LITERATURE REVIEW:
Parallel Metropolis-Hastings Algorithms

Boyan Bejanov
bbejanov@bankofcanada.ca

February 8, 2014

1 Introduction

The Monte Carlo methods (MC) are a class of numerical integration techniques, where the integral is interpreted as the expected value of some function of some random variable. The expectation can be statistically estimated from a sample drawn from the same distribution as the underlying random variable and this estimate serves as the numerical approximation of the integral. Needless to say, the MC methods are also applicable to problems other than numerical integrations, so long as they can be reduced to the evaluation of an expected value.

The successful application of MC relies on the availability of numerical techniques for simulation of random samples from a distribution, which is given by its probability density function (pdf). We will denote this *target* density by $\pi(x)$. We always assume that we have a stream of independent random numbers with uniform distribution in the interval $(0, 1)$. For some of the other standard probability distributions there are direct methods of simulation, which perform some computation on the uniform random numbers, thus transforming them into random numbers of the desired distribution. However, these techniques are limited to special cases. One of the most important general sampling technique is the Acceptance-Rejection sampling (AR). Each iteration of the AR goes in two steps. We first simulate a random number, X, from another distribution, which we can simulate by another known method. We call it *proposal* density and denote by $q(x)$. In the second step, we simulate a uniform random number and compare it to the adjusted ratio of the target and proposal densities. Based on this comparison, we either accept X as a random number from π, or we reject and discard it. The AR method is embarassingly parallel, and produces a sample of independednt random samples, which can be used directly in an MC method. However its applicability is limited to the cases where $\pi(x)$ is given explicitly by a known formula. Even then, the computational efficiency of AR (as measured by the acceptance rate) depends on the choice of the proposal, $q(x)$, and a tight estimate of the adjustment constant in the AR test, both of which can require some non-trivial mathematical derivations, especially for multivatiate distributions.

The method of choice in the cases of a multivariate distribution of high dimension, or when the target distribution formula is too complicated or can only be computed numerically, is the Markov Chain Monte Carlo method (MCMC). A Markov Chain is a process, which is described by a *state space*, i.e. the set of all possible values the process can take, and a transition rule, which may depend only on the current state, but not any earlier ones. It is known from the theory of Markov chains that if certain conditions are satisfied, then the process has an *invariant distribution* and the distributions of successive states converge to the invariant distribution. In the MCMC method, we construct a transition rule, such that the resulting Markov chain has an invariant distribution, and this invariant distribution is exactly $\pi(x)$. The initial part of the chain, before the distribution of states gets close enough to the invariant distribution, is known as *burn-in* and is discarded. The remaining chain can be used as the random sample in a Monte Carlo method, however with the caveat that it is not an independent random sample, so the auto-correlation of the process must to be taken into account.

The Metropolis-Hastings algorithm (M-H) combines AR and MCMC. Effectively, it is a recipe for constructing a Markov chain transition rule. Once again we have a proposal distribution, however this time the proposal distribution depends on the current state of the chain, i.e. $q = q(x, y)$. The good news is that the accept-reject test is a ratio of the values of q and π at the current and the proposed points with no adjustment constants to be derived in advance. If the proposed point is accepted, then it becomes the next state of the chain, otherwise the next state is equal to the current state, i.e. no points are discarded as it was the case in the AR sampling method.

Other than Monte Carlo type estimations of expected values, the M-H is a general method for simulating random samples from distributions that are otherwise impossible to simulate. It is particularly popular for Bayesian analysis of complex stochastic models. Such models always involve a number of parameters, which rarely can be derived from first principles. More frequently, the paremeters are estimated from observed data using maximum-likelihood, or some other methods of statistical estimation. When the model is sufficiently complex, the usual inference based on confidence intervals for the estimated parameters is impossible to deriv. In such cases, the approach of Bayesian statistics is to consider the parameters as random variables, simulate a sample from their joint distribution, and use this sample to infer the properties of the distribution that are of interest.

The specific focus in this project is the application of M-H to Bayesian inference of large Dynamic Stochastic General Equilibrium models (DSGE), which are used in the field of macro-economics to simulate the economy. The likelihood function is computed using a Kalman filter, which can be computationally very expensive, especially in the nonlinear case. Therefore, when considering various methods, we will keep in mind the (simplifying) assumprion that the evaluation of $\pi(x)$ is much more time consuming than all the other computations in a single step (such as the computation of $q(x, y)$, simulation from $q(x, y)$, individual arithmetic operations).

The objectives of this project include

- Survey the existing methods and evaluate them.
- Identify the most promising method, for the particular setting we have, and implement it.
- Investigate the benefits of parallelization in terms of theoretical and practically achieved speedup.
- Consider the question of P-completeness of the M-H algorithm.

2 Literature Review

Give an overview of the relevant literature. Cite all relevant papers, like [2], [7], [3], [6], [5], [1], and [4]. Outline for each paper the relevant results in relation to your project. Make sure that you don't just list all relevant papers in random order. Devise a scheme to group papers by subject. The goal is to present to the reader the state-of-the-art in the field selected for your project.

References

[1] Y. Chen, F. Dehne, T. Eavis, D. Green, A. Rau-Chaplin, and E. Sithirasenan. cgmOLAP: Efficient parallel generation and querying of terabyte size ROLAP data cubes. In *Proc. 22nd Int. Conf. on Data Engineering (ICDE)*, pages 164–164. IEEE Comp. Soc. Dig. Library, 2006.

[2] F. Dehne, T. Eavis, and B. Liang. Compressing data cubes in parallel OLAP systems. *Data Science Journal*, 6:S184–S197, 2007.

[3] F. Dehne, T. Eavis, and A. Rau-Chaplin. The cgmCUBE project: Optimizing parallel data cube generation for ROLAP. *Distributed and Parallel Databases*, 19(1):29–62, 2006.

[4] F. Dehne, M. Fellows, M. Langston, F. Rosamond, and K. Stevens. An $o(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. In *Proc. 11th Int. Computing and Combinatorics Conf. (COCOON)*, pages 859–869. Springer LNCS 3595, 2005.

[5] F. Dehne, M. Langston, X. Luo, S. Pitre, P. Shaw, and Y. Zhang. The cluster editing problem: Implementations and experiments. In *Proc. Int. Workshop on Parameterized and Exact Computation (IWPEC)*, pages 13–24. Springer LNCS 4169, 2006.

[6] M. Lawrence, F. Dehne, and A. Rau-Chaplin. Implementing OLAP query fragment aggregation and recombination for the OLAP enabled grid. In *Proc. International Parallel and Distributed Processing Symposium (IPDPS), High-Performance Grid Computing Workshop*, pages 1–8. IEEE Comp. Soc. Dig. Library, 2007.

[7] S. Pitre, F. Dehne, A. Chan, J. Cheetham, A. Duong, A. Emili, M. Gebbia, J. Greenblatt, M. Jessulat, N. Krogan, X. Luo, and A. Golshani. PIPE: a protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. *BMC Bioinformatics*, 7:365 (15 pages), 2006, available via PubMed at http://www.biomedcentral.com/pubmed/16872538.