News

The Newsletter of the R Project — Volume 6/1, March 2006

Editorial

by Martyn Plummer and Paul Murrell

Welcome to the first issue of R News for 2006. This is my first issue as Editor-in-Chief and, in the spirit of "start as you mean to go on", I have had someone else do all the work. Martyn Plummer is guest editor for this special issue, which has a distinctly Bayesian flavour. Thanks to Martyn and his merry band of contributors and reviewers for getting this special issue together.

Normal service and regular features will resume with a regular issue following the release of R 2.3.0. For now, sit back, relax, and prepare to gorge yourself on a feast of Bayes.

Paul Murrell
The University of Auckland, New Zealand
paul.murrell@R-project.org

This special issue of R News is dedicated to Bayesian inference and Markov Chain Monte Carlo (MCMC) simulation. The choice of articles for this issue is subjective. We aim to give you a snapshot of some current work on Bayesian statistical computing in R without any claim to comprehensiveness. A broader view is provided by the CRAN task view on Bayesian inference maintained by Jong Hee Park. This currently lists 33 packages, some of which have been discussed in previous R News articles (Yan, 2004; Raftery et al., 2005).

MCMC has become the numerical method of choice for many Bayesians. Although computationally expensive, it enables highly complex probability models to be analyzed. This capability, combined with cheap, abundant computing power, has contributed to the increasing popularity of Bayesian methods in applied statistics. The field has now matured to the point where most users should not expect to write custom software, but may use one of several existing "sampling engines". R provides a natural front end for these engines. This is nicely illustrated by the package **MCMCpack**, subject of the first article by Andrew Martin and Kevin Quinn. The computational back-end of **MCMCpack** is provided by the Scythe C++ statistical library. The front end is a collection of R functions for fitting models commonly used in the social and behavioural sciences. This is followed by an article on **coda**, which provides the infrastructure for analyzing MCMC output, and is used by **MCMCpack** among other packages.

Samantha Cook and Andrew Gelman then discuss validation of MCMC software using the **BayesValidate** package. Since the results of MCMC simulation are never exact, and the models used are typically very complex, validation of the sampling software can be quite difficult. However, the lack of validation of Bayesian software may be holding back its acceptance in regulated environments such as clinical trials approved by the US Food and Drug Administration, so this is an important problem.

We continue with a pair of articles on OpenBUGS. The BUGS (Bayesian inference Using Gibbs

Contents of this issue:

Editorial 1
Applied Bayesian Inference in R using MCMCpack 2
CODA: Convergence Diagnosis and Output Analysis for MCMC 7
Bayesian Software Validation 11
Making BUGS Open 12
The BUGS Language 17
Bayesian Data Analysis using R 21
Erratum 24

Sampling) project is a long-running project to provide a user-friendly language and environment for Bayesian inference. The first article, by Andrew Thomas and colleagues, describes the **BRugs** package which provides an R interface to the OpenBUGS engine. The second article by Andrew Thomas describes the BUGS language itself and the design philosophy behind it. Somewhat unusually for an article in R News, this article does not describe any R software, but it is included to highlight some of the differences in the way statistical models are represented in R and OpenBUGS.

The issue ends with an article by Jouni Kerman and Andrew Gelman, who give a personal perspective on what the next generation of Bayesian software may look like, and preview some of their own work in this area, notably the **rv** package for representing simulation-based random variables, and the forthcoming "Universal Markov Chain Sampler" package, **Umacs**.

Bibliography

A. E. Raftery, I. S. Painter, and C. T. Volinsky. BMA: An R package for bayesian model averaging. *R News*, 5(2):2–8, November 2005. URL http://CRAN.R-project.org/doc/Rnews/. 1

J. Yan. Fusing R and BUGS through Wine. *R News*, 4(2):19–21, September 2004. URL http://CRAN.R-project.org/doc/Rnews/. 1

Martyn Plummer
International Agency for Research on Cancer
Lyon, France
plummer@iarc.fr

Applied Bayesian Inference in R using MCMCpack

by Andrew D. Martin and Kevin M. Quinn

Introduction

Over the past 15 years or so, data analysts have become increasingly aware of the possibilities afforded by Markov chain Monte Carlo (MCMC) methods. This is particularly the case for researchers interested in performing Bayesian inference. Here, MCMC methods provide a fairly straightforward way for one to take a random sample approximately from a posterior distribution. Such samples can be used to summarize any aspect of the posterior distribution of a statistical model. While MCMC methods are extremely powerful and have a wide range of applicability, they are not as widely used as one might guess. At least part of the reason for this is the gap between the type of software that many applied users would like to have for fitting models via MCMC and the software that is currently available. **MCMCpack** (Martin and Quinn, 2005) is an R package designed to help bridge this gap.

Until the release of **MCMCpack**, the two main options for researchers who wished to fit a model via MCMC were to: a) write their own code in R, C, FORTRAN, etc., or b) write their own code (possibly relying heavily on the available example programs) using the BUGS language[1] in one of its various implementations (Spiegelhalter et al., 2004; Thomas, 2004; Plummer, 2005). While both of these options offer a great deal of flexibility, they also require non-trivial programming skills in the case of a) or the willingness to learn a new language and to develop some modest programming skills in the case of b). These costs are greater than many applied data analysts are willing to bear. **MCMCpack** is geared primarily towards these users.

The design philosophy of **MCMCpack** is quite different from that of the BUGS language. The most important design goal has been the implementation of MCMC algorithms that are model-specific. This comports with the manner in which people oftentimes think about finding software to fit a particular class of models rather than thinking about writing code from the ground up. The major advantage of such an approach is that the sampling algorithms, being hand-crafted to particular classes of models, can be made dramatically more efficient than black box approaches such as those found in the BUGS language, while remaining robust to poorly conditioned or unusual data. All the **MCMCpack** estimation routines are coded in C++ using the Scythe Statistical Library (Martin et al., 2005). We also think it is easier to call a single R function to fit a model than to code a model in the BUGS language. It should also be noted that **MCMCpack** is aimed primarily at so-

[1] The BUGS language is a general purpose language for simulation from posterior distributions of statistical models. BUGS exploits conditional independence relations implied by a particular graphical model in order to automatically determine an MCMC algorithm to do the required simulation. In order to fit a model, the user must specify a graphical model using either the BUGS language or (in the case of WinBUGS) a graphical user interface.