Bayesian communication software

The software provided at this site enables you to accomplish many of the goals of Bayesian communication in a user-friendly way. The software uses the output of any posterior simulator to accomplish many tasks:

- Calculate posterior means and posterior standard deviations, and assess the numerical accuracy of the posterior mean (`/moment`)
- Evaluate the marginal likelihood of a model (`/mlike`)
- Combine posterior moments from two or more posterior simulations, and compute conventional test statistics for equality of these moments (`/apm`)
- Change the prior distribution by reweighting the posterior simulator output, and/or evaluate alternative posterior moments (`/reweight`)
- Prepare input for standard plotting software, to approximate posterior probability density functions (`/graph`)
- Calculate upper and lower bounds on posterior moments, as the prior distribution is varied from its original specification (`/robust`)
- Prepare a summary of a posterior simulator file that is useful in designating the transient phase of a Markov chain Monte Carlo posterior simulator (`/converge`)

To use any of the seven routines, it is necessary that posterior simulator output be organized in a standard way. The `reweight` software generates new posterior simulator files in this same, standard way.

Posterior simulator file structure

All the software is organized around the creation and subsequent use of posterior simulator files. A posterior simulator for a particular model initially writes a posterior simulator file. For each iteration this file records, at a minimum, the full parameter vector. The initial record of the file indicates the number of iterations and the number of entries recorded for each iteration.

For each iteration, *m*, the posterior simulator file has two records. The first record has four entries. The first is *m*. The second entry is the logarithm of the weighting function, $\log \mathrm{w}\left(\theta^{(m)}\right)$, the log ratio of posterior density kernel to importance sampling density (if any). For most Markov chain Monte Carlo (MCMC) methods, this value is zero. The third entry is the logarithm of the prior density, $\log \mathrm{p}\left(\theta^{(m)}|A\right)$, and the last entry is the logarithm of the data density, $\log \mathrm{p}\left(\mathbf{Y}_T\middle|\theta^{(m)}, A\right)$. The second record is the *iteration vector*, consisting of $\theta^{(m)}$ and (perhaps) functions of $\theta^{(m)}$. It is written five entries per line and in

general occupies multiple lines. The organization of this vector is specific to the particular application, and it is necessary to know how the vector has been set up in order to make sense of the posterior simulator file.

Structure of the next level of subdirectories

The software is available in six languages: Fortran 77; c; Gauss; Mathematica; MatLab; and S-plus. There is a subdirectory for each kind of source code within each software subdirectory.