Analyzing Hogwild! Gaussian Gibbs Sampling

Matt Johnson   James Saunderson   Alan Willsky

January 30, 2014

Overview

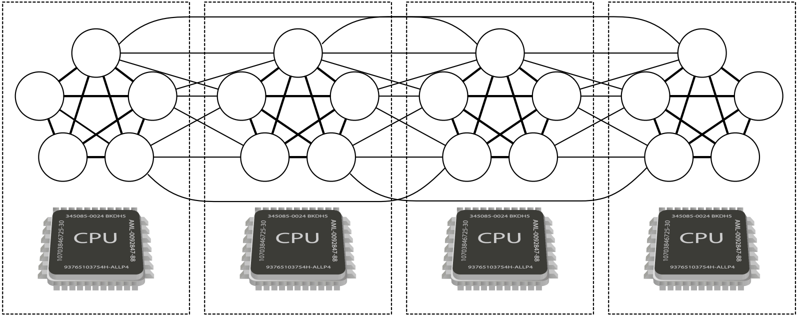
- Inference in dense graphical models is hard to parallelize
- Simply running Gibbs updates in parallel can be very effective
  - going "Hogwild!"[1]
  - but no theory!
- We analyze the Gaussian case
- Connections to numerical linear algebra and general results on synchronous and asynchronous methods[2]

---

[1] F. Niu et al. (2011). "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent". In: *Advances in Neural Information Processing Systems*.

[2] Dimitri P Bertsekas and John N Tsitsiklis (1989). *Parallel and distributed computation*. Old Tappan, NJ (USA); Prentice Hall Inc.

Gibbs sampling in dense graphs



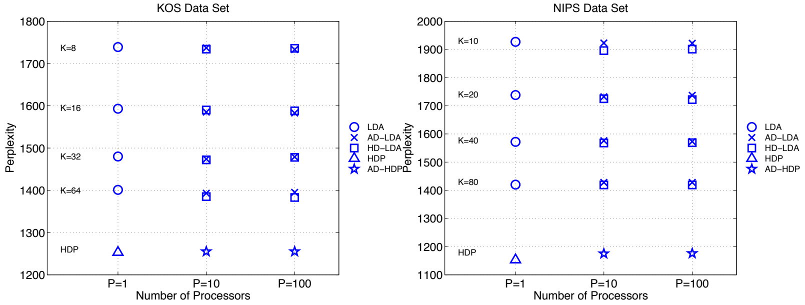
- Without structure, variables must be resampled sequentially
- What if we just run parallel updates anyway. . . ?

Going "Hogwild!" with Gibbs

Require: Data distributed on K processors

```
Initialize latent variables
for ℓ = 1, 2, ... until convergence do
    Communicate global statistics
    for each processor k = 1, 2, ..., K in parallel do
        Run q(k, ℓ) local Gibbs steps on processor k
```

Hogwild! Gibbs on LDA



- Figure reproduced from Newman et al.[3]
- Very effective at fitting LDA topic models on real data

[3] D. Newman et al. (2009). "Distributed algorithms for topic models". In: *The Journal of Machine Learning Research* 10, pp. 1801–1828.

Analysis?

- Hogwild! Gibbs sometimes works!
  - at least for one interesting model...
  - at least for the datasets that were tried...
- When should we expect it to work? Can we analyze it?
- Start by analyzing Gaussian distributions

Gibbs for Gaussians

Goal: Given (J, h) where $J^{-1} = \Sigma$ and $J\mu = h$, sample $x \sim \mathcal{N}(\mu, \Sigma)$

Note: Computing μ is solving a linear system

Gibbs for Gaussians

Goal: Given (J, h) where $J^{-1} = \Sigma$ and $J\mu = h$, sample $x \sim \mathcal{N}(\mu, \Sigma)$

Note: Computing μ is solving a linear system

- **Gibbs sampling** iterates linear Gaussian updates

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

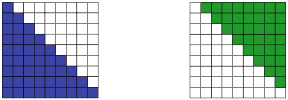
i.e. $x_i \leftarrow \frac{1}{J_{ii}}(h_i - J_{i\neg i}\bar{x}_{\neg i}) + v_i$ where $v_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \frac{1}{J_{ii}})$.

Gaussian Gibbs and Gauss-Seidel

- We can write one Gibbs sweep as

$$x^{(t+1)} = M^{-1}Nx^{(t)} + M^{-1}h + v^{(t)}$$

where $J = M - N$ and $v^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, M^{\mathsf{T}} + N)$



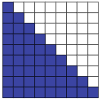
[4] P-regular

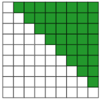
Gaussian Gibbs and Gauss-Seidel

- We can write one Gibbs sweep as

$$x^{(t+1)} = M^{-1}Nx^{(t)} + M^{-1}h + v^{(t)}$$

where $J = M - N$ and $v^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, M^{\mathsf{T}} + N)$





- Expectation is **Gauss-Seidel** on $J\mu = h$
- Gauss-Seidel + diagonal noise = Gaussian Gibbs sampler

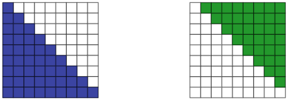
[4] P-regular

Gaussian Gibbs and Gauss-Seidel

- We can write one Gibbs sweep as

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

where $J = M - N$ and $v^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, M^{\mathsf{T}} + N)$

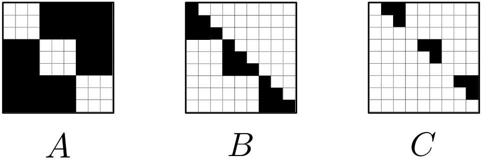


- Expectation is **Gauss-Seidel** on $J\mu = h$
- Gauss-Seidel + diagonal noise = Gaussian Gibbs sampler
- splitting-based[4] iterative solver + noise = Gaussian sampler

[4] P-regular

Hogwild! Gaussian Gibbs as linear dynamics

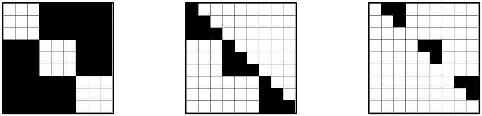
- Split $J = A + B + C$ intra- and inter-processor potentials



[5] A. Frommer and D.B. Szyld (1994). "Asynchronous two-stage iterative methods". In: *Numerische Mathematik* 69.2, pp. 141–153.

Hogwild! Gaussian Gibbs as linear dynamics

- Split $J = A + B + C$ intra- and inter-processor potentials



A B C

- Hogwild! Gibbs dynamics are

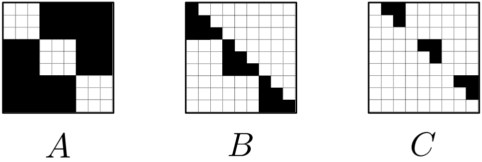
$$x^{(t+1)} = (B^{-1}C)^q x^{(t)} + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1} \left(A x^{(t)} + h + v^{(t,j)} \right)$$

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

[5] A. Frommer and D.B. Szyld (1994). "Asynchronous two-stage iterative methods". In: *Numerische Mathematik* 69.2, pp. 141–153.

Hogwild! Gaussian Gibbs as linear dynamics

- Split $J = A + B + C$ intra- and inter-processor potentials



- Hogwild! Gibbs dynamics are

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

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

- Expectation is the update of a "two-stage"[5] linear solver

[5] A. Frommer and D.B. Szyld (1994). "Asynchronous two-stage iterative methods". In: *Numerische Mathematik* 69.2, pp. 141–153.

Results: stability and means

Prop. 1. *If stable then* $\mu_{hog} = \mu$ *(satisfies fixed-point)*

- But when can we guarantee stability?

Results: stability and means

Prop. 1. *If stable then* $\mu_{hog} = \mu$ *(satisfies fixed-point)*

- But when can we guarantee stability?

Theorem 1. *If there exists diagonal* R *such that*

$$(JR)_{ii} \geq \sum_{j \neq i} |(JR)_{ij}|$$

then for any h*, processor partition, and any number of local iterations* q*, Hogwild Gibbs is stable when run on* (J, h)*.*

Results: stability and means

Prop. 1. *If stable then* $\mu_{hog} = \mu$ *(satisfies fixed-point)*

- But when can we guarantee stability?

Theorem 1. *If there exists diagonal* R *such that*

$$(JR)_{ii} \geq \sum_{j \neq i} |(JR)_{ij}|$$

then for any h*, processor partition, and any number of local iterations* q*, Hogwild Gibbs is stable when run on* (J, h)*.*

- Implies stability for **diagonally dominant**, **walk-summable**, and **latent tree** models
- Reminiscent of Hogwild! SGD condition (Niu et al., 2011)

Results: exact local samples

- What if local samplers converge between global syncs?
- Simple stability condition from **block bipartite lifting**
- Allows **inexpensive correction** to covariance estimate (but not samples)

Results: exact local samples

- What if local samplers converge between global syncs?
- Simple stability condition from **block bipartite lifting**
- Allows **inexpensive correction** to covariance estimate (but not samples)

Prop. 4. *With exact local samples, stable if*

$$((B - C)^{-\frac{1}{2}} A (B - C)^{-\frac{1}{2}})^2 \prec I$$

Prop. 5. *If we run local samplers to convergence, then*

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

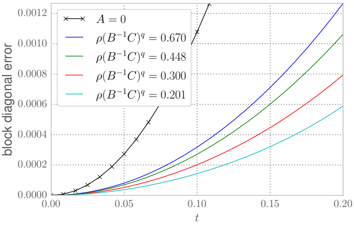
$$||\Sigma - \Sigma_{Hog}|| \leq ||(B - C)^{-1} A|| \; ||\Sigma_{Hog}||$$

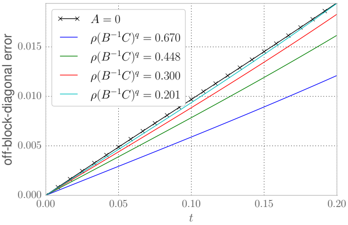
Results: covariances when interactions are small

- Linearized analysis for error in covariance with **small** A
- **Tradeoff** between local mixing and inter-processor covariances:

Block diagonal cov. entries not affected to first order

Off-block diagonal cov. entries degrade with local mixing





Summary

- Gaussian analysis framework and easy proofs
- A new reason to love diagonal dominance
- Can say some things about async case too
- See the paper[6] for more!

[6] Matthew J. Johnson et al. (2013). "Analyzing Hogwild Parallel Gaussian Gibbs Sampling". In: *Advances in Neural Information Processing Systems 26*, pp. 2715–2723.