end{pmatrix}^\mathsf{T} > 0 \tag{7.7.27}$$

element-wise, we have $\frac{d}{dt} f(\gamma(t)) = \langle \nabla f(x_0^t, y_0^t), \frac{d}{dt}\gamma(t) \rangle < 0$ and f is monotonically decreasing along γ. □

Proposition 7.7.2 shows that, to second order in A, the block-diagonal of the Hogwild covariance is always improved relative to simply ignoring cross-processor effects by approximating $A = 0$, and that the amount of second-order improvement is monotonically increasing with the number of local iterations q.

Off-block-diagonal error

Returning to the first-derivative equation (7.7.20), we see that both $D_0\Sigma_{\text{Hog}}(A)$ and $\widetilde{A}^{(1)}$ are nonzero off the block-diagonal, and therefore to analyze the off-block-diagonal covariance error for small A we compare the first-order terms. Analogous to the argument in the previous section, as derived in Appendix A we can decompose the error in the first-order terms as

$$\left|\left|\Pi_{\text{obd}}\left(D_0\Sigma_{\text{Hog}} - \widetilde{A}^{(1)}\right)\right|\right|_{P,\text{Fro}}^2 = \sum_{\substack{k_1,k_2\in[K]\\ k_1\neq k_2}} \sum_{i\in\mathcal{I}_{k_1}} \sum_{j\in\mathcal{I}_{k_2}} |\widetilde{a}_{ij}^{(1)}|^2 g(\lambda_i^q, \lambda_j^q)^2 \tag{7.7.28}$$

where each (λ_i, λ_j) is a pair of eigenvalues from distinct blocks of S and we set $\widetilde{a}_{ij}^{(1)} \triangleq (Q^\mathsf{T}P^{-1}\widetilde{A}^{(1)}P^{-\mathsf{T}}Q)_{ij}$, where Q is the orthogonal matrix such that $Q^\mathsf{T}P^{-1}SPQ$ is diagonal. The function $g : (-1,1)^2 \to \mathbb{R}_+$ is defined by

$$g(\lambda_i^q, \lambda_j^q) \triangleq \left|\frac{(1-\lambda_i^q)(1-\lambda_j^q)}{1-\lambda_i^q\lambda_j^q}\right|. \tag{7.7.29}$$

We plot the function g in Figure 7.5. In the figure, the color corresponds to the value of g on a logarithmic scale, and we label some level sets of g. Note in particular

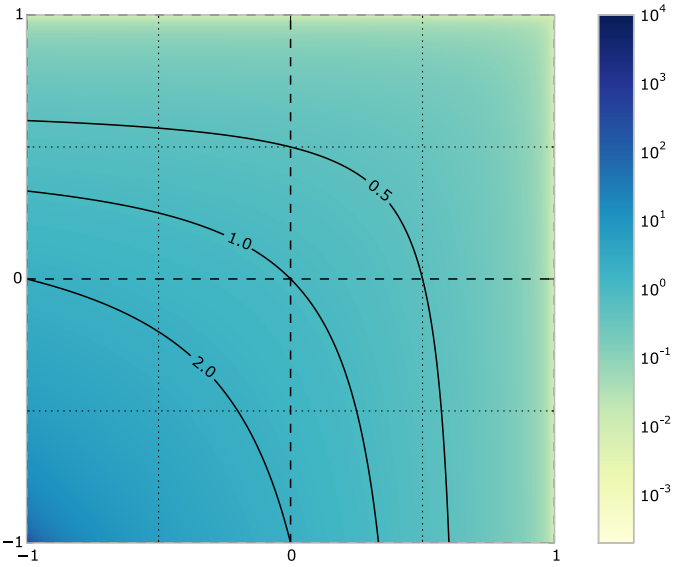


Figure 7.5: A plot of the function g defined in (7.7.29).

that $g(0,0) = 1$ and that $0 \leq g(x,y) < 1$ for $(x,y) \in [0,1)^2$. Using these properties of g and the decomposition (7.7.28) yields the following proposition.

Proposition 7.7.3. *Using* $||\cdot|| = ||\cdot||_{P,\mathrm{Fro}}$, *we have*

(1) For all $q \geq 1$, *the off-block-diagonal diagonal Hogwild covariance satisfies*

$$||\Pi_{\mathrm{obd}}\left(\Sigma_{\mathrm{Hog}}(A) - \Sigma(A)\right)|| \leq ||\Pi_{\mathrm{obd}}\left(\Sigma(0) - \Sigma(A)\right)|| + \mathcal{O}(||A||^2) \tag{7.7.30}$$

where the inequality is strict if S *has a nonzero eigenvalue;*

(2) The dominant (first-order) error term increases with increasing q *in the sense that*

$$||\Pi_{\mathrm{obd}}\left(D_0\Sigma_{\mathrm{Hog}}(A,A) - D_0\Sigma(A,A)\right)|| \to ||\Pi_{\mathrm{obd}}\left(\Sigma(0) - \Sigma(A)\right)|| \tag{7.7.31}$$

monotonically as $q \to \infty$.

Proof. As in the proof for Proposition 7.7.2, (1) follows immediately from the decomposition (7.7.28), Lemma 7.7.1, and the observation that $0 \leq g(x,y) < 1$ for $(x,y) \in [0,1)^2$. To show (2), note that $\lim_{q\to\infty} g(\lambda_i^q, \lambda_j^q) = 1$. By comparing the level sets of g to those

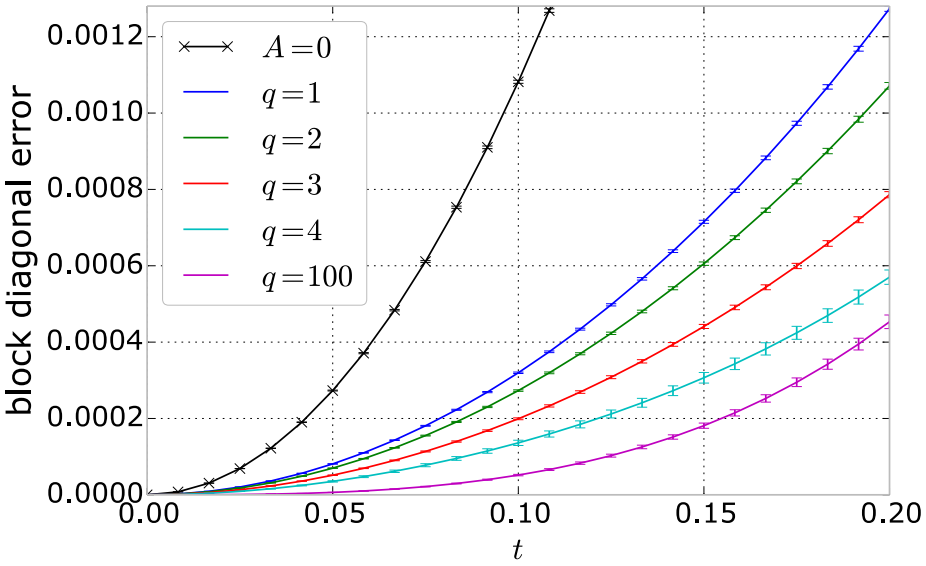
of f, we see that if for any $(x_0, y_0) \in [0,1)^2$ we define $\gamma(t) = (x_0^t, y_0^t)^\mathsf{T}$ for $t \in \mathbb{R}_+$, then we have $\frac{d}{dt} g(\gamma(t)) = \langle \nabla g(x_0^t, y_0^t), \frac{d}{dt}\gamma(t) \rangle > 0$ and so g is monotonically increasing along the path γ. □

Proposition 7.7.3 shows that, to first order in A, the off-block-diagonal of the Hogwild covariance is always an improvement relative to the $A = 0$ approximation (assuming S has a nonzero eigenvalue), yet the amount of first-order improvement is monotonically decreasing with the number of local iterations q. Therefore there is a tradeoff in covariance performance when choosing q, where larger values of q improve the Hogwild covariance on the block diagonal but make worse the covariance error off the block diagonal, at least to low order in A.

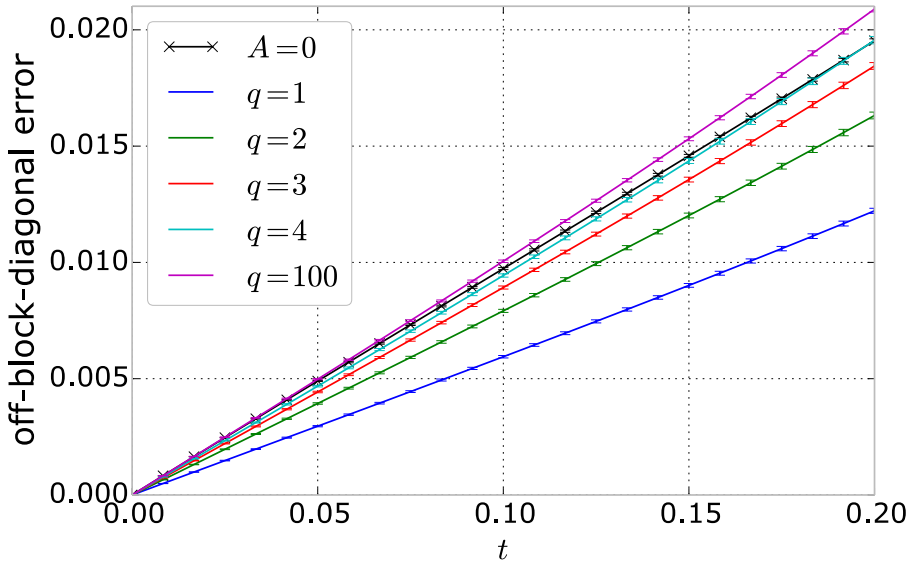
We validate these qualitative findings in Figure 7.6. Model families parameterized by t are generated by first sampling $J = B - C - A = QQ^\mathsf{T}$ with $Q_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ where Q is $n \times n$ and then letting $J(t) = B - C - tA$, so that $t = 0$ corresponds to a model with zero off-block-diagonal entries and off-block-diagonal effects increase with t. The sampling procedure is repeated 10 times with $n = 150$ and a partition with $K = 3$ and each $|\mathcal{I}_k| = 50$. The plotted lines show the average error (with standard deviation error bars) between the block diagonal of the true covariance $\Sigma(t) = J(t)^{-1}$ and the block diagonal of the Hogwild covariance $\Sigma_{\text{Hog}}(t)$ as a function of t for $q = 1, 2, 3, 4, 100$, where varying q shows the effect of local mixing rates. That is, in Figure 7.6(a) each line plots the block-diagonal error $||\Pi_{\text{bd}}(\Sigma(t) - \Sigma_{\text{Hog}}(t))||_{P,\text{Fro}}$ and in Figure 7.6(b) each line plots the off-block-diagonal error $||\Pi_{\text{obd}}(\Sigma(t) - \Sigma_{\text{Hog}}(t))||_{P,\text{Fro}}$. Note that separate black line is plotted for $||\Pi_{\text{bd}}(\Sigma(t) - \Sigma(0))||_{P,\text{Fro}}$ and $||\Pi_{\text{obd}}(\Sigma(t) - \Sigma(0))||_{P,\text{Fro}}$, respectively; that is, the black lines plot the respective errors when ignoring cross-processor effects and approximating $A = 0$.

Figure 7.6(a) shows that to first order in t the block diagonal of the process covariance Σ_{Hog} is identical to the true covariance Σ, since all slopes are zero at $t = 0$. Second-order effects contribute to improve the Hogwild covariance relative to the $A = 0$ approximation. Furthermore, we see that the second-order effects result in lower errors on the block diagonal when there is more processor-local mixing, i.e. larger values of q. Similarly, Figure 7.6(b) shows that first-order effects contribute to improve the Hogwild off-block-diagonal covariance relative to the $A = 0$ approximation. The Hogwild slopes at $t = 0$ are lower than that of the $A = 0$ approximation, and the relative improvement decreases monotonically as q grows and the slopes approach that of the $A = 0$ approximation. These features and their dependence on q are described in general by Propositions 7.7.2 and 7.7.3.

Figure 7.6(b) also shows that, for larger values of t, higher-order terms contribute to make the Hogwild off-block-diagonal covariance error larger than that of the $A = 0$ approximation, especially for larger q. The setting where q is large and global commu-



(a) Π_{bd} projects to the block diagonal



(b) Π_{obd} projects to the off-block-diagonal

Figure 7.6: Typical plots of the projected error $||\Pi(\Sigma(t) - \Sigma_{\text{Hog}}(t))||_{P,\text{Fro}}$ for random model families of the form $J(t) = B - C - tA$. In (a) Π projects to the block diagonal; in (b) Π projects to the off-block-diagonal. The sampled models had $\rho(S) \approx 0.67$. Hogwild covariances were computed numerically by solving the associated discrete time Lyapunov equation [111, 112].

nication is infrequent is of particular interest because it reflects greater parallelism (or an application of more powerful local samplers [90, 29]). In the next subsection we show that this case admits a special analysis and even an inexpensive correction to recover asymptotically unbiased estimates for the full covariance matrix.

■ 7.7.2 Exact local block samples

As local mixing increases, e.g. as $q \to \infty$ or if we use an exact block local sampler between global synchronizations, we are effectively sampling in the block lifted model of Eq. (7.3.13) and therefore we can use the lifting construction to analyze the error in variances.

Proposition 7.7.4. *When local block samples are exact, the Hogwild covariance* Σ_{Hog} *satisfies*

$$\Sigma = (I + (B-C)^{-1}A)\Sigma_{\text{Hog}} \quad \textit{and} \quad ||\Sigma - \Sigma_{\text{Hog}}|| \leq ||(B-C)^{-1}A||\ ||\Sigma_{\text{Hog}}|| \tag{7.7.32}$$

where $\Sigma = J^{-1}$ *is the exact target covariance and* $||\cdot||$ *is any submultiplicative matrix norm. In particular, we may compute*

$$\Sigma = \Sigma_{\text{Hog}} + (B-C)^{-1}A\Sigma_{\text{Hog}} \tag{7.7.33}$$

as a correction which requires only a large matrix multiplication and solving the processor-local linear systems because $B - C$ *is block-diagonal.*

Proof. Using the block lifting in (7.3.13), the Hogwild process steady-state covariance is the marginal covariance of half of the lifted state vector, so using Schur complements we can write

$$\begin{aligned}
\Sigma_{\text{Hog}} &= ((B-C) - A(B-C)^{-1}A)^{-1} && (7.7.34)\\
&= (B-C)^{-\frac{1}{2}}\left[I + ((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^2\right.\\
&\qquad \left. +((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^4 + \cdots\right](B-C)^{-\frac{1}{2}}. && (7.7.35)
\end{aligned}$$

We can compare this series to the exact expansion in (7.7.16) to see that Σ_{Hog} includes exactly the even powers, so therefore

$$\begin{aligned}
\Sigma - \Sigma_{\text{Hog}} &= (B-C)^{-\frac{1}{2}}\left[((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})\right.\\
&\qquad \left. +((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^3 + \cdots\right](B-C)^{-\frac{1}{2}} && (7.7.36)\\
&= (B-C)^{-1}A\Sigma_{\text{Hog}}. && (7.7.37)
\end{aligned}$$

□

Note that this result does not place any assumptions on the off-block-diagonal A.

■ 7.8 Summary

We have introduced a framework for understanding Gaussian Hogwild Gibbs sampling and shown some results on the stability and errors of the algorithm, including (1) quantitative descriptions for when a Gaussian model is not too dependent to cause Hogwild sampling to be unstable (Proposition 7.6.2, Theorems 7.6.6 and 7.6.7, Proposition 7.6.8); (2) given stability, the asymptotic Hogwild mean is always correct (Proposition 7.6.1); (3) in the low-order regime with small cross-processor interactions, there is a tradeoff between the block-diagonal and off-block-diagonal Hogwild covariance errors (Propositions 7.7.2 and 7.7.3); and (4) when local samplers are run to convergence we can bound the error in the Hogwild variances and even efficiently correct estimates of the full covariance (Proposition 7.7.4). We hope these ideas may be extended to provide further insight into Hogwild Gibbs sampling, in the Gaussian case and beyond.