Lecture 14: A Survey of Automatic Bayesian Software and Why You Should Care

Zhenke Wu
BIOSTAT 830 Probabilistic Graphical Models
October 25th, 2016
Department of Biostatistics, University of Michigan
Bayes Formula

Model likelihood for observed data \( x \)

\[
P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}
\]

- Marginal distribution of data given the model;
- “Evidence” that this data \( x \) are generated by this model (Box 1980, JRSS-A)
- Exact computation possible (junction-tree algorithms), but hard for complex likelihood and priors (e.g., a graphical model with large tree-width, Dirichlet process prior etc.)

Thomas Bayes (1701-1761)

*figure from Wikipedia; some say this is not Bayes
Gibbs Sampling

Use simulated samples to approximate the *entire* joint posterior distribution
Why Automatic Software for Bayesian Inference?

• Self-coded simulation algorithms usually require extra tuning and cost much time (**share your experience**)

• General formula/recipes exist for sampling from common distributions (adaptive rejection sampling, slice sampling, Metroplis-Hastings algorithm)

• Modelers generally want **reasonable** and **fast** model outputs to speed up model building, testing and interpretation
Analytic Pipeline
Bayesian Software and Google Trends

• WinBUGS/OpenBUGS
• JAGS
• Stan
• PyMC3
• Others, e.g, R-INLA, NIMBLE, MCMCpack...

https://goo.gl/YNQbCP
If Adding the Trend for $R$?

https://goo.gl/orflY
The R Inferno

By Patrick Burns

Abstract: If you are using R and you think you’re in hell, this is a map for you.
WinBUGS  

http://www.mrc-bsu.cam.ac.uk/software/bugs/

- Bayesian inference Using Gibbs Sampling
- Latest Version: 1.4.3; Add-on modules, e.g., GeoBUGS
- Call from R by “R2WinBUGS”
- Since 1989 in Medical Research Council (MRC) Biostatistics Unit, Cambridge --- David Spiegelhalter with chief programmer Andrew Thomas; Motivated by Artificial Intelligence research
- 1996 to Imperial College, London --- Nicky Best, Jon Wakefield and Dave Lunn
- No change since 2007
- In 2004 OpenBUGS is branched from WinBUGS by Andrew Thomas (http://www.openbugs.net/w/FrontPage); still under development
Good Experience - WinBUGS

• GUI, easy for visual inspection of chains without too much posterior sample processing

• Good teaching tool with a companion book: *The BUGS Book - A Practical Introduction to Bayesian Analysis*

• Coded in many common distributions suitable for different types of data (see *Manual*)

• Relative easy for debugging because it points to specific errors
Bad Experiences - WinBUGS

• “Why you should not use WinBUGS or OpenBUGS” - Barry Rowlingson
  http://geospaced.blogspot.com/2013/04/why-you-should-not-use-winbugs-or.html

• Odd errors, e.g., “trap” messages for memory errors

• Written in Component Pascal; can only be read with BlackBox Component
  Builder from Oberon Microsystems, which only runs on Windows. Also
  BlackBox was abandoned by its own developers in 2012.

• Not very open-source, although with tools to extend WinBUGS

• Essentially sample nodes **univariately**; block sampling only available for
  multivariate nodes, or fixed-effect parameters in GLMs by Metropolis-
  Hastings algorithm proposed by Iteratively Reweighted Least Squares.
Example: Penalized-Spline Regression

WinBUGS (500 data points; 10,000 iterations; 5.87 secs)

```
for (i in 1:N){
  M[i] ~ dnorm(mu[i],prec)
  #mu[i] <- inprod2(ZB[i,],beta[])
}

sigma <- pow(prec,-0.5)
# prior for B-spline coefficients: first-order penalty matrix:
beta[1] ~ dnorm(0,prec_beta1)
for (c in 2:C){
  beta[c] ~ dnorm(beta[c-1],taubeta)
}
taubeta ~ dgamma(3,2)
prec_beta1 <- 1/4*prec
prec ~ dgamma(1.0E-2,1.0E-2)
```
Example: Penalized-Spline Regression
WinBUGS (10,000 iterations; 5.87 secs)

Data points

B-spline basis multiplied by estimated coefficients

True mean curve

Posterior samples of mean curves
JAGS (Just Another Gibbs Sampler)

- Latest version 4.0.0; Author: Martyn Plummer; first release: 2007
- “not wholly unlike BUGS” with three aims:
  - cross-platform engine (written in C++), e.g., Mac OS X, Linux, Windows
  - extensibility
  - a platform for experimentation
- **Experience**:
  - great speed (load the “glm” module!); built-in vectorization
  - responsive online community (mostly responded in a day by Martyn himself)
  - generic error messages hard to know exactly what went wrong
  - no GUI
Example: Penalized-Spline Regression
JAGS (10,000 iterations; 4.15 secs)

```r
model{
  for (i in 1:N){
    M[i] ~ dnorm(mu[i], prec)
  }
  sigma <- pow(prec,-0.5)
  mu <- ZB%*%beta # vectorized.
  # prior for B-spline coefficients: first-order penalty matrix:
  beta[1] ~ dnorm(0, prec_beta1)
  for (c in 2:C){
    beta[c] ~ dnorm(beta[c-1], taubeta)
  }
  taubeta ~ dgamma(3,2)
  prec_beta1 <- 1/4*prec
  prec ~ dgamma(1.0E-2,1.0E-2)
}
```
Stan

http://mc-stan.org/interfaces/

- named in honor of Stanislaw Ulam, pioneer of the Monte Carlo method (Metropolis, Nicholas, and Stanislaw Ulam (1949). The Monte Carlo method. JASA)
- Inferential Engine:
  - MCMC sampling (No U-Turn Sampler; Hamiltonian Monte Carlo)
  - Approximate Bayesian inference (variational inference)
  - Penalized maximum likelihood estimation (Optimization)
- Latest version 2.12.0; Developed by at Columbia; initial release August 2012
- Cross-platform; Written in C++; Open-source
- Call from R by “rstan”; can also be called from Python by “PyStan”; Julia...
- Very sweet part: “shinyStan” package; see demo.
Example: Penalized-Spline Regression
Stan(10,000 iterations; 9.44 secs)

data {
  int<lower=0> N; // number of observations
  int<lower=0> C; // number of B-spline bases
  matrix[N,C] ZB; // predictor for observation n
  vector[N] M;  // outcome for observation n
}
parameters {
  real<lower=0> sigma; // noise variance
  real<lower=0> sigma_beta;  // smoothing parameter.
  vector[C] beta; // regression
}
transformed parameters{
  vector[N] mu;
  mu <- ZB * beta;
}
model {
  sigma ~ cauchy(0,5);
  sigma_beta ~ cauchy(0,5);
  beta[1] ~ normal(0,2*sigma);
  for (l in 2:C) beta[l] ~ normal(beta[l-1],sigma_beta);
  M ~ normal(mu, sigma);
}
shinyStan
RStan Experience

- Vectorized functions --- fast! (built upon Eigen, a C++ template library for linear algebra)
  - Good when the data are big but the model is small
- C type variable declaration; provides extensive warning/error messages
- Not reliant upon conjugate priors (compare to BUGS)
- Convenient to install by `install.packages("rstan")`
- Hosted by GitHub

- Currently cannot sample discrete unknown parameters
- Not always faster than BUGS/JAGS: “Slower per iteration but much better at mixing and converging” Bob Carpenter; The hope is to trade-off wall time for shorter chains.
PyMC3

- Based on Hamiltonian Monte Carlo
- Require gradient information, calculated by Theano (fast; tightly integrated with NumPy)
- Model specification directly in Python code:

  “There should be one—and preferably only one—obvious way to do it”
  – Zen of Python

- Readings:
INLA

- Integrated nested Laplace approximation (Rue, Martino and Chopin (2009) JRSS-B)
- Suitable for latent Gaussian Markov random field models, e.g., Generalized additive models, Time series models, Geoadditive models... (recommend to your friends who do spatial statistics!)
- Fast for marginal posterior densities, hence summary statistics of interest, posterior means, variances or quantiles


Reference:
R Package “baker”: https://github.com/zhenkewu/baker

• Bayesian Analytic Kit for Etiology Research
• Call JAGS or WinBUGS from R
• Automatically write the full model file using an R wrapper function
• “Plug-and-Play” to add extra likelihood components and priors
• Built-in visualizations for interpreting results
Summary

• Modeler’s time:
  - model design/interpretation (iterative nature of modelling)
  - write one’s own code for posterior computing

• Surveyed software that does automatic posterior inference

• Choice of software depends on
  - Stage of model development (debugging or mass production)
  - Scale of analysis
  - Documentation and online community
  - R or Python as the primary data processing language

• P-spline regression done by different software; comparisons

• Introduced an R package “baker” for disease etiology research; used JAGS or WinBUGS; potential improvements
Comments

- Run and learn the workflow of the code for JAGS and Stan (from course website)

- Optional reading: