Imperial College London

PETUUM

Asynchronous Markov Chain Monte Carlo and Gibbs Sampling

Alexander Terenin (Petuum, Inc. and Imperial College London) · Joint work with Dan Simpson (University of Toronto), David Draper (University of California, Santa Cruz), and Eric P. Xing (Petuum, Inc. and Carnegie Mellon University)

Abstract

Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling are widely used for training Bayesian models. Parallelizing these methods for faster computation and distributed deployment is an area of active research. We present a theoretical framework for convergence analysis of MCMC methods executed asynchronously.

Markov Chain Monte Carlo

$$x_0 \xrightarrow{P} x_1 \xrightarrow{P} x_2 \xrightarrow{P} x_3 \xrightarrow{P} x_4 \xrightarrow{P} x_5 \xrightarrow{P} x_6 \xrightarrow{P} x_7$$

Two Theoretical Perspectives

A Markov Chain can be viewed in two ways: as a random algorithm on a fixed state, or a fixed algorithm on a random state.

(1) *Markov transition kernel*: $P : \Omega \times \mathscr{F} \to [0,1]$,

$$P(x,\cdot) = \mathbb{P}(X \in \cdot \mid X = x).$$

For a given $x_k \in \Omega$, the next state x_{k+1} is a *random variable*.

(2) *Markov operator*: $P : \mathscr{M} \to \mathscr{M}$,

$$P(\mu) = \int_\Omega P(x,\cdot) \,\mathrm{d}\mu(x).$$

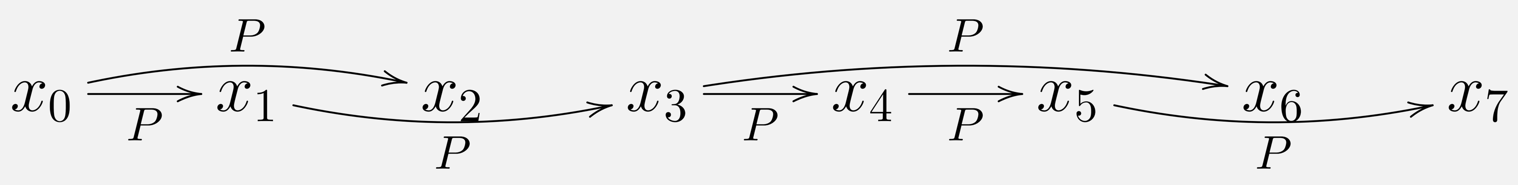
For a given $\mu_k \in \mathscr{M}$, the next measure μ_{k+1} is *deterministic*.

Some properties of valid MCMC methods.

– Convergence: $P(\pi) = \pi$ and $\lim_{k\to\infty} P^k(\mu) = \pi$.

– Monotonicity: $||P^{k+1}(\mu) - \pi||_{\mathrm{TV}} \leq ||P^k(\mu) - \pi||_{\mathrm{TV}}$.

Asynchronous MCMC with shared memory



Convergence Theory

Algorithm 1. For a set of *threads* do the following in parallel.

1. *Read* a value x_k from shared memory.
2. *Update* the value by computing x_{k+1} using any MCMC method.
3. *Write* x_{k+1} to shared memory.

Algorithm 1 is *not a Markov chain*.

– Analyzing using perspective (1) is unhelpful, so use perspective (2).

Theorem: Algorithm 1 *always converges* [6], assuming the following.

– Underlying MCMC update is valid.

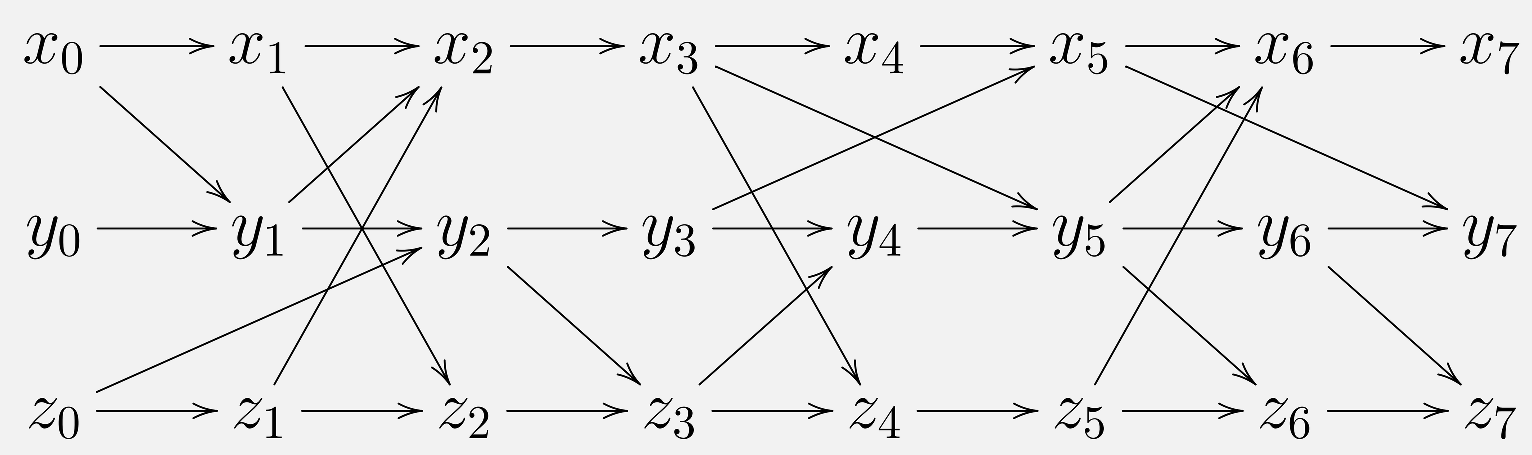
– Maximum time between reads and writes is bounded.

Key idea: by *monotonicity*, applying P can only reduce distance to stationarity, even if relying on out-of-date states.

Related Work

- Asynchronous optimization.
  - Recent analysis: [5], industrial deployment: [2].
- Asynchronous Monte Carlo.
  - Recent analysis: [1, 3, 7], industrial deployment: [4].

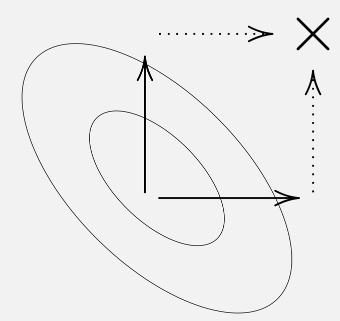
Asynchronous Gibbs Sampling on a compute cluster



Distributed Convergence Theory

Algorithm 2. For a set of *workers*, repeat without synchronization.

1. *Update* the local value by computing x_{k+1} using an MCMC method.
2. *Transmit* x_{k+1} to all other workers, *process* any received messages.



Algorithm 2 *does not converge* in general.

– Shown to diverge for a Gaussian target on $\mathbb{R}^3$ [3].

Problem: updates may interfere with one another.

– Workaround: analysis via *coupled chains*.

Theorem: a coupled Markov operator $H : E \to E$ *converges asynchronously* under appropriate conditions if the following holds.

$$\text{Box Condition: } E = \bigtimes_{i=1}^{m} \mathscr{M}.$$

To satisfy this condition, we use the Metropolis-Hastings method.

– If workers are allowed to *reject messages* in systematic way, then Asynchronous Gibbs sampling *always converges*.

– In real-world problems, *vast majority* of messages are *accepted*, *explaining why* existing methods *perform well* in practice.

References

[1] C. De Sa, K. Olukotun, and C. Ré. Ensuring rapid mixing and low bias for asynchronous Gibbs sampling. *arXiv:1602.07415*, 2016.
[2] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, and E. P. Xing. More effective distributed machine learning via a stale synchronous parallel parameter server. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2013.
[3] M. Johnson, J. Saunderson, and A. Willsky. Analyzing Hogwild parallel Gaussian Gibbs sampling. In *Advances in Neural Information Processing Systems*, pages 2715–2723, 2013.
[4] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10:1801–1828, 2009.
[5] F. Niu, B. Recht, C. Re, and S. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.
[6] A. Terenin and E. P. Xing. Techniques for proving Asynchronous Convergence results for Markov Chain Monte Carlo methods. In *Workshop on Advances in Approximate Bayesian Inference, 31st Conference on Neural Information Processing Systems*, 2017.
[7] A. Terenin, D. Simpson, and D. Draper. Asynchronous Gibbs Sampling. *arXiv:1509.08999*, 2016.

Acknowledgments

We are grateful to Petuum, Inc. for funding this research and providing computational resources. We are grateful to eBay, Inc. for funding previous research, and providing computational resources used in that work. Membership on this list does not imply agreement with the ideas given here, nor responsibility for any errors may be present.

LaTeX TikZposter