Parallel Metropolis-Hastings Algorithms

Boyan Bejanov
bbejanov@bankofcanada.ca

March 2014

1 Title slide

- Parallel Metropolis-Hasting Algorithms
- author info
- course info
- date of presentation

2 Motivation *(1-2 slides)*

- Examples of applications of Markov chain Monte Carlo (MCMC) and Metropolis-Hastings (M-H)

3 Outline

- Background on Markov chains and the M-H algorithm *(4 slides)*
- Parallelization Approaches *(4 slides)*
- Prefetching Algorithm (2 slides)
- Analysis (2-3 slides)
- Implementation details (5-6 slides)
- Computational experiments (4-5 slides)
- Conclusion

4 Markov Chains, definition

- Markovian property
- Transition kernels

5 Markov Chains, properties

- Ergodicity
- Limiting distributions

6 Metropolis-Hastings (2 slides)

- Description of the algorithm

7 Assumptions / Goals

- Target distribution is high-dimensional
- Computation of target density is the most expensive operation
- Chain has long burn-in period
- Algorithm is general, not specific to a narrow class of chains

8 Parallel M-H approaches

- Multiple independent chains: caveats
- Single chain: within-draw vs. between-draw parallelization
- references

9 Parallel M-H approaches (cont'd)

- Independence Metropolis-Hastings
- description
- difficulties: ergodicity problems, feasibility problem
- *(mention adaptation to transition to next slide)*
- references

10 Parallel M-H approaches (cont'd)

- Independent tours
- regeneration times: definition, advantages, difficulties
- references

11 Parallel M-H approaches (cont'd)

- Prefetching
- description *(just the general idea, hand-waving)*
- references

12 The Method of Choice

- Prefetching is the parallelization method of choice: explain why

13 Prefetching

- proper description
- variants: Brockwell 2006, Byrd et. al. 2008, Strid 2010

14 Analysis

- complexity math

15 Analysis (cont'd)

- P-completeness *(1-2 slides max)*

16 Implementation details

- General remarks *(2-3 slides max)*

17 Implementation details

- Multicore take one: Cilk

18 Implementation details

- Multicore take two: OpenMP

19 Implementation details

- Coarse grained cluster: MPI

20 Computational Experiments *(4-5 slides)*

- Tables of run times
- Speedup graphs

21 Conclusion

- Conclusion