Version 1.1

Programming with Big Data in R

Speaking Serial R with a Parallel Accent

Package Examples and Demonstrations

pbdR Core Team

Speaking Serial R with a Parallel Accent

pbdR Package Examples and Demonstrations

Drew Schmidt
Remote Data Analysis and Visualization Center,
University of Tennessee, Knoxville

Wei-Chen Chen
Computer Science and Mathematics Division,
Oak Ridge National Laboratory

Pragneskumar Patel
Remote Data Analysis and Visualization Center,
University of Tennessee, Knoxville

George Ostrouchov
Computer Science and Mathematics Division,
Oak Ridge National Laboratory

Version 1.1

August 7, 2013

This publication was typeset using LaTeX. Illustrations were created using the **ggplot2** package (Wickham, 2009), native R functions, and Microsoft Powerpoint.

Contents

List of Figures

List of Tables

Acknowledgements

Schmidt, Ostrouchov, and Patel were supported in part by the project "NICS Remote Data Analysis and Visualization Center" funded by the Office of Cyberinfrastructure of the U.S. National Science Foundation under Award No. ARRA-NSF-OCI-0906324 for NICS-RDAV center. Chen and Ostrouchov were supported in part by the project "Visual Data Exploration and Analysis of Ultra-large Climate Data" funded by U.S. DOE Office of Science under Contract No. DE-AC05-00OR22725.

This work used resources of National Institute for Computational Sciences at the University of Tennessee, Knoxville, which is supported by the Office of Cyberinfrastructure of the U.S. National Science Foundation under Award No. ARRA-NSF-OCI-0906324 for NICS-RDAV center. This work also used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This work used resources of the Newton HPC Program at the University of Tennessee, Knoxville.

We also thank Brian D. Ripley, Kurt Hornik, Uwe Ligges, and Simon Urbanek from the R Core Team for discussing package release issues and helping us solve portability problems on different platforms.

Disclaimer

Warning: The findings and conclusions in this article have not been formally disseminated by the U.S. Department of Energy and should not be construed to represent any determination or policy of University, Agency and National Laboratory.

This document is written to explain the main functions of **pbdDEMO** (Schmidt *et al.*, 2013), version 0.1-1. Every effort will be made to ensure future versions are consistent with these instructions, but features in later versions may not be explained in this document.

Information about the functionality of this package, and any changes in future versions can be found on website: "Programming with Big Data in R" at `http://r-pbd.org/`.

1 Bayesian MCMC

No, I am not tired. I have a curious constitution. I never remember feeling tired by work, though idleness exhausts me completely.

—**Sherlock Holmes**

1.1 Introduction

In modern statistics, likelihood principle introduced in Chapter **??** has produced several advantages to data analysis and statistical modeling. However, as model getting larger and data size getting bigger, the maximization of likelihood function becomes infeasible analytically and numerically. Bayesian statistics based on Bayes theorem somehow relieves the burden of optimization, but it changes the way of statistical inference.

In likelihood principle, we based on maximum likelihood estimators for estimations, hypothesis testings, confidence intervals, etc. In Bayesian framework, we make inference based on posterior distribution, which is a composition of likelihood and prior information, such as for posterior means and creditable intervals. For more information about Bayesian statistics, readers are encouraged to read Berger (1993); Gelman *et al.* (2003).

Mathematically, we denote $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ for posterior, $p(\boldsymbol{x}|\boldsymbol{\theta})$ for likelihood, and $\pi(\boldsymbol{\theta})$ for prior where $\boldsymbol{x}$ is a collection of data and $\boldsymbol{\theta}$ is a set of interesting parameters. The idea of Bayes theorem says

$$\begin{aligned} \pi(\boldsymbol{\theta}|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int p(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}} && (1.1) \\ &\propto p(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) && (1.2) \end{aligned}$$

in short, the posterior is proportional to the product of likelihood and prior. Note that the integral denominator of Equation (1.1) can be seen as a normalizing constant, and is usually ignorable in most of Bayesian methods, then Equation (1.2) provides great reduction tricks for analytical and simulated solutions.

For example, suppose $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$ are random samples from $N(\mu, \sigma^2)$ where μ is un-

known and needed to be inferred (i.e. $\boldsymbol{\theta} = \{\mu\}$), and σ^2 is known. Suppose further μ has a prior distribution $N(\mu_0, \sigma_0^2)$ where μ_0 and σ_0^2 are known. After a few calculation, we have the posterior for $\mu|\boldsymbol{x}$ in next denoted in a conventional syntax.

$$\boldsymbol{x} \overset{i.i.d.}{\sim} N(\mu, \sigma^2) \tag{1.3}$$

$$\mu \sim N(\mu_0, \sigma_0^2) \tag{1.4}$$

$$\mu|\boldsymbol{x} \sim N(\mu_n, \sigma_n^2) \tag{1.5}$$

where $\mu_n = \sigma_n^2 \left(\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{x}}{\sigma^2}\right)$, $\sigma_n^2 = \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right)^{-1}$, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. This means the posterior mean of location parameter μ is estimated by weighted of sample mean $\bar{x}$ and prior mean μ_0 via their precisions σ^2 and σ_0^2. A nice interpretation of the posterior mean is that it combines information of data (sample mean) and knowledge (prior) together into the model, Equation (1.5). Further, a new prediction of x given this model is also a normal distribution that

$$\hat{x} \sim N(\mu_n, \sigma_n^2 + \sigma^2). \tag{1.6}$$

In this example, the prior and the posterior are both normal distributions that we call this kind of prior as a conjugate prior. In general, a conjugate prior may not exist and may not have a good interpretation to the application. The advantage is that the analytical solution is feasible for conjugate cases. However, a prior may be better to based on known information such as previous experiments or domain knowledge. For instance, empirical Bayes relies on empirical data information, or non-informative priors provide wider domain of parameters. Nevertheless, Markov Chain Monte Carlo (MCMC) is a typical solution when an analytical solution is tedious.

1.2 Hastings-Metropolis Algorithm

In reality, a proposed distribution may not be easy to obtain samples or to generate from, while Acceptant-Rejection Sampling algorithm is a fundamental method in Computational Statistics to deal with this situation by generating data from a relative easier distribution and based on the acceptant-rejection probability to keep or drop the samples. See Ross (1996) for more details about Acceptant-Rejection Sampling algorithm.

Hastings-Metropolis algorithm (Hastings, 1970; Metropolis *et al.*, 1953) is one of Markov Chain Monte Carlo method to obtain a sequence of random samples where a proposed distribution is difficult to sample from. The idea is to utilize Acceptant-Rejection Sampling algorithm to sample sequentially from conditional distributions provided relative easier than the proposed distribution, and via acceptance rejection probability to screen appropriate data from an equilibrium distribution. The computation of π in Section **??** is an example of Acceptant-Rejection Sampling algorithm for Monte Carlo case but without Markov Chain.

Suppose a stationary distribution exists for θ in the domain of investigation Θ. Provided the Markov Chain is adequate (periodic, irreducible, time reversible, ...), we may have

$$\pi(\theta^{(t)})p(\theta|\theta^{(t)}) = \pi(\theta)p(\theta^{(t)}|\theta) \tag{1.7}$$

where $p(\theta|\theta^{(t)})$ is a transition probability at the t-th step from the current state $\theta^{(t)}$ to a new state θ for all $\theta^{(t)}, \theta \in \Theta$. Since $p(\theta|\theta^{(t)})$ may not be easy to sample, Hastings-Metropolis algorithm

suggests a proposal distribution $q(\theta|\theta^{(t)})$ with an acceptant probability $a(\theta|\theta^{(t)})$ such that

$$a(\theta|\theta^{(t)}) = \frac{p(\theta|\theta^{(t)})}{q(\theta|\theta^{(t)})}. \tag{1.8}$$

Equation (1.7) becomes

$$\frac{a(\theta|\theta^{(t)})}{a(\theta^{(t)}|\theta)} = \frac{\pi(\theta)q(\theta^{(t)}|\theta)}{\pi(\theta^{(t)})q(\theta|\theta^{(t)})}. \tag{1.9}$$

The acceptant probability will be

$$a(\theta|\theta^{(t)}) = \min\left\{1, \frac{\pi(\theta)q(\theta^{(t)}|\theta)}{\pi(\theta^{(t)})q(\theta|\theta^{(t)})}\right\} \tag{1.10}$$

that $\theta^{(t+1)} = \theta$ if accepted, otherwise $\theta^{(t+1)} = \theta^{(t)}$ (new θ is rejected).

The steps of Hastings-Metropolis Algorithm are summarized next:

Step 1: Initial a $\theta^{(0)}$ from $\pi(\theta)$. Set $i = 1$.

Step 2: Generate a new θ' from $g(\theta|\theta^{(0)})$.

Step 3: Compute $a(\theta'|\theta^{(i)})$.

Step 4: Genera a uniform random variable U.

Step 5: If $U \leq a(\theta'|\theta^{(i)})$, then set $\theta^{(i+1)} = \theta'$. Otherwise, set $\theta^{(i+1)} = \theta^{(i)}$.

Step 6: Set $i = i + 1$ and repeat Steps 2 to 5.

Typically, we repeat Steps 2 to 5 until the process is burn-in, says $I_b = 1,000$ iterations, after that we continuously collect $\{\theta^{(i)}\}$ for thinning every $I_t = 10$ iterations to release time dependent problems. Repeat the thinning process until I_n samples are reached. We also repeat $I_c = 5$ Markov Chains with different initial values to verify the stationary. The determinations of I_b, I_t, I_n, and I_c are dependent on models, data, and prior, see Spiegelhalter *et al.* (2003) for more information.

Although Hastings-Metropolis algorithm can solve complex problem, larger number of I_b, I_t, I_n, and I_c result in time consuming computations and large storage space. An easy way to rescue this burden is to parallelization the algorithm. At least three possible parallelizations for N processors can be considered in following.

1. Each Markov Chain is executed on each processor. Only I_n/N samples are needed to be collected for each processor provided every Markov Chain is burn-in.

2. Execute one Markov Chain on one processor. Until the Markov Chain is burn-in, then the burn-in state is broad casted to all processors. Set different random seeds on all processors, then all processors proceed the Markov Chain until I_n/N samples are collected for each processor.

3. For large size problem, distributing data is unavoidable, then N processors execute one common Markov Chain to collect I_n samples.

Note that the second one is only useful for short burn-in chains. We next use a galaxy velocity example to demonstrate the first parallelization above, and make statistical inference based on the Bayesian framework.

1.3 Galaxy Velocity

Velocities of 82 galaxies in the region of Corona Borealis are measured and reported in (Roeder, 1990), and the `galaxies` dataset is available in **MASS** package of R. The mean is about $20,828.17$ km/sec and the standard deviation is about $4,563.758$ km/sec. Figure 1.1 shows the distribution of data.

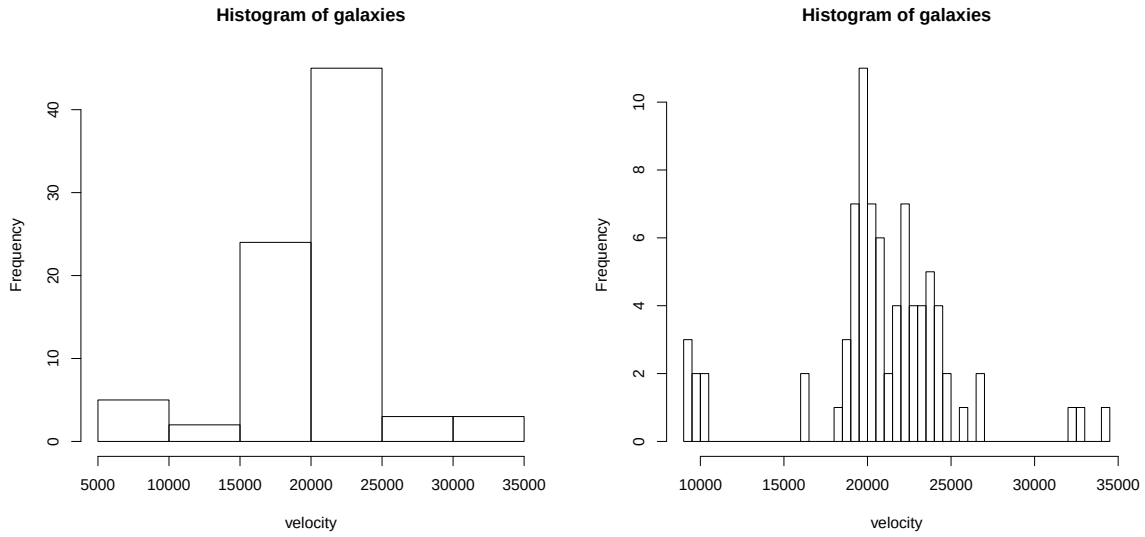

Figure 1.1: Histograms of velocities of 82 galaxies. The left plot is based on default setting of `hist(galaxies)` and the right plot is based on `hist(galaxies, nclass=50)` providing more details of distribution.

Suppose we are interesting in the mean velocity of those galaxies and want to model them as Equations (1.3), (1.4), and (1.5). An example code is given in the **pbdDEMO** demo via

```
### At the shell prompt, run the demo with 4 processors by
### (Use Rscript.exe for windows system)
mpiexec -np 4 Rscript -e "demo(galaxy_mcmc,'pbdDEMO',ask=F,echo=F)"
```

which provides Figure **??**.

1.4 Exercises

1-1 Prove Equation (1.5) and claim it is conjugate. Hint: Equation (1.2).

1-2 Prove Equation (1.6) and explain intuitively why the variance of predictive sample is increased comparing with that of observed samples. Hint: is a 95% predictive interval wider than a 95% confidence interval.

1-3 Claim that Equation (1.10) is the solution of Equation (1.9). Hint: when is $a(\theta^{(t)}|\theta) = 1$?

1-4 Prove the proposal distribution q with Equation (1.10) provides the desired distribution p. Hint: Acceptance-Rejection Sampling algorithm.

1-5 Claim that the upper bound of Equation (1.8) controls the performance of Hastings-Metropolis algorithm. Hint: what if $p(\theta|\theta^{(t)}) = q(\theta|\theta^{(t)})$?

1-6 Discuss when Hastings-Metropolis algorithm fails. Provide an example that is an inefficient case of Hastings-Metropolis algorithm. Hint: What are requirements of Markov Chain?

1-7 Extend the model and algorithm of galaxy velocities example for unknown mean and unknown variance. e.g.

$$\begin{aligned} \boldsymbol{x} &\overset{i.i.d.}{\sim} N(\mu, \sigma^2) \\ \mu &\sim N(\mu_0, \sigma_0^2) \\ \sigma &\sim Gamma(\alpha_0, \beta_0) \end{aligned}$$

Find the 95% creditable region for $(\mu|\boldsymbol{x}, \sigma|\boldsymbol{x})$.

1-8 Section 1.3 only considers homogeneous distribution for all galaxy velocities. As model-based clustering in Section **??**, please extend to a two clusters problem and implement it in Bayesian framework.

Part I

Miscellany

References

Berger J (1993). *Statistical Decision theory and Bayesian Analysis*. 2nd edition. Springer.

Gelman A, Carlin J, Stern H, Rubin D (2003). *Bayesian Data Analysis*. 2nd edition. Chapman & Hall/CRC.

Hastings W (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika*, **57**, 97–109.

Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953). "Equations of State Calculations by Fast Computing Machines." *Journal of Chemical Physics*.

Roeder K (1990). "Density estimation with confidence sets exemplified by superclusters and voids in the galaxies." *Journal of the American Statistical Association*, **85**, 617–624.

Ross S (1996). *Simulation*. 2nd edition. Oxford.

Schmidt D, Chen WC, Ostrouchov G, Patel P (2013). "pbdDEMO: Programming with Big Data – Demonstrations of pbd Packages." R Package, URL http://cran.r-project.org/package=pbdDEMO.

Spiegelhalter D, Thomas A, Best N, Lunn D (2003). *WinBUGS User Manual*. URL http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/manual14.pdf.

Wickham H (2009). *ggplot2: elegant graphics for data analysis*. Springer New York. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book.

Index