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

Thomas Bayes (1701-1761)

*figure from Wikipedia; some say this is not Bayes

Model likelihood for observed data x

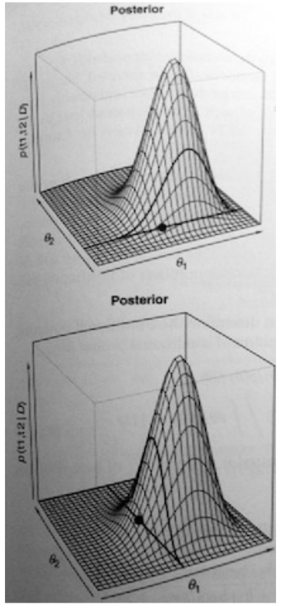
Prior on model parameter θ

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

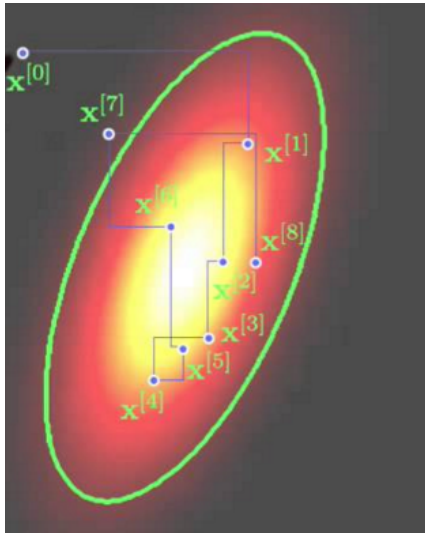
- Marginal distribution of data given the model;
- "Evidence" that this data x are generated by this model (Box 1980, JRSS-A)
- Exact computation possible (junction-tree algorithms), but hard for complex likelihood and priors (e.g., a graphical model with large tree-width, Dirichlet process prior etc.)

Gibbs Sampling

Use simulated samples to approximate the *entire* joint posterior distribution



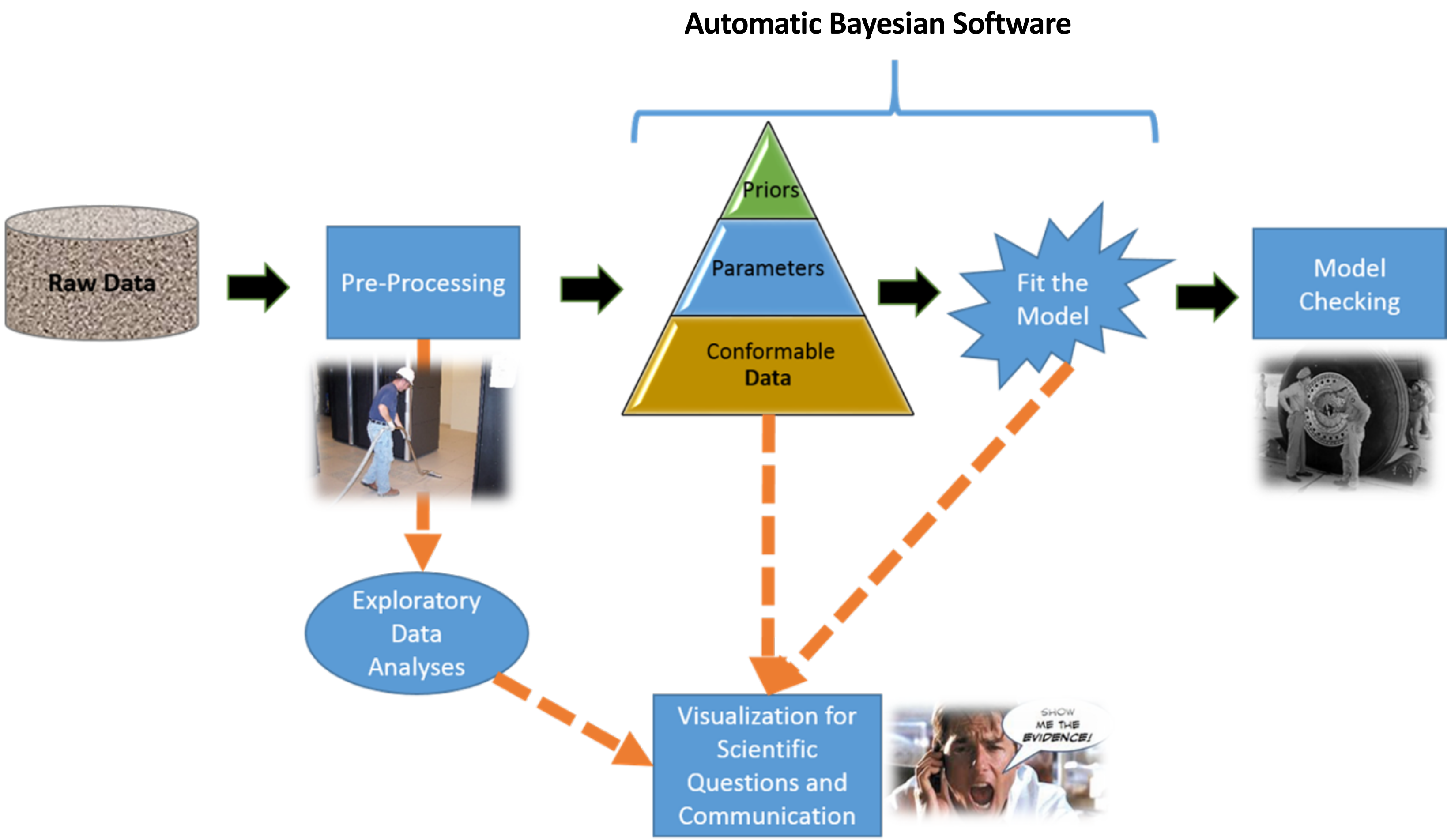
Look from the top



Why Automatic Software for Bayesian Inference?

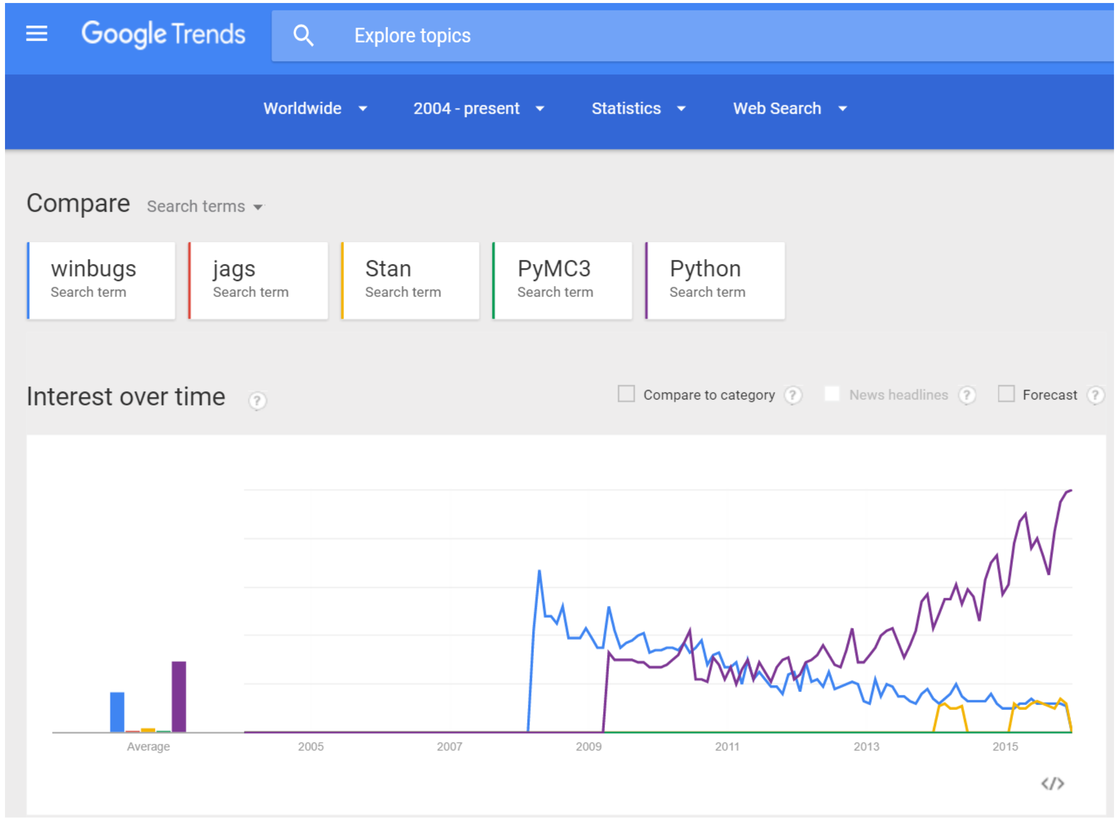
- Self-coded simulation algorithms usually require extra tuning and cost much time (share your experience)
- General formula/recipes exist for sampling from common distributions (adaptive rejection sampling, slice sampling, Metroplis-Hastings algorithm)
- Modelers generally want reasonable and fast model outputs to speed up model building, testing and interpretation

Analytic Pipeline



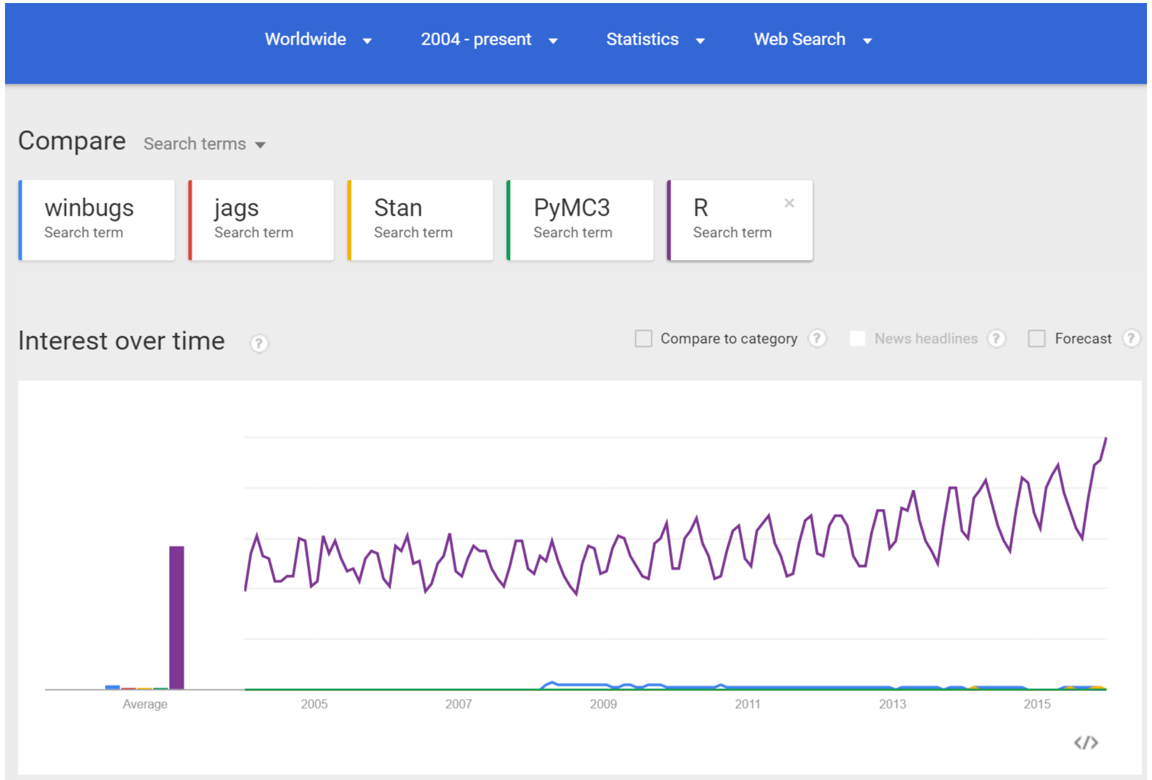
Bayesian Software and Google Trends

- 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/orflly

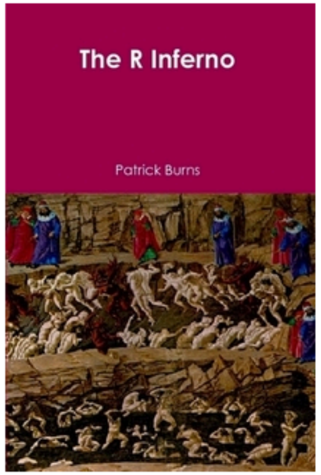
We will Connect These Software to *R*…

The R Inferno

By Patrick Burns

View this Author's Spotlight

Paperback, 153 Pages ☆☆☆☆☆ This item has not been rated yet



Preview

Price: £39.96

Ships in 3–5 business days

An essential guide to the trouble spots and oddities of R. In spite of the quirks exposed here, R is the best computing environment for most data analysis tasks. R is free, open-source, and has thousands of contributed packages. It is used in such diverse fields as ecology, finance, genomics and music. If you are using spreadsheets to understand data, switch to R. You will have safer -- and ultimately, more convenient -- computations.

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/

- **B**ayesian inference **U**sing **G**ibbs **S**ampling
- 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
- **Reference**: Lunn, D., Spiegelhalter, D., Thomas, A. and Best, N. (2009) The BUGS project: Evolution, critique and future directions (with discussion), *Statistics in Medicine* **28**: 3049--3082.

Good Experience - WinBUGS

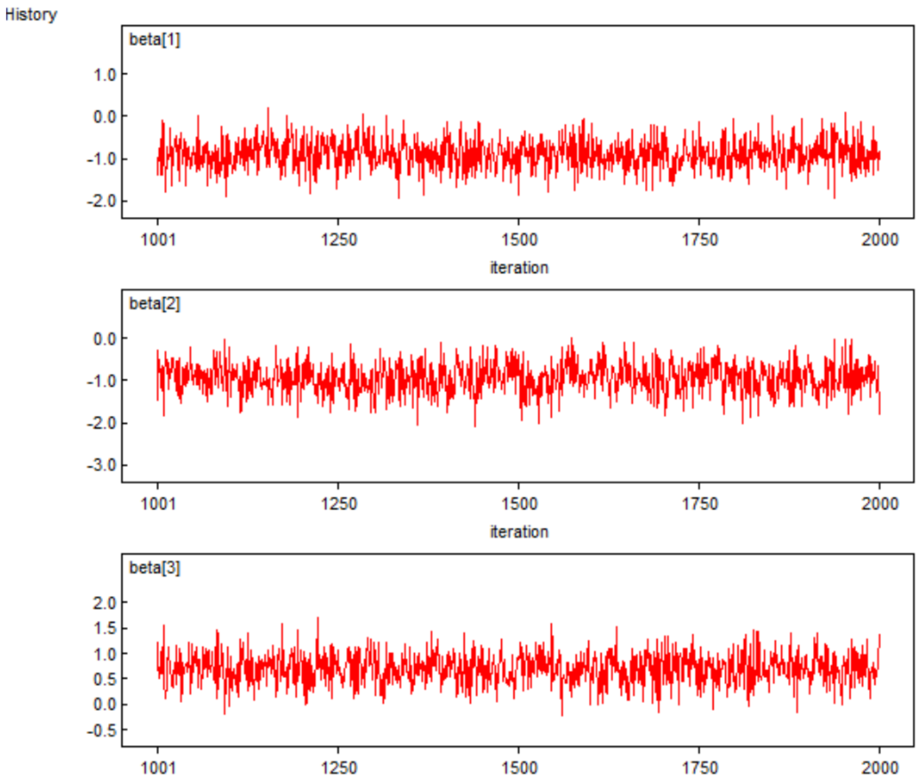
- GUI, easy for visual inspection of chains without too much posterior sample processing
- Good teaching tool with a companion book: The BUGS Book - A Practical Introduction to Bayesian Analysis
- Coded in many common distributions suitable for different types of data (see Manual)
- Relative easy for 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 Metroplis-Hastings algorithm proposed by Iteratively Reweighted Least Squares.

Example: Penalized-Spline Regression WinBUGS (500 data points; 10,000 iterations; 5.87 secs)

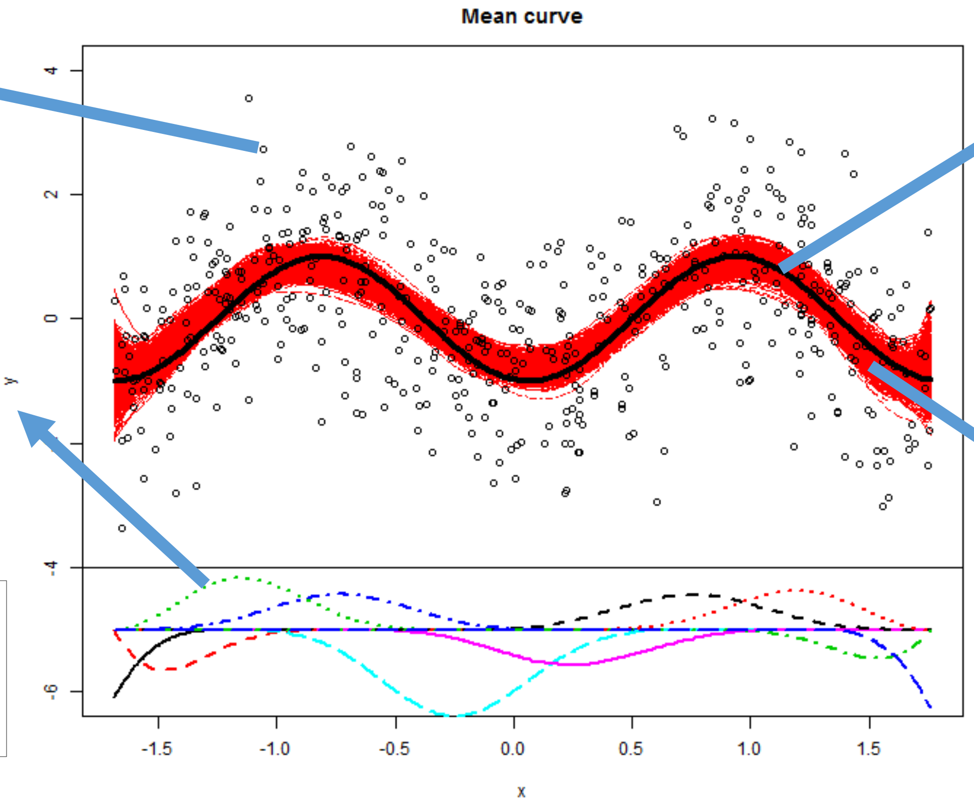
```
for (i in 1:N){
    M[i]~dnorm(mu[i],prec)
    #mu[i] <- inprod2(ZB[i,],beta[])
    mu[i] <- ZB[i,1]*beta[1]+ZB[i,2]*beta[2]+ZB[i,3]*beta[3]+ZB[i,4]*beta[4]+
          ZB[i,5]*beta[5]+ZB[i,6]*beta[6]+ZB[i,7]*beta[7]+ZB[i,8]*beta[8]+
          ZB[i,9]*beta[9]+ZB[i,10]*beta[10] # scalar calculations.
 }
 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)

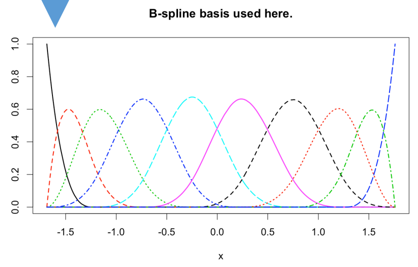
Data points

True mean curve



Posterior samples of mean curves

B-spline basis multiplied by estimated coefficients

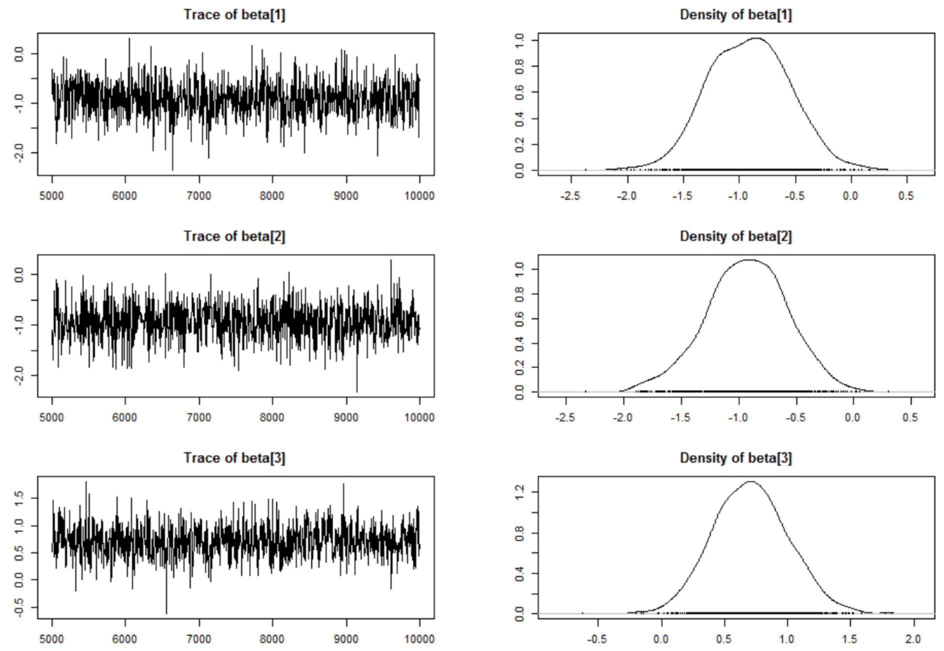


JAGS (Just Another Gibbs Sampler)

- http://mcmc-jags.sourceforge.net/
- 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)

```
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

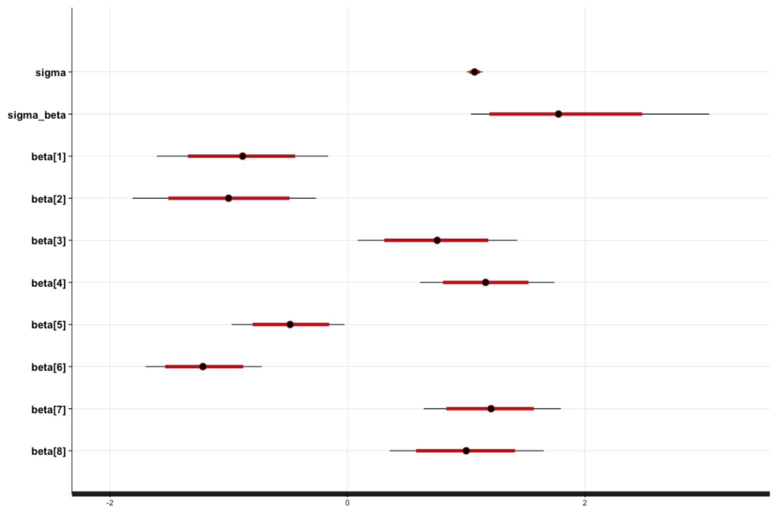
http://mc-stan.org/interfaces/

- 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.

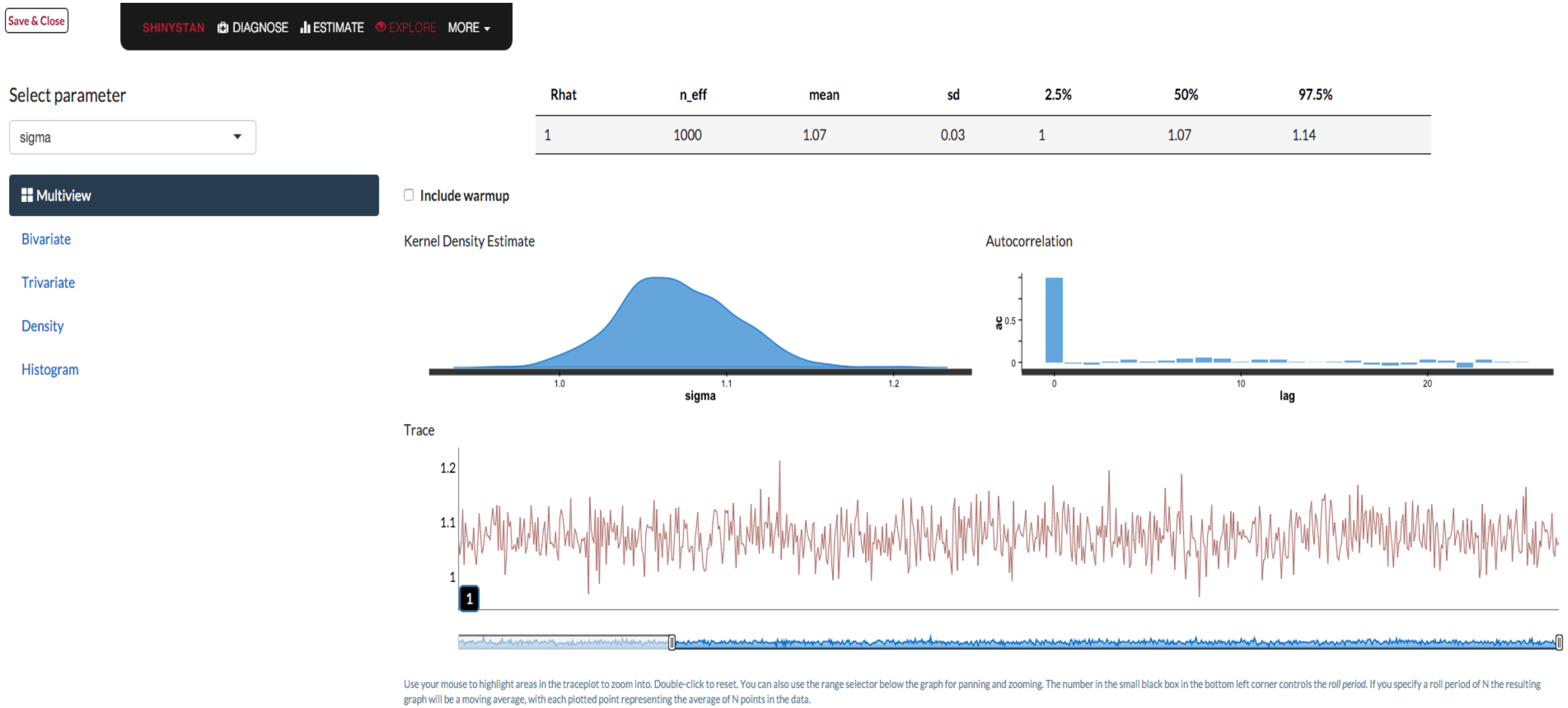
Example: Penalized-Spline Regression
Stan(10,000 iterations; 9.44 secs)

```
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);
}
```

Posterior Intervals



shinyStan



Save & Close
SHINYSTAN
DIAGNOSE
ESTIMATE
EXPLORE
MORE
Select parameter
sigma
Rhat n_eff mean sd 2.5% 50% 97.5%
1 1000 1.07 0.03 1 1.07 1.14
Multiview
Bivariate
Trivariate
Density
Histogram
Include warmup
Kernel Density Estimate
1.0
1.1
1.2
sigma
Autocorrelation
ac
0.5
0
10
20
lag
Trace
1.2
1.1
1
1
Use your mouse to highlight areas in the traceplot to zoom into. Double-click to reset. You can also use the range selector below the graph for panning and zooming. The number in the small black box in the bottom left corner controls the roll period. If you specify a roll period of N the resulting graph will be a moving average, with each plotted point representing the average of N points in the data.

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:
  - https://pymc-devs.github.io/pymc3/getting_started/
  - http://andrewgelman.com/2015/10/15/whats-the-one-thing-you-have-to-know-about-pystan-and-pymc-click-here-to-find-out/

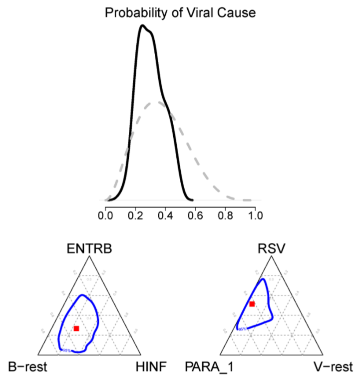
INLA

- 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

- More at: http://www.math.ntnu.no/~hrue/r-inla.org/doc/Intro/Intro.pdf
- **Reference:**
  Rue, H., Martino, S., & Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, *71*(2), 319-392.

R Package "baker": https://github.com/zhenkewu/baker

- **B**ayesian **A**nalytic **K**it for **E**tiology **R**esearch
- 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:
  - Casella, G, and Edward IG (1992). Explaining the Gibbs sampler. *The American Statistician* 46, 3: 167-174.
  - Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, *49*, 4:327-335.