

Practical Considerations for WinBUGS Users

Kate Cowles, Ph.D.

Department of Statistics and Actuarial Science

University of Iowa

22S:166

Lecture 15

Oct. 16, 2006

Using MCMC for Bayesian inference (idealized sequence!)

- specify a Bayesian model
- construct a Markov chain whose target distribution is the joint posterior distribution of interest
- run the chain until output converges in distribution to draws from the target distribution
- base inference regarding unknown quantities in model on output of subsequent iterations of the chain

Issues in MCMC use for Bayesian model fitting

- Deciding how many chains to run
- Choosing initial values
- Assessing whether sampler has “converged”
- Choosing model parameterizations and MCMC algorithms that will lead to convergence in a reasonable amount of time
- Using correlated samples for estimation and inference
 - adjusting estimates of standard errors

Convergence assessment

- How many initial iterations need to be discarded in order that remaining samples are drawn from a distribution close enough to the true stationary distribution to be usable for estimation and inference?
 - “burn-in”
- Has the sampler traversed the entire support of the posterior distribution?
 - “support”
- How many (dependent) samples are needed in order to provide the desired precision in estimation and inference?
 - “variance”

Using MCMC for Bayesian inference (more realistic sequence)

1. specify a Bayesian model
2. construct a Markov chain whose target distribution is the joint posterior distribution of interest
3. run one or more chain(s) as long as you can stand it
4. assess convergence
 - retune model parameterization, Markov chain algorithm, priors, etc.
 - or increase number of iterations
 - return to step 3 or continue to step 5 as appropriate
5. base inference regarding unknown quantities in model on portion of output identified during convergence assessment

– Monte Carlo errors

- decide whether to
 - discard some initial burn-in iterations and use remaining sampler output for inference
 - or take some corrective action
 - * run more iterations
 - * change parameterization of model

Suggested steps in WinBUGS

- run a minimum of 3 parallel chains from overdispersed initial values
- monitor all unknown model quantities beginning from the first iteration
 - if not feasible to monitor all, then monitor at least representative samples of each *kind* of parameter
 - let WinBUGS help you determine how many early iterations to throw out for burn-in
- inspect the following WinBUGS output to evaluate sampler performance
 - history plots
 - autocorrelation plots
 - Brooks, Gelman, and Rubin diagnostic

Initial values

- Initial values are not like priors!
 - Priors are part of the model specification.
 - Priors must *not* be derived from the current dataset.
 - Initial values are part of the computing process.
 - Initial values can be derived from the current dataset.
- Choosing initial values
 - Run more than one chain with initial values selected to give you information about sampler performance.
 - Initial values may be generated from priors.
 - Initial values may be based on frequentist estimates.
 - * e.g. mle, mle - 4 standard errors, mle + 4 standard errors

- Initial values may be chosen systematically to represent extreme regions of the parameter space.
- Initial values must be specified for variance components
 - WinBUGS usually can automatically generate initial values for other parameters
 - But it's often advantageous to specify even those WinBUGS can generate

Convergence diagnostics

- Statistical methods applied to the *output* of MCMC samplers in an effort to assess convergence.
- Gelman and Rubin (1992): Early diagnostic comparing variance between/within multiple chains
 - requires running multiple parallel chains from overdispersed initial values
 - applied to each univariate quantity of interest
 - always applied to last half of sampler output (i.e., assumes that first half of sampler output is burn-in)
 - computes “potential scale reduction factor”: the factor by which the scale parameter of the estimated marginal distribution might shrink if sampling were continued indefinitely

History plots: Early graphical methods of convergence assessment

- trajectories of sampler output for each model unknown
- can quickly reveal failure to reach stationarity
- can give qualitative information about sampler behavior
- cannot confirm that any of the three aspects of convergence have occurred

- Brooks and Gelman (1998)
 - corrected computation of Gelman and Rubin (1992) potential scale reduction factor
 - propose a multivariate potential scale reduction factor to simultaneous assessment of convergence of all model unknowns
 - extended method to other measures besides estimate of posterior variance, particularly widths of credible sets

Gelman-Rubin convergence diagnostic in WinBUGS

- obtain in graphical form using “GRdiag” button on Inference menu
- plots the following versus iteration
 - width of central 80% interval constructed from pooled runs (plotted in green)
 - average width of 80% intervals constructed from each run (plotted in blue)
 - ratio $R = \frac{\text{pooled}}{\text{within}}$ in red
- both widths are scaled so maximum values are 1
- convert plot to numeric output by
 - double-clicking on plot

Monte Carlo error

- on “Stats” output
- similar to standard error of the mean, but adjusted for autocorrelated sample
- it will get smaller as more iterations are run
- use it to guide your decision as to how many iterations you need to run after burn-in is done

– then control-left-mouse-click on the window)

- want to see:
 - BGR R (value of numeric diagnostic) close to 1
 - convergence of both pooled- and within-interval widths to stability

Software for MCMC convergence assessment

- Rudimentary facilities built into WinBUGS
- BOA (Bayesian Output Analysis)
 - developed in 1999 by Brian Smith of University of Iowa
 - includes all the convergence diagnostics/output analysis features in WinBUGS and much more
 - available for free download from
<http://www.public-health.uiowa.edu/BOA/>
 - use “Coda” button on WinBUGS Inference menu to export WinBUGS output to a file that BOA can process
- downside: by definition canned software cannot perform problem-specific convergence assessment procedures

Recommendations

- Learn as much as possible about model before ever running an MCMC sampler.
 - maximum likelihood
 - noniterative Bayesian approximations
 - numerical mode-finding
 - simplified models
- Pre-assess burn-in time if possible; then run a single chain.
 - how to do this is beyond scope of this course
- If this is not possible, run a small number (3 – 5?) of parallel chains from overdispersed starting values.

Conclusions

- MCMC methods have enabled the fitting of complex, realistic models.
- Use of MCMC methods requires careful attention to
 - model parameterization
 - MCMC sampler algorithms
 - choice of initial values
 - convergence assessment
 - output analysis
- Ongoing research in theoretical verification of convergence, MCMC acceleration, and exact sampling holds great promise.

- Monitor
 - all types of model parameters, not only parameters of substantive interest
 - sample paths graphically
 - autocorrelations
 - cross-correlations between parameters
- Apply more than one diagnostic, including one or more that uses information about the specific model.