Bayesian Methods

LABORATORY

Lesson 6: Feb 28 2002

Software: **BUGS**

Metropolis algorithm for generalized nonlinear models

The Generalized logit model. Beetles data

Example taken from Bliss (1935), and analysed in

Prentice, R. L., (1976) *A generalization of the Probit and Logit Methods for Dose Response Curves*, *Biometrics*, **32**, Issue 4, pp. 761-768;

Carlin, B. P., Louis, T. A., (2000) BAYES AND EMPIRICAL BAYES METHODS FOR DATA ANALYSIS, *2nd edition*.

BUGS programs (on the course web page):

I *Beetles: a generalized logit model* in **Beetles.b**

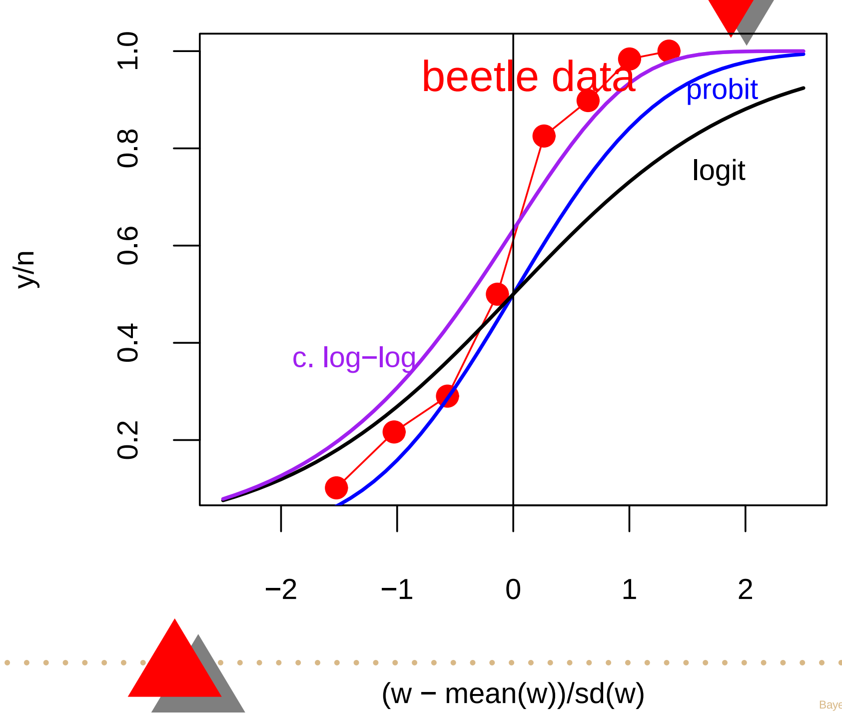
II *Beetles: choice of link function* in **BUGS Examples** Volume 2 (*http://www.mrc-bsu.cam.ac.uk/bugs/documentation/contents.shtml*)

Flour Beetle mortality data

Dosage w	# killed y	# exposed n
1.69	6.00	59.00
1.72	13.00	60.00
1.76	18.00	62.00
1.78	28.00	56.00
1.81	52.00	63.00
1.84	53.00	59.00
1.86	61.00	62.00
1.88	60.00	60.00

The data relate mortality of adult flour beetles to exposure.

The next figure plots (standardized) w versus y/n and overimposes the most common link functions for binomial data.



Issue: there is no closed form available for any of the three full conditionals needed to implement the Gibbs sampler.

Classic Bugs is not able to fit this model.
In fact, log-concavity requisite for updating nonconjugate full conditionals is not satisfied.

WinBUGS is able to sample from the full conditionals of Prentice model, by means of :

1. **Current point Metropolis** for continuous target distribution with unrestricted range
2. **Slice sampling** (Neal, 1997) for continuous target distribution with restricted range

Text cited from Prentice (1976)

The plot suggests that the dose response curve may be **skewed** for beetle data.

However, the usual χ^2 statistic does not suggest a lack of fit for either the probit or logit model.

It seems then opportune to embed the *logit*, *probit*, *complementary log-log* as well as other common models in a more general parametric setting, for

- a more specialized model testing: specific alternatives are tested relative to the more general one;
- a potential improvement in fit.

Slice sampling

Idea: Slice samplers simulate from a distribution, by **sampling uniformly from the region underneath the graph of the density function**.

An applet page with **a simulation of a one-dimensional "slice sampler" Markov chain - where the density $f(x)$ is considered as an auxiliary variable -** is in *http://markov.utstat.toronto.edu/jeff/java/slice.html*.

References:

R. Neal (1997) Markov Chain Monte Carlo Methods Based on 'Slicing' the Density Function. University of Toronto (available via anonymous ftp at ftp.cs.toronto.edu in the file /pub/radford/slice.ps.Z).
R. Neal (2000) Slice sampling. University of Toronto (available at www.cs.utoronto.ca in the file /pub/radford/slc-samp.ps.Z).

Other references to Slice Sampler and other Auxiliary Variable techniques are in **MCMC Preprint Service** (*http://www.statslab.cam.ac.uk/ mcmc/*) and at the applet page written above.

Generalized logit model *for y successes in n trials*

$$y_i \sim Bin(\pi_i, n_i) \quad i = 1, \ldots, k,$$

$$logit(\pi_i) = m_1(x_i - log(1 + e^{x_i})),$$

or, equivalently,

$$\pi_i = (e_i^x/(1 + e_i^x))^{m_1},$$

where

$$x_i = \frac{w_i - \mu}{\sigma}$$

Priors for a Bayesian analysis:

$$m_1 \sim Gamma(a, b)$$
$$\mu \sim Normal(c, d)$$
$$\sigma^2 \sim IGamma(e, f)$$

Various estimates from previous analyses

- *empirical* values: $\bar{\mu} = 1.793$, $\bar{\sigma} = 0.067$
- MLE: $\hat{\mu} = 1.818$, $\hat{\sigma} = 0.16$, $\hat{m_1} = 0.279$
- CL (Carlin and Louis, 2000):
  Metropolis algorithm for the transformed parameters $\boldsymbol{\theta} = (\mu, 1/2log\sigma^2, logm_1) \in R^3$ with Gaussian proposal density $N_3(\boldsymbol{\theta}^{t-1}, \Sigma)$, where

  1) $\Sigma = Diagonal(.00012, .033, .10)$: results in table 1 indicate slow mixing (extremely high auto-correlation together with high cross-correlation) that determines low acceptance rate (13.5%) and slow convergence;

	.025	.5	.975	lag1 acf
μ	1.78	1.81	1.83	0.981
σ	0.01	0.02	0.03	0.975
m_1	0.20	0.38	0.78	0.984

Table 1: Diagonal proposal covariance matrix

Cross-correlation:
$cor(\mu, log\sigma) = -0.78$,
$cor(\mu, logm_1) = -0.94$,
$cor(m_1, log\sigma) = 0.89$.

2) nondiagonal $\Sigma = 2 \cdot \hat{\Sigma}$, with $\hat{\Sigma} = \frac{1}{N}\sum_{j=1}^{N}(\boldsymbol{\theta}_j - \bar{\boldsymbol{\theta}})(\boldsymbol{\theta}_j - \bar{\boldsymbol{\theta}})^T$ estimate of the posterior covariance from the output of the first algorithm, in order to better mimic the posterior surface itself and accelerate convergence: results in table 2 indicate improved convergence with lower auto-correlation and acceptance rate increases to 27.3%.

	.025	.5	.975	lag1 acf
μ	1.78	1.81	1.83	0.834
σ	0.01	0.02	0.03	0.84
m_1	0.20	0.36	0.73	0.835

Table 2: Nondiagonal proposal covariance matrix

Beetles in WinBugs

1. **Noninformative priors for μ and σ^2**
   $m_1 \sim dgamma(1, 1)$
   $mu \sim dnorm(0, 0.001)$
   $tau \sim dgamma(0.001, 0.001)$

   As in CL Metropolis computation, parameter auto-correlation and cross-correlation are notably high. Then, our first attempt of lowering correlation is:
2. **Fixing μ^2 at its empirical value**
   We obtain this way an improved MCMC.
   Viceversa, fixing sigma does not improve significantly the MCMC. Besides, inferential results show to be very sensitive to the value plugged in.

It is noteworthy that mu is the node updated by Metropolis in WinBUGS, while m1 and tau are updated by Slice sampling.

3. **informative priors for μ and σ^2**
   We adopt the same priors as in LC. Yet, the more informative content does not improve significantly the MCMC.
   $m_1 \sim dgamma(.25, 0.25)$ –> mean=1, sd=2
   $mu \sim dnorm(2, 1/10)$
   $tau \sim dgamma(2.000004, 0.001)$ –> σ^2: mean=0.001 (0.032), sd=0.5 (0.707)

4. **Over relax** Over-relaxed form of MCMC (Neal, 1998) within-chain correlation is reduced (more efficient sampling) at the cost of increased time.

WinBUGS computation times

1. 12000 iterates, 3 chains: $108 sec$
2. 2000 iterates, 3 chains: $25 sec$
3. 10000 iterates, 2 chains: $64 sec$
4. 9000 iterates, 2 chains: $141 sec$

Reference to Over-relaxed form of MCMC:

Neal, R.M. (June, 1995) **Suppressing Random Walks in MCMC Using Ordered Overrelaxation** in *Learning in Graphical Models*, Kluwer Academic Publishers, Dordrecht, pp 205–230.

Available in MCMC Preprint Service (*http://www.statslab.cam.ac.uk/ mcmc/*) and also via anonymous ftp at ftp.cs.toronto.edu in the file /pub/radford/over.ps.Z.