PARALLEL MARKOV CHAIN MONTE CARLO COMPUTATION FOR VARYING-DIMENSION SIGNAL ANALYSIS

Jing Ye[1], *Andrew Wallace*[1] *and John Thompson*[2]

Edinburgh Joint Research Institute in Signal and Image Processing
[1] Heriot-Watt University, UK {jy68, A.M.Wallace}@hw.ac.uk
[2] University of Edinburgh, UK John.Thompson@ed.ac.uk

Abstract. Parallel implementation of Markov Chain Monte Carlo (MCMC) algorithms for Bayesian inference has been effective but is usually restricted to the case where the dimension of the parameter vector is fixed. We propose an efficient parallel solution for the varying-dimension problem by constructing multiple within-model MCMC chains and then combining the separate results to analyze the posterior distribution of dimensionality. We aim for parallel speed-up by reducing the length of the burn-in period and the individual chains in comparison with a serial, reversible jump MCMC (RJMCMC) algorithm. The parallel methodology is illustrated with application to a benchmarking, change point problem.

1. INTRODUCTION

Recently, parallel computing has received impetus due to the increasing availability of cheap computing power and networking. The great potential is to split a task into sub-tasks, which are then distributed onto multiple processors and executed concurrently. Consequently, both the computation time and memory space requirement on individual machines are reduced, offering the possibility to cope with previously intractable calculations in a number of application areas. Complex statistical model analysis becomes attractive, and in particular, we are concerned with parallel MCMC implementation in the context of Bayesian inference.

This presents challenges in implementation and statistical analysis. For example, a principal factor affecting parallel performance is inter-processor communication. When increasing the number of processors, the speed may slow down dramatically due to additional communications overhead. Therefore, it is crucial to design algorithms with a relatively low message passing frequency and coarse *granularity* [1]. MCMC is serial by nature and does not easily migrate onto a parallel system; a particular concern is whether a parallel implementation produces the invariant distribution of interest. Moreover, since MCMC estimation is based on trajectory averaging, another significant task is to reduce the correlation among random number streams on separated processors that can arise by assigning identical random number seeds to each machine [1]. Since the serial MCMC sampler is the prototype of parallel MCMC, its statistical properties such as the burn-in period and mixing performance could finally determine the selection of a parallel strategy and the potential for speed-up.

The objective of parallel MCMC is to make use of the conditional independence structure of underlying models combined with parallel computing strategies and MCMC properties. Current algorithms can be classified into two main categories: one is parallelization of a single chain, and the other is parallel generation for multiple different chains. Although some of these methods have shown convincing parallel performance, most of them are restricted to the case where the dimensionality of the parameter vector is fixed. The development of parallel methodologies for varying-dimension signals remains comparatively neglected.

We address parallel computation for Bayesian model selection with respect to the above considerations. The designed method takes advantage of the faster convergence rate present in within-model chains than in a trans-model chain. Speed-up is therefore achieved by running shorter MCMC chains in parallel rather than a serial RJMCMC chain. In Section 2, we introduce MCMC and RJMCMC algorithms associated with corresponding diagnostic methods. In Section 3, our proposed solution for the varying-dimension problem is presented. Section 4 presents the results from a specific benchmarking example, and in Section 5, we summarise our results and suggest future work.

2. MCMC AND RJMCMC ALGORITHMS

2.1 MCMC and Convergence Assessment

The objective of Bayesian inference is to estimate the posterior distribution of a parameter set θ based on the prior knowledge of the statistical model and observation Y using Bayes's formula as:

$$p(\theta|Y) = \frac{p(Y|\theta)p(\theta)}{p(Y)} \tag{1}$$

where $p(\theta)$ is the prior distribution on θ and $p(Y|\theta)$ is the likelihood function.

The basic idea of MCMC analysis is to construct a Markov chain whose limiting distribution is the invariant distribution π of interest, and then estimate the characteristics of π using sample path averages. In the context of Bayesian inference, π is $p(\theta|Y)$. When θ is of fixed dimension, the Metropolis-Hastings algorithm is used as follows:

1. Initialize the Markov chain with θ_0 and set $t = 0$.

2. For $t = 1$ to the maximum number of iterations, repeat (a) to (d) :
   (a) Propose a new sample θ' from $q(\cdot|\theta_t)$.
   (b) Calculate the acceptance probability $\alpha(\theta_t, \theta')$.
   (c) Draw a random variable u from a uniform distribution $U(0,1)$.
   (d) If $u < \alpha(\theta_t, \theta')$, accept the proposed state and set $\theta_{t+1} = \theta'$. Otherwise remain in the current state and set $\theta_{t+1} = \theta_t$

The proposal probability $q(\cdot|\theta_t)$ can be in any form, but may affect the Markov chain mixing performance and convergence rate. The expression of the acceptance probability for Bayesian inference is,

$$\alpha(\theta_t, \theta') = \min\{1, \frac{p(\theta'|Y)q(\theta_t|\theta')}{p(\theta_t|Y)q(\theta'|\theta_t)}\} \tag{2}$$

If the chain is constructed properly, as t goes to infinity, the generated sequence $\{\theta_j\}, j = 1, 2, ...$ should converge to the target distribution π. However, in practice, we can only produce a finite number of samples, and it is therefore important to choose the chain length appropriately and assess the convergence of the Markov chain to the stationary distribution.

The convergence diagnostic presented in [2, 3] compares the samples drawn from several independent sequences with different starting points and quantitatively evaluates the mixing by analyzing the within-sequence and between-sequence variance. The idea is that as the number of samples increases, each individual chain will explore larger parts of the parameter space, and consequently, the overall variance and within-sequence variance will both converge to the true model variance. Assume that we simulate $I > 2$ independent sequences initialized with over dispersed starting points, each of length $2T$ and discard the first T samples treated as the *burn-in* period. For any scalar function $x(\theta)$, we label the t^{th} observation in chain i as x_i^t and calculate the between-sequence variance B:

$$B = \frac{T}{I-1}\sum_{i=1}^{I}(\overline{x_i^{\cdot}} - \overline{x_{\cdot}^{\cdot}})^2 \tag{3}$$

where

$$\overline{x_i^{\cdot}} = \frac{1}{T}\sum_{t=T+1}^{2T} x_i^t, \text{ and } \overline{x_{\cdot}^{\cdot}} = \frac{1}{I}\sum_{i=1}^{I}\overline{x_i^{\cdot}} \tag{4}$$

The within-sequence variance W is estimated by:

$$W = \frac{1}{I}\sum_{i=1}^{I} s_i^2 \tag{5}$$

where

$$s_i^2 = \frac{1}{T-1}\sum_{t=T+1}^{2T}(x_i^t - \overline{x_i^{\cdot}})^2 \tag{6}$$

The variance of x in the target distribution, V is estimated by:

$$\widehat{V} = \frac{T-1}{T}W + (1+\frac{1}{I})\frac{B}{T} \tag{7}$$

The convergence of the Markov chain is monitored by the estimated *potential scale reduction factor* (PSRF),

$$\sqrt{\widehat{R}} = \sqrt{\frac{\widehat{V}}{W}} \tag{8}$$

As $T \to \infty$, the total variance estimation $\widehat{V}$ should decrease while the within-sequence variance W might increase, and finally PSRF should theoretically decline to 1. If $\widehat{R}$ is large, it indicates the posterior distribution should be further explored. Once PSRF close to 1, we assume the Markov chain convergences to the target distribution.

2.2 RJMCMC and Convergence Assessment

In the MCMC approach, the dimension of the parameter space is fixed. However, in many cases, the dimensionality is unknown and hence the joint posterior $p(k, \theta_k|Y)$ of the model indicator k and the parameter vector of model k forms the objective of inference. The RJMCMC algorithm proposed by [4] is an effective solution to this Bayesian model determination problem. RJMCMC is an extension of the Metropolis-Hastings algorithm that allows jumps between models with different dimensions in addition to within-model parameter updates. By satisfying the *dimension-matching* requirement, the relationship between the current and proposal states is defined and the corresponding acceptance probability is derived. Still following the Metropolis-Hastings steps with $p(k, \theta_k|Y)$ as the target distribution, a single serial Markov chain is constructed which explores both the within-model state space and between-model mixing.

Convergence assessment for RJMCMC becomes much more difficult as some models may be seldom visited and hence a convergence diagnostic for these models is almost impossible to estimate. To circumvent this, [5] extended the work of [2] by splitting the variance not only between sequences but also between models. Let x be the scalar summary of the parameters of interest. Suppose the total number of samples of model m in chain i is $K(i,m)$, and x_{im}^k stands for the k^{th} observation of model m in chain i with $i = 1, ..., I, m = 1, ...M$ and $k = 1, ..., K(i,m)$. The total variance of x is estimated by

$$\widehat{V} = \frac{1}{IT-1}\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{m=1}^{M}(x_{im}^k - x_{\cdot\cdot}^{\cdot})^2 \tag{9}$$

where $x_{\cdot\cdot}^{\cdot}$ is the total average for all the samples drawn in I chains. If $x_{i\cdot}^{\cdot}$ is the sample average in chain i, the within-model variance is defined as:

$$W_c = \frac{1}{I}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K(i,m)}\frac{(x_{im}^k - x_{i\cdot}^{\cdot})^2}{\sum_{m=1}^{M}K(i,m) - 1} \tag{10}$$

Then, the variance is split between and within models. The between-model variance B_m, between-model within-chain variance $B_m W_c$; within-model variance W_m and within-model within-chain variance $W_m W_c$ are:

$$B_m = \frac{\sum_{m=1}^{M}(x_{\cdot m}^{\cdot} - x_{\cdot\cdot}^{\cdot})^2}{M-1} \tag{11}$$

$$B_m W_c = \frac{\sum_{i=1}^{I}\sum_{m=1}^{M}(x_{im}^{\cdot} - x_{i\cdot}^{\cdot})^2}{I(M-1)} \tag{12}$$

$$W_m = \frac{1}{M}\sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{k=1}^{K(i,m)}\frac{(x_{im}^{k} - x_{\cdot m}^{\cdot})^2}{\sum_{i=1}^{I}K(i,m)-1} \tag{13}$$

$$W_m W_c = \frac{1}{IM}\sum_{i=1}^{I}\sum_{m=1}^{M}\frac{\sum_{k=1}^{K(i,m)}(x_{im}^{k} - x_{im}^{\cdot})^2}{K(i,m)-1} \tag{14}$$

From these six statistics, three ratios are created, between $\widehat{V}$ and W_c, W_m and $W_m W_c$; B_m and $B_m W_c$ respectively. As the RJMCMC chain is much more complex than the MCMC counterpart, all of the original six parameters are monitored to gain insight into the chain mixing rather than just evaluate how close these three ratios are to 1. The conclusion obtained in [5] is that although $\widehat{V}$ and W_c have stabilized, the other four statistics may show dramatic sudden changes in value and differ significantly from one when one chain visits a rather improbable model. This phenomenon indicates that even though some chains have already approach the steady states, they may have not visited some models. From this point of view, more iterations may be required to further enhance the between-model mixing.

3. PARALLEL MCMC FOR VARYING-DIMENSION PROBLEMS

The considerable merit of RJMCMC for across-model simulation is that the joint posterior inference for (k, θ_k) can be obtained directly from a single serial chain. However, to obtain adequate mixing within and between models, a sufficiently long run is required, as discussed above. Our basic premise is that parallel within-model MCMC chains mix better and converge faster than serial RJMCMC chains, and offer coarse grain parallelism that conforms well to the desirable characteristics of parallel SIMD programming. We would expect the within-model chains to have a shorter burn-in period and chain length as each chain only needs to explore one particular parameter subspace and poor mixing between models is also circumvented. Therefore, each within-model MCMC chain proceeds in parallel, with target distribution $p(\theta_k|k,Y)$, and the results from separate chains are combined to construct the posterior inference for k. The within-model posterior density $p(\theta_k|k,Y)$ is defined by the parallel MCMC runs, and Bayesian model selection is computed by pairwise comparisons,

$$\frac{p(k_1|Y)}{p(k_2|Y)} = \frac{p(k_1)}{p(k_2)}\frac{p(Y|k_1)}{p(Y|k_2)} \tag{15}$$

where the first ratio on the right-hand side is of prior probabilities, and the second ratio is the *Bayes factor* for model k_1 and k_2 [6]. To estimate the posterior model probability $p(k|Y)$, we require the *marginal likelihood* of model k

$$p(Y|k) = \int p(\theta_k, Y|k)\mathrm{d}\theta_k \tag{16}$$

As $p(k|Y)$ is the normalizing factor of the posterior density, we can express $p(Y|k)$ as:

$$p(Y|k) = \frac{p(\theta_k, Y|k)}{p(\theta_k|Y,k)} = \frac{p(Y|k,\theta_k)p(\theta_k|k)}{p(\theta_k|Y,k)} \tag{17}$$

Equation 17 holds for any fixed parameter point of θ_k, say θ_k^*, and now the target becomes the estimation of $p(\theta_k^*|Y,k)$ from MCMC runs since we can easily obtain the conditional prior density $p(\theta_k^*|k)$ and the likelihood value $p(Y|k,\theta_k^*)$.

According to [7, 8], although we can choose any θ_k, parameter points with high density are likely to provide more accurate estimation of $p(\theta_k^*|Y,k)$. Considering the output of parallel MCMC runs, the θ_k^* corresponding to the posterior mode or maximum likelihood estimate can be selected.

In [7, 8], the $p(\theta_k^*|Y,k)$ estimate is a sample average using either the Gibb's sampler or Metropolis-Hastings algorithm. Although these are valid in general, the computation time can grow as the number of mixture components increases since extra MCMC chains are constructed for the marginal posterior estimation. Our alternative method is that when there have been sufficient samples in each parameter subspace as the chains converge to the stationary distribution, we can directly estimate the posterior density $p(\theta_k|Y,k)$ using kernel density estimation techniques.

To summarise, parallel generation of multiple within-model chains has the following advantages. Since the separate MCMC chains are independent, inter-communication is not required and time is not spent in message passing among processors except for initialization and between model selection. Even if computer failures appear, other processors can proceed without interruption; each sequence is still serial, and therefore the desired limiting distribution can be guaranteed. Proper parallelization methods for MCMC chains can also be applied to further speed up single chain generation.

4. EXPERIMENTAL RESULTS

Our motivation for RJMCMC analysis has been the interpretation of LiDAR data for 3D imaging. We have already shown [9] that RJMCMC methods can lead to dramatically improved measurement accuracy, resolution, and sensitivity, but at the cost of much greater complexity. In this paper, we use the coal mining disaster data in [4] as an exemplar, frequently used for multiple change-point problems (see Figure 1). Our goal here is to analyze the joint posterior density for $2k+1$ parameters containing change-point positions and the Poisson rate in model k, with k the number of change points. This allows us to compare our results with other published work on this established benchmark, but also has similarities with the more complex problem of LiDAR analysis [9] in that the final data can be considered as a multiplicity of distinct returns (change points or optical reflections).

We use the RJMCMC sampler of [4] and restrict the maximum number of change points to be 6 for comparison with [6], since the range $k = 1,\ldots,6$ covers most of the posterior probability. To compare the convergence diagnostics and parameter estimations between within-model chains and a

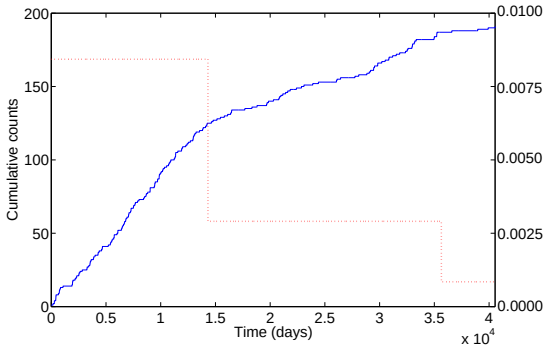

Figure 1: Coal mining disaster data, year 1851-1962: dates of disasters, cumulative counting process (solid curve) and posterior mode of Poisson rate for $k = 2$ (dotted curve)

trans-model chain, the proposal distributions we choose for MCMC samplers are the same as for the RJMCMC chain but prohibit between-model jumps.

4.1 Convergence assessment

In the context of the coal mining disaster problem, the specific target of the RJMCMC sampler is to achieve the Bayesian analysis of both k and the corresponding $2k+1$ parameters. We aim to carry out the convergence assessment on each individual parameter rather than their scalar summary, in which case the convergence diagnostics defined in [5] are not satisfactory. To compare the convergence of RJMCMC and MCMC chains, we refer to the RJMCMC simulation results in [6]. These indicate that the chain has safely converged at 1,000,000 iterations and that 200,000 samples provide very similar posterior density estimation results.

To assess the convergence for MCMC chains, we generate four separate sequences for each model, and analyze the diagnostic statistics defined in [2] every 100 iterations after the burn-in period (500 samples). The chain length is set to be 15,000 and Figure 2 presents two examples of PSRF values for parameter sets in model $k = 2$ and $k = 6$. The parameter solution for $k = 2$ is also shown in Figure 1. Here, the convergence of a MCMC chain means that the PSRF's for all the parameters reduce to less than 1.2.

It is found that for $k = 1, \ldots, 6$, within-model chains converge at 350, 5400, 10300, 9900, 9500 and 14400 separately, which means 15000 is a safe length. The reason that the chains with higher dimension might converge slower than lower dimensional chains is that only one single parameter is updated in each iteration and therefore to provide sufficient samples for each dimension, more iterations are required. Unfortunately, this goes against the load balancing requirement for parallel implementation efficiency. This problem could be overcome by updating one set of parameters instead of one parameter in one iteration, or modifying the proposal distribution to improve the acceptance rate to achieve better mixing performance and in turn shorten the chain length as well as the convergence length differences.

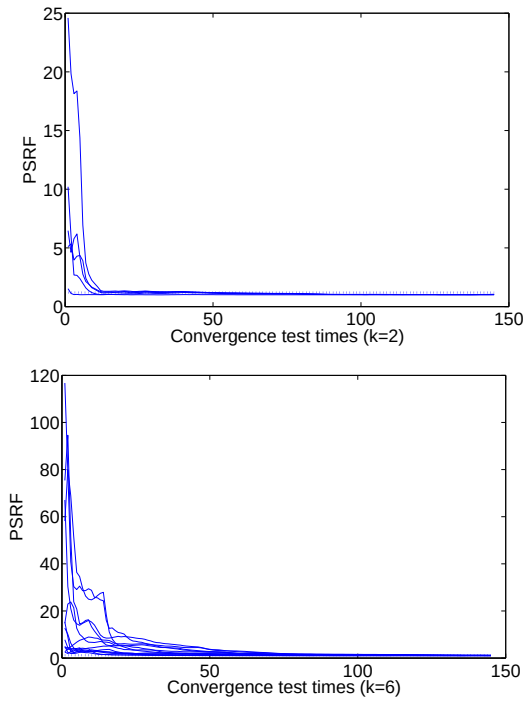

Figure 2: Diagnostic plots of $\widehat{R}$ for $2k+1$ parameters in the within-model MCMC sequences ($k = 2$ and $k = 6$) for the coal mining disaster data. Dotted curve is the convergence threshold with PSRF$= 1.2$

MCMC	k=1	k=2	k=3	k=4	k=5	k=6
	6.7758	6.8051	6.9262	6.7864	6.8514	6.7584
RJMCMC	90.3120					

Figure 3: Timing results for within-model MCMC chains and trans-model RJMCMC chains measured in seconds

4.2 Simulation efficiency

To evaluate the parallel improvement, we set the number of iterations for within-model chains to be 15,000 to achieve more accurate estimation of the posterior density $p(\theta_k|Y,k)$, which is in turn used for the Bayes factor calculation, and the length of RJMCMC chain to be 200,000. The timing results are shown in Figure 3. Using 6 processors in a Beowulf network [10] including 32 Intel Pentium 4 3GHz processors with distributed memories, the speed-up of the longest process $(k = 3)$ is 13.04. Although this appears super-linear, the computations are not exactly equivalent. First the process serial chain performs 2.22 more iterations in total. This would reduce the speed-up to a more plausible 5.87. Moreover, the algorithms are not equivalent, since we cannot predict how much time the serial chain spends in each parameter dimension, so any comparison can only be approximate and may vary from run to run. Nevertheless, there is near-linear parallel advantage due to the coarse grain parallelism.

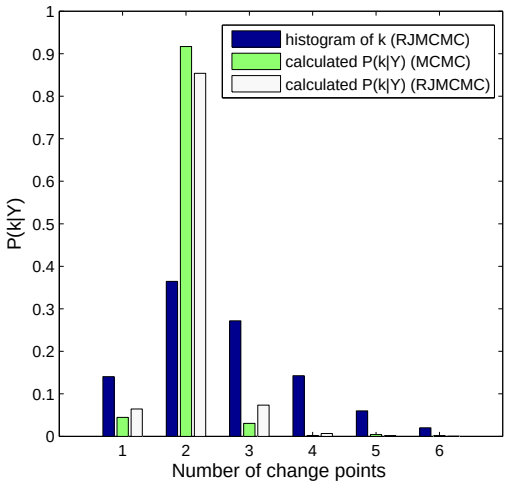

Figure 4: $p(k|Y)$ obtained from RJMCMC histogram, numerical calculation using RJMCMC samples and numerical calculation using MCMC samples

4.3 Posterior distribution of model indicator

Figure 4 compares the results of $p(k|Y)$ obtained from the RJMCMC chain and parallel within-model MCMC chains. First, for the RJMCMC chain, we construct a histogram for k using the direct sampling results. This can be compared to the results presented in [4] and [6] which differ and do in fact come from different numbers of iterations. We then compare the histogram to the results computed from the Bayes factor using the RJMCMC chain by selecting samples belonging to different models, i.e. using the same set of samples. The differences between the $p(k|Y)$ estimates result from the approximation of the marginal posterior densities $p(\theta_k|Y,k)$ in each model. When we apply the Gaussian kernel smoother to the histogram of θ_k, bias is introduced in the $p(\theta_k|Y,k)$ approximation, and therefore there is bias in $p(\theta_k^*|Y,k)$ for a chosen θ_k^*. This in turn gives a biased estimation of $p(Y|k)$ using Equation 17 and finally the biased $p(k|Y)$ obtained from Equation 15. We also compare the results of $p(k|Y)$ using samples from the RJMCMC and multiple MCMC chains. Since they are analyzed in the same way, the results illustrate the differences of within-model samples from within-model and trans-model chains. The point is that in RJMCMC chains, because of the between model jumps, samples belonging to one particular subspace could be treated as the connection of separate sub-chains. In contrast, MCMC chains provide the continuously serial samples in each model.

5. CONCLUSIONS AND FUTURE WORK

We have implemented a method for parallel MCMC chains of different fixed dimension, and compared the results to the prototypical serial RJMCMC chain which allows dimension-changing moves. Comparing the convergence diagnostics of trans-model and within-model chains, the parallel, multiple within-model chains method improves simulation efficiency because the MCMC chains converge faster than the RJMCMC chains. In a parallel MCMC implementation, proposals are made, accepted or rejected within a fixed dimension, but in serial RJMCMC, the sample space for a given k is explored using steps between different dimensions, which means that the exploration path must be quite different. This warrants further investigation. There is also a difference in the dimensional prediction of the serial RJMCMC and parallel MCMC methods, whether the number of iterations in each dimension or the Bayes factor is used respectively, and this two must be addressed in determining the correct distribution for k. This particular parallel strategy is simple and has obvious benefits, as no communication is necessary during chain generation. In this context, there are many further improvements that could be made, notably in load balancing between chains for differing k since the number of parameter estimations are different in each chain, or in parallelising the individual chains using other methods.

REFERENCES

[1] A.E.Brockwell, "Parallel Markov chain Monte Carlo simulation by pre-fetching," *Journal of Computational & Graphical Statistics*, vol. 15, pp. 246–261, 2006.

[2] A.Gelman and D.B.Rubin, "Inference from iterative simulation using multiple sequences," *Statistical Science*, vol. 7, no. 4, pp. 457–472, 1992.

[3] A.Gelman, *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*, chapter 8, Chapman & Hall/CRC, 1995.

[4] P.J.Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711–732, 1995.

[5] S.Brooks and P.Giudici, "Convergence assessment for reversible jump MCMC simulations," *Bayesian Statistics*, vol. 6, pp. 733–742, 1999.

[6] P.J.Green, *Trans-dimensional Markov chain Monte Carlo*, chapter 6, Oxford University Press, 2003.

[7] S.Chib, "Marginal likelihood from the Gibbs output," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1313–1321, December 1995.

[8] S.Chib and I.Jeliazkov, "Marginal likelihood from the Metropolis-Hastings output," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 270–281, March 2001.

[9] S.Hernandez-Marin, A.M.Wallace and G.J.Gibson, "Bayesian analyis of LiDAR signals with multiple returns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2170–2180, 2007.

[10] E.Lusk W.Gropp and T.Sterling, *Beowulf Cluster Computing with Linux*, The MIT Press, second edition, December 2003.