Bayesian Modeling and Simulation Methods

Peter Rossi

ICE 2008

A Challenge

Consider the problem of approximating an unknown joint density. This problem arises frequently in econometrics for distributions of unobservables such as error terms/random coefficients.

There are many possible bases which could be used as the basis of an approximation.

Mixture of Normals is appealing.

A Challenge

$$p(y) \approx \sum_{k=1}^{K} \pi_k \varphi\left(y \middle| \mu_k, \Sigma_k\right)$$

Problems:

1. Very large number of parameters, e.g. dim(y)=5, K=10, n parm = (10-1) + 10 x 5 + 10 x 5 x 6/2 = 209
2. Optimization methods (however sophisticated) will fail. Likelihood has poles!
3. How can you make this truly non-parametric (e.g. make K adapt to N) and keep things smooth?

Bayesian Essentials

The Goal of Inference

Make inferences about unknown quantities using available information.

Inference -- make probability statements

unknowns --

parameters, functions of parameters, states or latent variables, “future” outcomes, outcomes conditional on an action

Information –

data-based

non data-based

theories of behavior; “subjective views” there is an underlying structure

parameters are finite or in some range

The likelihood principle

$$p(D \mid \theta) \equiv \ell(\theta)$$

LP: the likelihood contains all information relevant for inference. That is, as long as I have same likelihood function, I should make the same inferences about the unknowns.

In contrast to modern econometric methods (GMM) which does not obey the likelihood principle (e.g., regression for a 0-1 binary dependent variable)

Implies analysis is done conditional on the data, in contrast to the frequentist approach where the sampling distribution is determined prior to observing the data.

Bayes theorem

$$p(\theta|D) = \frac{p(D,\theta)}{p(D)} = \frac{p(D|\theta)p(\theta)}{p(D)}$$

$$p(\theta|D) \propto p(D|\theta)\, p(\theta)$$

Posterior $\propto$ "Likelihood" × Prior

Modern Bayesian statistics – simulation methods for generating draws from the posterior distribution $p(\theta|D)$.

Identification

$$R = \left\{ \theta : p\left(\text{Data} \middle| \theta\right) = k \right\}$$

If dim(R) >= 1, then we have an “identification” problem. That is, there are a set of observationally equivalent values of the model parameters. The likelihood is “flat” or constant over R.

But, Bayesian doesn’t care – unless he has a non-informative or flat prior!

Should the Bayesian care? Some functions of the parameters will be entirely influenced by prior.

Identification

define $\tau = \begin{pmatrix} \tau_1(\theta) \\ \tau_2(\theta) \end{pmatrix}$

such that $\dim(\tau_1) = \dim(R)$

and $p(\tau_1 | D) = p(\tau_1)$

τ_2 is the identified parameter. Report only on the posterior distribution of this function of θ.

Bayes Inference: Summary

Bayesian Inference delivers an integrated approach to:

- Inference – including “estimation” and “testing”
- Prediction – with a full accounting for uncertainty
- Decision – with likelihood and loss (these are distinct!)

Bayesian Inference is conditional on available info.

The right answer to the right question.

Bayes estimators are admissible. All admissible estimators are Bayes (Complete Class Thm).

Summarizing the posterior

Output from Bayesian Inf: $p(\theta|D)$

A high dimensional dist

Summarize this object via simulation:

marginal distributions of θ, $h(\theta)$

don't just compute $E[\theta|D]$, $Var(\theta|D)$

Contrast with Sampling Theory:

point est/standard error

summary of irrelevant dist

bad summary (normal)

Limitations of Asymptotics

Bayesian Regression

Bayesian Regression

<u>Prior:</u> $$p(\beta,\sigma^2) = p(\beta \mid \sigma^2)p(\sigma^2)$$

$$p(\beta \mid \sigma^2) \propto (\sigma^2)^{-k/2} \exp\left[-\frac{1}{2\sigma^2}(\beta-\bar{\beta})' A(\beta-\bar{\beta})\right]$$

$$p(\sigma^2) \propto (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left[-\frac{\nu_0 s_0^2}{2\sigma^2}\right]$$

Inverted Chi-Square: $$\sigma^2 \sim \frac{\upsilon_0 s_0^2}{\chi^2_{\upsilon_0}}$$

Chi-squared distribution

Note: χ^2 and gamma are the same distributions:

x ~ gamma(θ,k) $\Leftrightarrow$ $f(x) = \dfrac{\theta^k}{\Gamma(k)} x^{k-1} \exp[-\theta x]$

y ~ chi squared(ν) $\Leftrightarrow$ $f(y) = \dfrac{1}{2^{\nu/2}\Gamma(\nu/2)} y^{(\nu/2)-1} \exp[-y/2]$

$$\chi^2_\nu \sim gamma\left(\frac{1}{2}, \frac{\nu}{2}\right)$$

Posterior

$$p(\beta,\sigma^2 \mid D) \propto \ell(\beta,\sigma^2)\,p(\beta \mid \sigma^2)\,p(\sigma^2)$$

$$\propto (\sigma^2)^{-n/2} \exp\left[\frac{-1}{2\sigma^2}(y-X\beta)'(y-X\beta)\right]$$

$$\times (\sigma^2)^{-k/2} \exp\left[\frac{-1}{2\sigma^2}(\beta-\bar{\beta})'A(\beta-\bar{\beta})\right]$$

$$\times (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left[\frac{-\nu_0 s_0^2}{2\sigma^2}\right]$$

Combining quadratic forms

$$(y-X\beta)'(y-X\beta)+(\beta-\bar{\beta})'A(\beta-\bar{\beta})$$
$$=(y-X\beta)'(y-X\beta)+(\bar{\beta}-\beta)'U'U(\bar{\beta}-\beta)$$
$$=(v-W\beta)'(v-W\beta)$$

$$v=\begin{bmatrix} y \\ U\bar{\beta} \end{bmatrix} \qquad W=\begin{bmatrix} X \\ U \end{bmatrix}$$

$$\boxed{(v-W\beta)'(v-W\beta)=\nu s^2+(\beta-\tilde{\beta})'W'W(\beta-\tilde{\beta})}$$

$$\tilde{\beta}=(W'W)^{-1}W'v=(X'X+A)^{-1}(X'X\hat{\beta}+A\bar{\beta})$$

$$\nu s^2=(v-W\tilde{\beta})'(v-W\tilde{\beta})=(y-X\tilde{\beta})'(y-X\tilde{\beta})+(\tilde{\beta}-\bar{\beta})'A(\tilde{\beta}-\bar{\beta})$$

Posterior

$$= (\sigma^2)^{-k/2} \exp\left[\frac{-1}{2\sigma^2}(\beta-\tilde{\beta})'(X'X+A)(\beta-\tilde{\beta})\right]$$

$$\times (\sigma^2)^{-\frac{n+\nu_0+2}{2}} \exp\left[\frac{-(\nu_0 s_0^2+\nu s^2)}{2\sigma^2}\right]$$

$$[\beta \mid \sigma^2] = N(\tilde{\beta}, \sigma^2(X'X+A)^{-1})$$

$$[\sigma^2] = \frac{\nu_1 s_1^2}{\chi^2_{\nu_1}} \quad \text{with} \quad \nu_1 = \nu_0 + n$$

$$s_1^2 = \frac{\nu_0 s_0^2 + \nu s^2}{\nu_0 + n}$$

Cholesky Roots

In Bayesian computations, the fundamental matrix operation is the Cholesky root. chol() in R

The Cholesky root is the generalization of the square root applied to positive definite matrices.

As Bayesians with proper priors, we don't ever have to worry about singular matrices!

$$\Sigma = U'U, \ \Sigma \text{ p.d.s.} \quad |\Sigma| = \left(\prod_i u_{ii} \right)^2$$

U is upper triangular with positive diagonal elements. U^{-1} is easy to compute by recursively solving TU = I for T, backsolve() in R.

Using Cholesky Roots

$$[\beta \mid \sigma^2] = N(\tilde{\beta}, \sigma^2 (X'X + A)^{-1})$$

$$\tilde{\beta} = (W'W)^{-1} W'v = (X'X + A)^{-1}(X'X\hat{\beta} + A\bar{\beta})$$

$$R'R = X'X + A \qquad R^{-1}\left(R^{-1}\right)' = \left(X'X + A\right)^{-1}$$

$$\tilde{\beta} = R^{-1}\left(R^{-1}\right)'\left(X'y + A\bar{\beta}\right)$$

$$\beta \mid y, X = \tilde{\beta} + \sigma R^{-1} z \qquad z \sim N\left(0, I\right)$$

IID Simulations

Scheme: $[y|X, \beta, \sigma^2]\ [\beta|\sigma^2]\ [\sigma^2]$

1) Draw $[\sigma^2 \mid y, X]$

2) Draw $[\beta \mid y, X, \sigma^2]$

3) Repeat

IID Simulator, cont.

$$\left[\beta \middle| y, X, \sigma^2\right] = N\left(\tilde{\beta}, \sigma^2 \left(X'X + A\right)^{-1}\right)$$

$$\tilde{\beta} = (X'X + A)^{-1}(X'X\hat{\beta} + A\bar{\beta})$$

$$\hat{\beta} = \left(X'X\right)^{-1} X'y$$

$$\text{note}: \theta \sim N(0, I);\ \beta = U'\theta + \tilde{\beta} \sim N\left(\tilde{\beta}, U'U = \sigma^2\left(X'X + A^{-1}\right)\right)$$

$$[\sigma^2 \,|\, y, X] = \frac{\nu_1 s_1^2}{\chi^2_{\nu_1}}$$

Shrinkage and Conjugate Priors

The Bayes Estimator is the posterior mean of β.

This is a "shrinkage" estimator.

$\tilde{\beta} = (X'X + A)^{-1}(X'X\hat{\beta} + A\bar{\beta})$ shrinks $\hat{\beta} \to \bar{\beta}$

as $n \to \infty$, $\tilde{\beta} \to \hat{\beta}$ (Why? X'X is of order n).

$\mathrm{Var}\left(\tilde{\beta} \middle| \sigma^2\right) = \sigma^2 (X'X + A)^{-1} < \sigma^2 A^{-1}$ or $\sigma^2 \left(X'X\right)^{-1}$

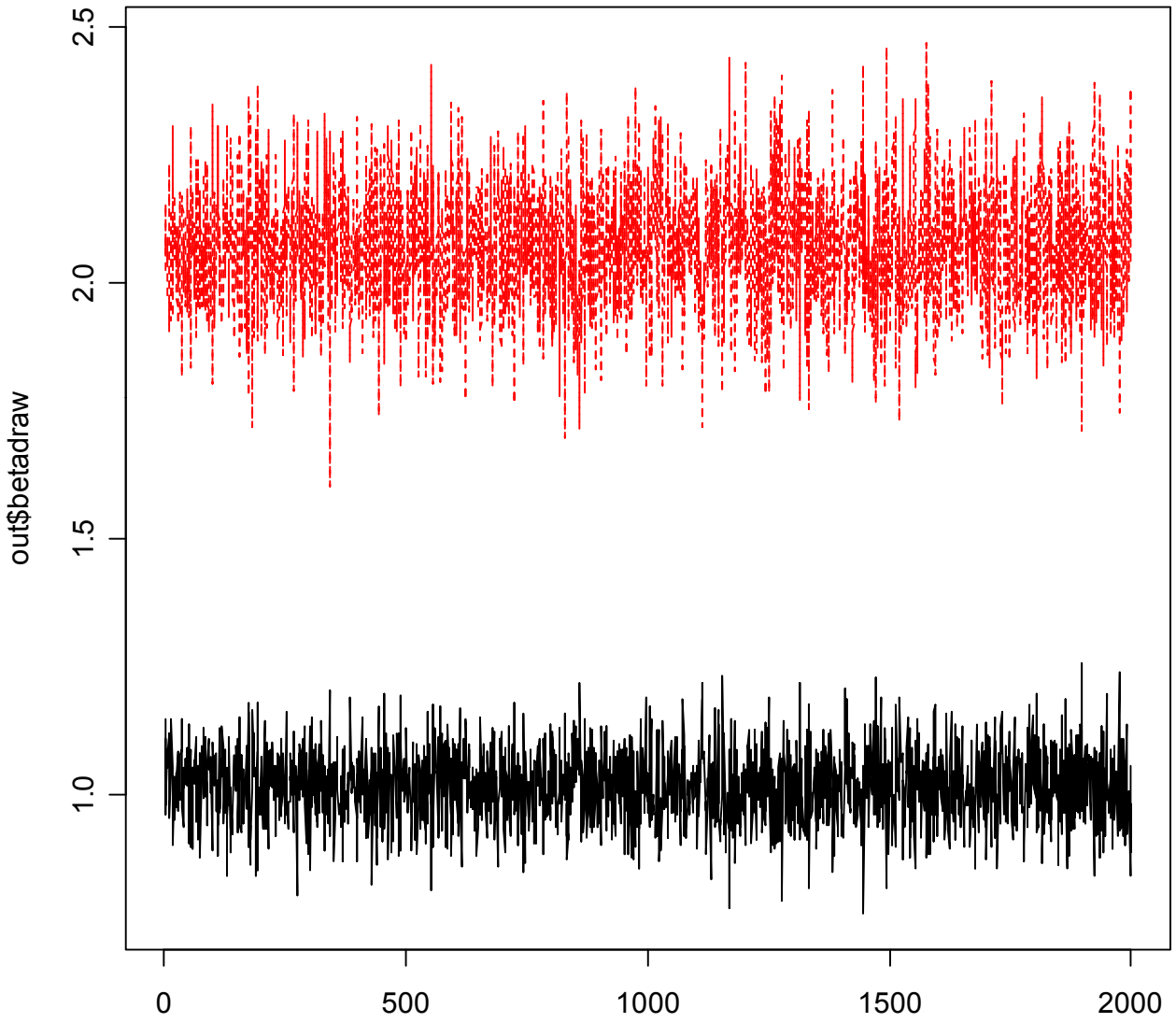
Is this reasonable?

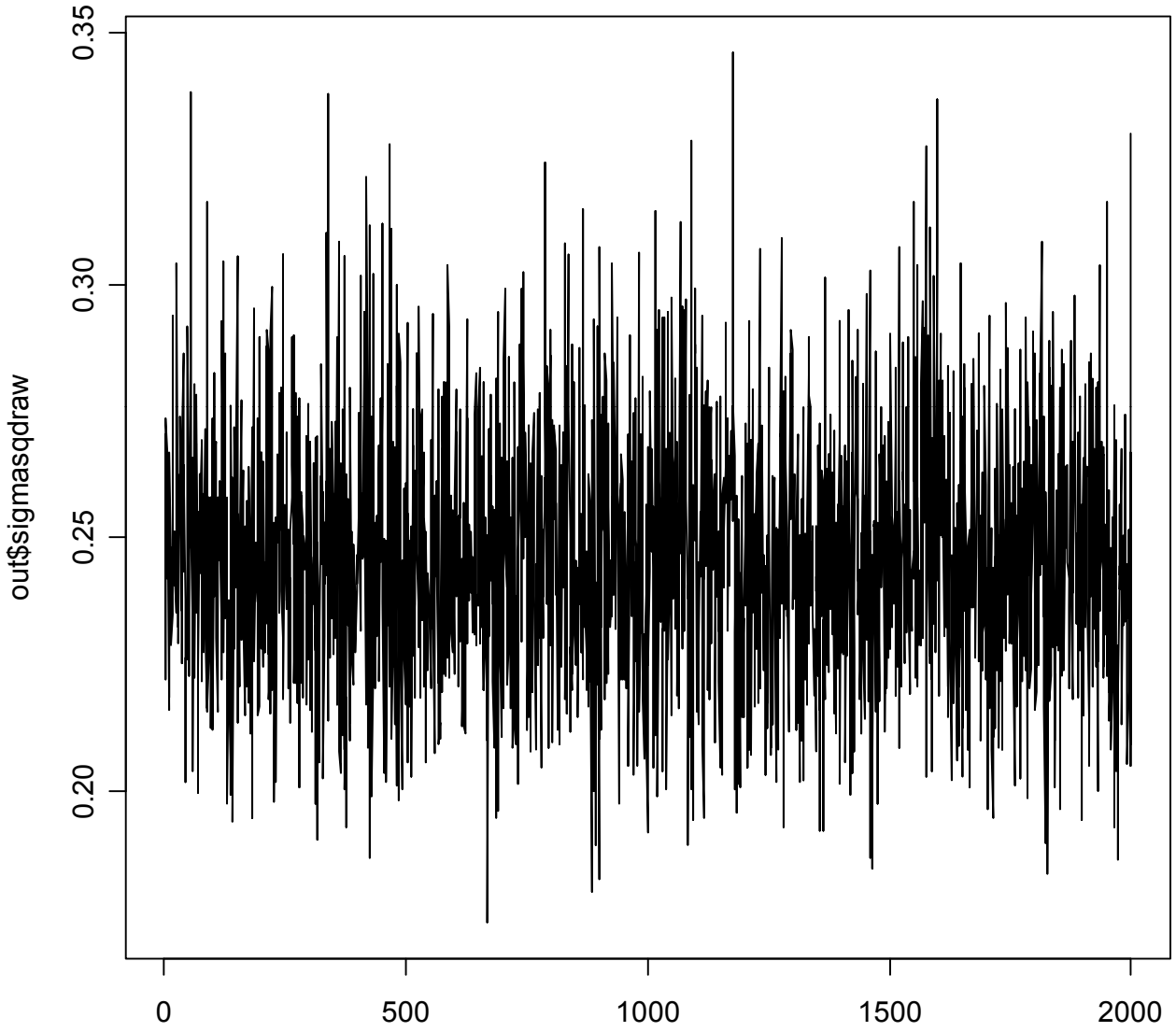
runireg

```
runireg=
function(Data,Prior,Mcmc){
#
# purpose:
#  draw from posterior for a univariate regression model with natural conjugate prior
#
# Arguments:
#   Data -- list of data
#          y,X
#   Prior -- list of prior hyperparameters
#     betabar,A      prior mean, prior precision
#     nu, ssq        prior on sigmasq
#   Mcmc -- list of MCMC parms
#     R number of draws
#     keep -- thinning parameter
#
# output:
#      list of beta, sigmasq draws
#         beta is k x 1 vector of coefficients
# model:
#      Y=Xbeta+e  var(e_i) = sigmasq
#      priors:  beta| sigmasq ~ N(betabar,sigmasq*A^-1)
#               sigmasq ~ (nu*ssq)/chisq_nu
```

runireg

```
RA=chol(A)
W=rbind(X,RA)
z=c(y,as.vector(RA%*%betabar))
IR=backsolve(chol(crossprod(W)),diag(k))
#      W'W=R'R ;  (W'W)^-1 = IR IR'  -- this is UL decomp
btilde=crossprod(t(IR))%*%crossprod(W,z)
res=z-W%*%btilde
s=t(res)%*%res
#
# first draw Sigma
#
sigmasq=(nu*ssq + s)/rchisq(1,nu+n)
#
# now draw beta given Sigma
#
beta = btilde + as.vector(sqrt(sigmasq))*IR%*%rnorm(k)
list(beta=beta,sigmasq=sigmasq)
}
```


out$betadraw
2.5
2.0
1.5
1.0
0
500
1000
1500
2000


out$sigmasqdraw
0.20
0.25
0.30
0.35
0
500
1000
1500
2000

Multivariate Regression

$$
\begin{aligned}
y_1 &= X\beta_1 + \varepsilon_1 \\
&\vdots \\
y_c &= X\beta_c + \varepsilon_c \\
&\vdots \\
y_m &= X\beta_m + \varepsilon_m
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
&Y = XB + E, \\
&Y = \left[y_1, \ldots, y_c, \ldots, y_m\right] \\
&B = \left[\beta_1, \ldots, \beta_c, \ldots, \beta_m\right] \\
&E = \left[\varepsilon_1, \ldots, \varepsilon_c, \ldots, \varepsilon_m\right] \\
&\varepsilon_{row} \sim \text{iid } N\left(0, \Sigma\right)
\end{aligned}
$$

Multivariate regression likelihood

$$
\begin{aligned}
p\left(Y \mid X, B, \Sigma\right) &\propto |\Sigma|^{-n/2} \exp\left\{-\frac{1}{2}\sum_{r=1}^{n}\left(y_r - B'x_r\right)'\Sigma^{-1}\left(y_r - B'x_r\right)\right\} \\
&= |\Sigma|^{-n/2} \operatorname{etr}\left\{-\frac{1}{2}\left(Y - XB\right)'\left(Y - XB\right)\Sigma^{-1}\right\} \\
&= |\Sigma|^{-(n-k)/2} \operatorname{etr}\left\{-\frac{1}{2}S\Sigma^{-1}\right\} \\
&\qquad \times |\Sigma|^{-k/2} \operatorname{etr}\left\{-\frac{1}{2}\left(B - \hat{B}\right)' X'X\left(B - \hat{B}\right)\Sigma^{-1}\right\}
\end{aligned}
$$

Multivariate regression likelihood

But,

$$\mathrm{tr}(A'B) = \left(\mathrm{vec}(A)\right)'\left(\mathrm{vec}(B)\right)$$

$$\left(B-\hat{B}\right)' X'X\left(B-\hat{B}\right)\Sigma^{-1} = \mathrm{vec}\left(B-\hat{B}\right)'\mathrm{vec}\left(X'X\left(B-\hat{B}\right)\Sigma^{-1}\right)$$

and

$$\mathrm{vec}(ABC) = \left(C'\otimes A\right)\mathrm{vec}(B)$$

$$= \mathrm{vec}\left(B-\hat{B}\right)'\left(\Sigma^{-1}\otimes X'X\right)\mathrm{vec}\left(B-\hat{B}\right)$$

therefore,

$$p\left(Y \mid X,B,\Sigma\right) \propto |\Sigma|^{-(n-k)/2}\,\mathrm{etr}\left\{-\frac{1}{2}S\Sigma^{-1}\right\}$$

$$\times\,|\Sigma|^{-k/2}\exp\left\{-\frac{1}{2}\left(\beta-\hat{\beta}\right)'\left(\Sigma^{-1}\otimes X'X\right)\left(\beta-\hat{\beta}\right)\right\}$$

Inverted Wishart distribution

Form of the likelihood suggests that natural conjugate (convenient prior) for Σ would be of the Inverted Wishart form:

$$p\left(\Sigma \middle| \upsilon_0, V_0\right) \propto \left|\Sigma\right|^{-(\upsilon_0+m+1)/2} \operatorname{etr}\left(-\tfrac{1}{2} V_0 \Sigma^{-1}\right)$$

denoted $\Sigma \sim IW\left(\upsilon_0, V_0\right)$

if $\upsilon_0 > m+2$, $E\left[\Sigma\right] = (\upsilon_0 - m - 1)^{-1} V_0$

if $\upsilon_0 > m+1$, proper

Wishart distribution (rwishart)

If $\Sigma \sim IW\left(\upsilon_0, V_0\right), \Sigma^{-1} \sim W\left(\upsilon_0, V_0^{-1}\right)$

if $\upsilon_0 > m + 1,\ E\left[\Sigma\right] = \upsilon_0 V_0^{-1}$

Generalization of χ^2 :

Let $\varepsilon_i \sim N_m(0,\Sigma)$ Then $W = \sum_{i=1}^{\nu} \varepsilon_i \varepsilon_i' \sim W(\upsilon,\Sigma)$

The diagonals are χ^2

Multivariate regression prior and posterior

Prior:

$$p(\Sigma, B) = p(\Sigma)p(B \mid \Sigma)$$

$$\Sigma \sim IW(\upsilon_0, V_0)$$

$$\beta \mid \Sigma \sim N\left(\bar{\beta}, \Sigma \otimes A^{-1}\right)$$

Posterior:

$$\Sigma \mid Y, X \sim IW(\upsilon_0 + n, V_0 + S)$$

$$\beta \mid Y, X, \Sigma \sim N\left(\tilde{\beta}, \Sigma \otimes (X'X + A)^{-1}\right)$$

$$\tilde{\beta} = \mathrm{vec}\left(\tilde{B}\right), \quad \tilde{B} = (X'X + A)^{-1}\left(X'X\hat{B} + A\bar{B}\right),$$

$$S = \left(Y - X\tilde{B}\right)'\left(Y - X\tilde{B}\right) + \left(\tilde{B} - \bar{B}\right)' A\left(\tilde{B} - \bar{B}\right)$$

Drawing from Posterior: rmultireg

```
rmultireg=
function(Y,X,Bbar,A,nu,V)
RA=chol(A)
W=rbind(X,RA)
Z=rbind(Y,RA%*%Bbar)
#   note:  Y,X,A,Bbar must be matrices!
IR=backsolve(chol(crossprod(W)),diag(k))
#                    W'W = R'R  &  (W'W)^-1 = IRIR'  -- this is the UL decomp!
Btilde=crossprod(t(IR))%*%crossprod(W,Z)
#                    IRIR'(W'Z) = (X'X+A)^-1(X'Y + ABbar)
S=crossprod(Z-W%*%Btilde)
#
rwout=rwishart(nu+n,chol2inv(chol(V+S)))
#
# now draw B given Sigma   note beta ~ N(vec(Btilde),Sigma (x) Cov)
#       Cov=(X'X + A)^-1  = IR t(IR)
#       Sigma=CICI'
#       therefore, cov(beta)= Omega = CICI' (x) IR IR' = (CI (x) IR) (CI (x) IR)'
#               so to draw beta we do beta= vec(Btilde) +(CI (x) IR)vec(Z_mk)
#                                          Z_mk is m x k matrix of N(0,1)
#               since vec(ABC) = (C' (x) A)vec(B), we have
#                            B = Btilde + IR Z_mk CI'
#
B = Btilde + IR%*%matrix(rnorm(m*k),ncol=m)%*%t(rwout$CI)
```

Introduction to the Gibbs Sampler and Data Augmentation

Simulating from Bivariate Normal

$$\theta \sim N\left(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$$

$$\theta_1 \sim N(0,1) \text{ and } \theta_2 \mid \theta_1 \sim N\left(\rho\theta_1, \left(1-\rho^2\right)\right)$$

In R, we would use the Cholesky root to simulate:

$$\theta_1 \sim z_1$$

$$\theta_2 = \rho z_1 + \sqrt{\left(1-\rho^2\right)}\, z_2$$

$$\theta = Lz \;;\; z \sim N(0, I)$$

$$L = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{\left(1-\rho^2\right)} \end{bmatrix}$$

Gibbs Sampler

A joint distribution can always be factored into a marginal × a conditional. There is also a sense in which the conditional distributions fully summarize the joint.

$$\theta_2 \middle| \theta_1 \sim N\left(\rho\theta_1, \left(1-\rho^2\right)\right) \qquad \theta_1 \middle| \theta_2 \sim N\left(\rho\theta_2, \left(1-\rho^2\right)\right)$$

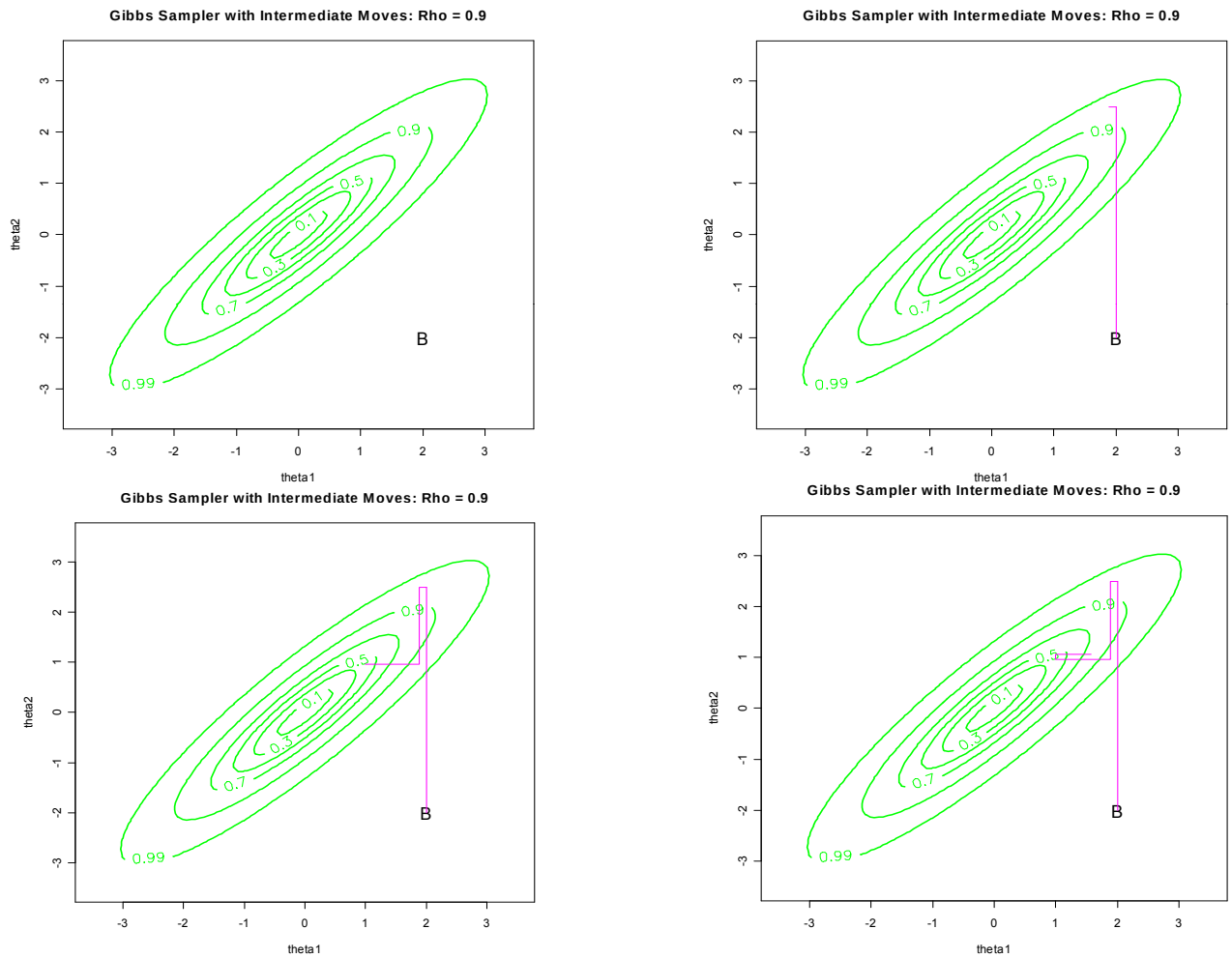
A simulator: Start at point θ_0

Draw θ_1 in two steps:

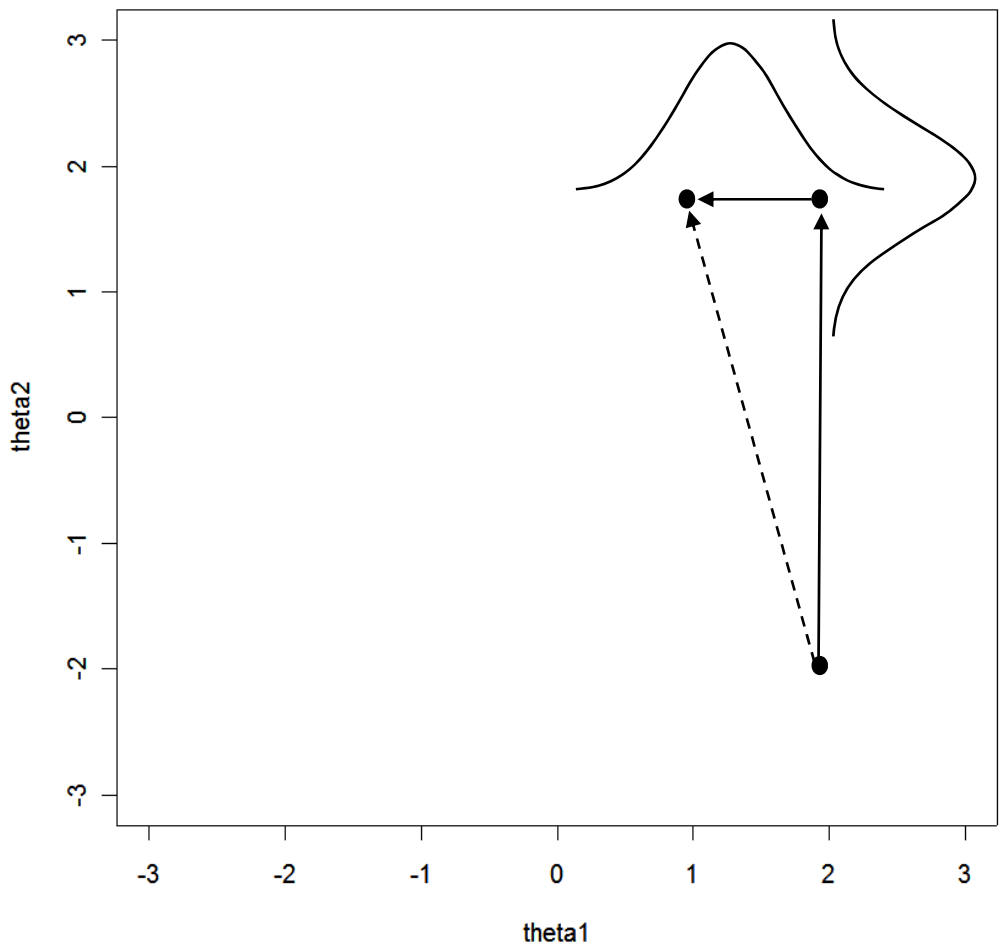
$$\theta_{1,2} \sim N\left(\rho\theta_{0,1}, 1-\rho^2\right)$$

$$\theta_{1,1} \sim N\left(\rho\theta_{1,2}, 1-\rho^2\right)$$

rbiNormGibbs



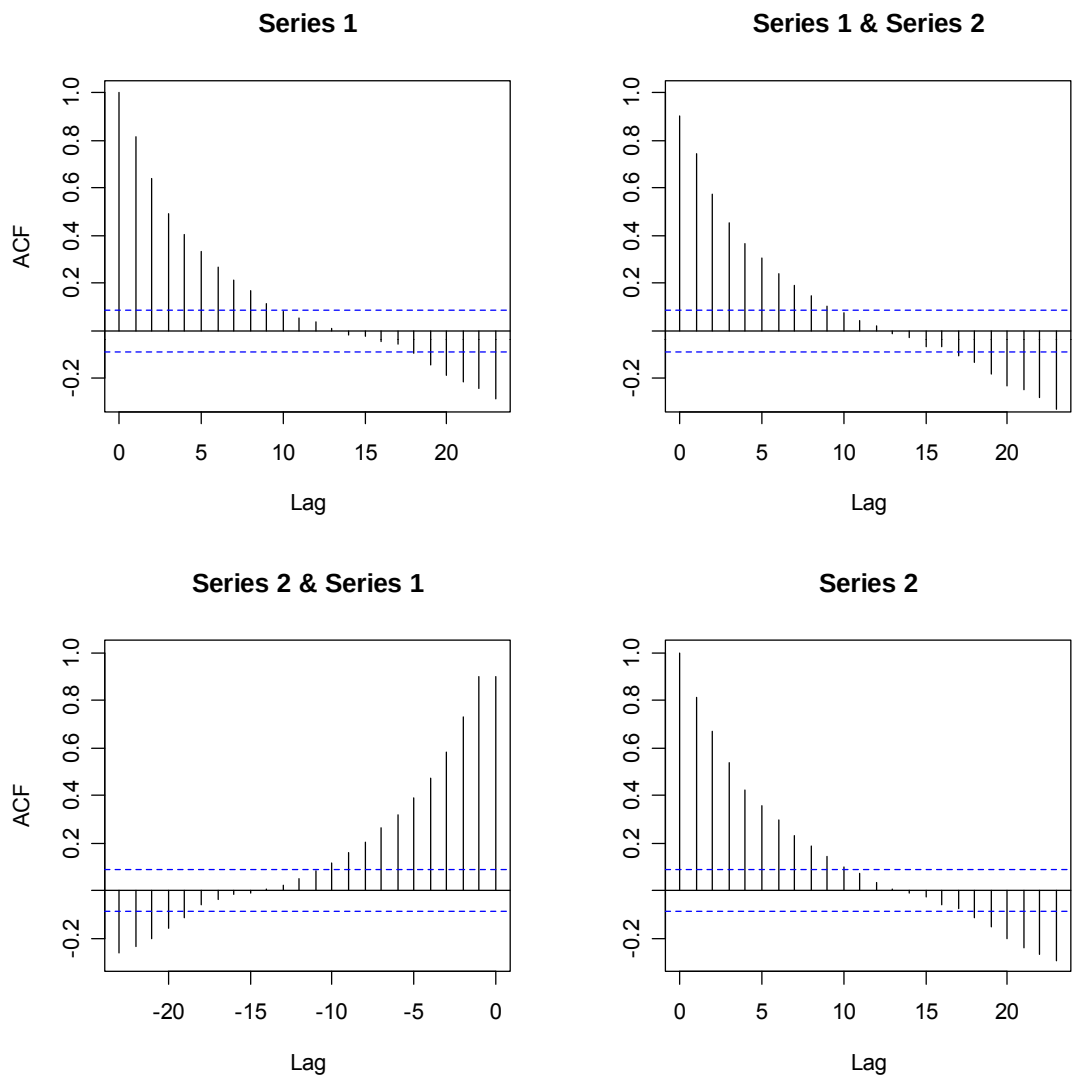
Intuition for dependence



This is a Markov Chain!

Average step size : $\sqrt{1-\rho^2}$

rbiNormGibbs



non-iid draws!

Who cares?

Loss of Efficiency

Gibbs Sampler is Non-IID

Gibbs Sampler defines a Markov chain (that is current value of draw summarizes entire past history of draws). The stationary distribution of this chain is the joint distribution of theta.

This means that we can estimate any aspect of the joint distribution using these sequence of draws.

estimate $\mu = E_\theta\left(g(\theta)\right) = \int g(\theta) p_\theta(\theta) d\theta;$

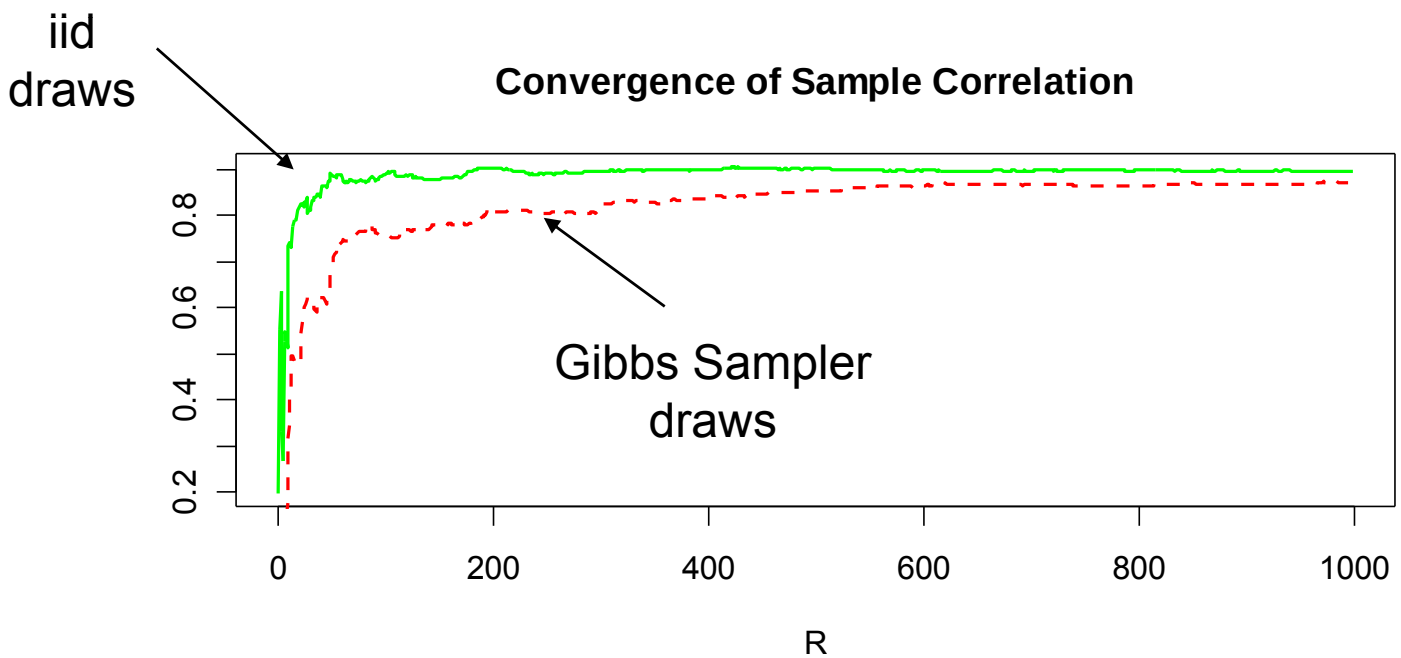
$\hat{\mu} = \frac{1}{R}\sum_r g\left(\theta^r\right)$ $\lim_{R\to\infty} \hat{\mu} = \mu$ (ergodic property)

i) $p = \Pr\left[\theta \in A\right] = \int_A p_\theta(\theta) d\theta$; $\hat{p} = \frac{1}{R} I_A\left(\theta^r\right)$

ii) $g(\theta) = \theta_i^m$

Ergodicity

$$\hat{\rho}_r = \frac{\frac{1}{r}\sum_{i=1}^{r}\left(\theta_1^i - \bar{\theta}_1\right)\left(\theta_2^i - \bar{\theta}_2\right)}{\sqrt{\frac{1}{r}\sum_{i=1}^{r}\left(\theta_1^i - \bar{\theta}_1\right)^2}\sqrt{\frac{1}{r}\sum_{i=1}^{r}\left(\theta_2^i - \bar{\theta}_2\right)^2}}$$



Relative Numerical Efficiency

Draws from the Gibbs Sampler come from a stationary yet autocorrelated process. We can compute the sampling error of averages of these draws.

Assume we wish to estimate $\mu = E_{\pi}\left[g(\theta)\right]$

We would use: $\hat{\mu} = \frac{1}{R}\sum_{r} g\left(\theta^{r}\right) = \frac{1}{R}\sum_{r} g^{r}$

$$\operatorname{var}\left(\hat{\mu}\right) = \frac{1}{R^{2}}\begin{bmatrix} \operatorname{var}\left(g^{1}\right) + \operatorname{cov}\left(g^{1},g^{2}\right) + \ldots + \operatorname{cov}\left(g^{1},g^{R}\right) + \\ \operatorname{cov}\left(g^{2},g^{1}\right) + \operatorname{var}\left(g^{2}\right) + \ldots + \operatorname{var}\left(g^{R}\right) \end{bmatrix}$$

Relative Numerical Efficiency

$$\mathrm{var}(\hat{\mu}) = \frac{\mathrm{var}(g)}{R}\left[1 + 2\sum_{j=1}^{R-1}\left(\frac{R-j}{R}\right)\rho_j\right] = \frac{\mathrm{var}(g)}{R}\boxed{f_R}$$

Ratio of variance to variance if iid.

$$\hat{f}_R = 1 + \sum_{j=1}^{m}\left(\frac{m+1-j}{m+1}\right)\hat{\rho}_j$$

Here we truncate the lag at m. Choice of m?

numEff in *bayesm* or use summary

General Gibbs sampler

$\theta' = (\theta_1, \theta_2, \ldots, \theta_p)$ "Blocking"

Sample from:

$$\theta_{1,1} = f_1(\theta_1 | \theta_{0,2}, \ldots, \theta_{0,p})$$
$$\theta_{1,2} = f_2(\theta_2 | \theta_{1,1}, \theta_{0,3}, \ldots, \theta_{0,p})$$

$$\theta_{1,p} = f_p(\theta_p | \theta_{1,1}, \ldots, \theta_{1,p-1})$$

to obtain the first iterate

where $f_i = \pi(\theta) / \int \pi(\theta)\, d\theta_{-i}$

$$\theta_{-i} = (\theta_1, \theta_2, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p)$$

Different prior for Bayes Regression

Suppose the prior for β does not depend on σ^2: $p(\beta,\sigma^2) = p(\beta)\, p(\sigma^2)$. That is, prior belief about β does not depend on σ^2. Why should views about depend on scale of error terms? Only true for data-based prior information NOT for subject matter information!

$$p(\beta) \propto \exp\left[\frac{-1}{2}(\beta-\bar{\beta})'A(\beta-\bar{\beta})\right]$$

$$p(\sigma^2) \propto (\sigma^2)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left[\frac{-\nu_0 s_0^2}{2\sigma^2}\right]$$

Different posterior

The posterior for σ^2 now depends on β:

$$[\beta \mid y, X, \sigma^2] = N(\tilde{\beta}, (\sigma^{-2}X'X + A)^{-1})$$

$$\text{with } \tilde{\beta} = (\sigma^{-2}X'X + A)^{-1}(\sigma^{-2}X'X\hat{\beta} + A\overline{\beta})$$

$$\hat{\beta} = (X'X)^{-1}X'y$$

$$[\sigma^2 \mid y, X, \beta] = \frac{\nu_1 s_1^2}{\chi^2_{\nu_1}} \text{ with } \nu_1 = \nu_0 + n$$

$$s_1^2 = \frac{\nu_0 s_0^2 + (y - X'\beta)'(y - X'\beta)}{\nu_0 + n}$$

Depends on β

Different simulation strategy

Scheme: $[y|X, \beta, \sigma^2]\,[\beta]\,[\sigma^2]$

1) Draw $[\beta \mid y, X, \sigma^2]$

2) Draw $[\sigma^2 \mid y, X, \beta]$ (conditional on β!)

3) Repeat

runiregGibbs

```
runiregGibbs=
function(Data,Prior,Mcmc){
#
# Purpose:
#   perform Gibbs iterations for Univ Regression Model using
#     prior with beta, sigma-sq indep
#
# Arguments:
#   Data -- list of data
#          y,X
#   Prior -- list of prior hyperparameters
#     betabar,A      prior mean, prior precision
#     nu, ssq        prior on sigmasq
#   Mcmc -- list of MCMC parms
#     sigmasq=initial value for sigmasq
#     R number of draws
#     keep -- thinning parameter
#
# Output:
#   list of beta, sigmasq
#
```

runiregGibbs (continued)

```
# Model:
#   y = Xbeta + e  e ~N(0,sigmasq)
#         y is n x 1
#         X is n x k
#         beta is k x 1 vector of coefficients
#
# Priors:  beta ~ N(betabar,A^-1)
#         sigmasq ~ (nu*ssq)/chisq_nu
#
#
# check arguments
#
.
sigmasqdraw=double(floor(Mcmc$R/keep))
betadraw=matrix(double(floor(Mcmc$R*nvar/keep)),ncol=nvar)
XpX=crossprod(X)
Xpy=crossprod(X,y)
sigmasq=as.vector(sigmasq)

itime=proc.time()[3]
cat("MCMC Iteration (est time to end - min) ",fill=TRUE)
flush()
```

runiregGibbs (continued)

```
for (rep in 1:Mcmc$R)
{
#
#     first draw beta | sigmasq
#
  IR=backsolve(chol(XpX/sigmasq+A),diag(nvar))
  btilde=crossprod(t(IR))%*%(Xpy/sigmasq+A%*%betabar)
  beta = btilde + IR%*%rnorm(nvar)
#
#    now draw sigmasq | beta
#
  res=y-X%*%beta
  s=t(res)%*%res
  sigmasq=(nu*ssq + s)/rchisq(1,nu+nobs)
  sigmasq=as.vector(sigmasq)
```

runiregGibbs (continued)

```
#
#print time to completion and draw # every 100th draw
#
  if(rep%%100 == 0)
    {ctime=proc.time()[3]
    timetoend=((ctime-itime)/rep)*(R-rep)
    cat(" ",rep," (",round(timetoend/60,1),")",fill=TRUE)
    flush()}

  if(rep%%keep == 0)
    {mkeep=rep/keep; betadraw[mkeep,]=beta; sigmasqdraw[mkeep]=sigmasq}
}
ctime = proc.time()[3]
cat('  Total Time Elapsed: ',round((ctime-itime)/60,2),'\n')

list(betadraw=betadraw,sigmasqdraw=sigmasqdraw)
}
```

R session

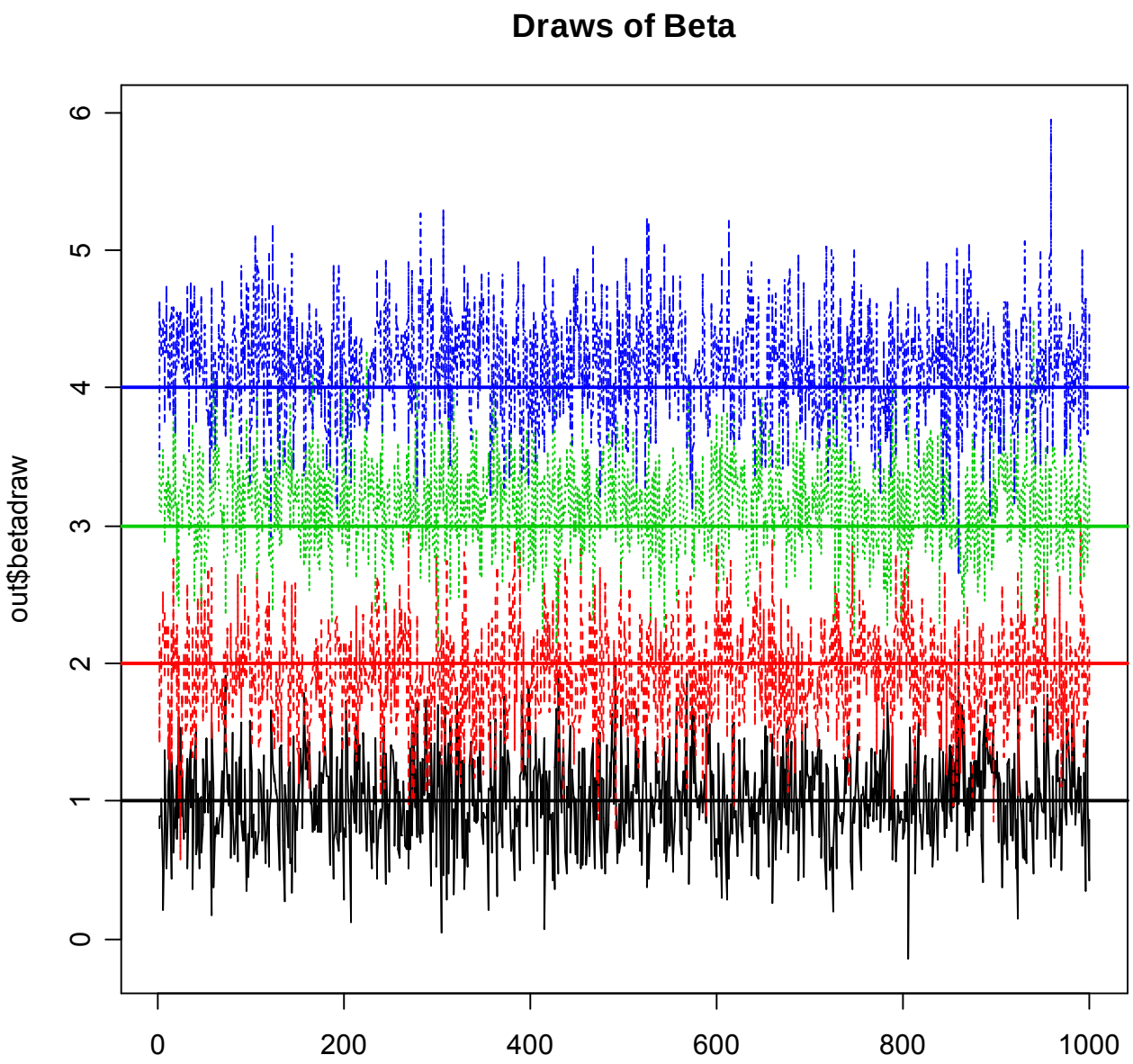
```
set.seed(66)
n=100
X=cbind(rep(1,n),runif(n),runif(n),runif(n))
beta=c(1,2,3,4)
sigsq=1.0
y=X%*%beta+rnorm(n,sd=sqrt(sigsq))

A=diag(c(.05,.05,.05,.05))
betabar=c(0,0,0,0)
nu=3
ssq=1.0

R=1000

Data=list(y=y,X=X)
Prior=list(A=A,betabar=betabar,nu=nu,ssq=ssq)
Mcmc=list(R=R,keep=1)

out=runiregGibbs(Data=Data,Prior=Prior,Mcmc=Mcmc)
```

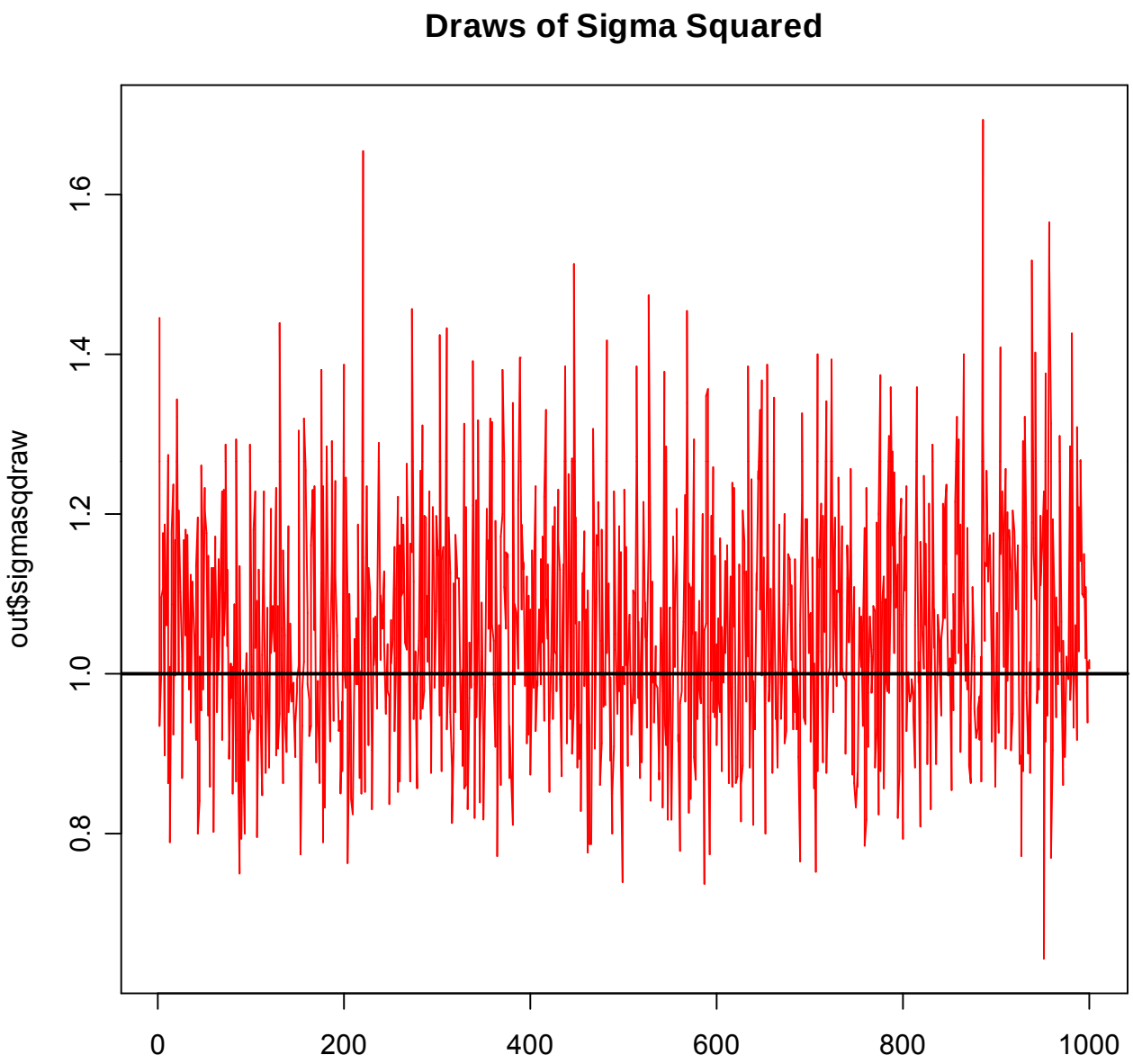
R session (continued)

```
Starting Gibbs Sampler for Univariate Regression Model
  with  100  observations

Prior Parms:
betabar
[1] 0 0 0 0
A
     [,1] [,2] [,3] [,4]
[1,] 0.05 0.00 0.00 0.00
[2,] 0.00 0.05 0.00 0.00
[3,] 0.00 0.00 0.05 0.00
[4,] 0.00 0.00 0.00 0.05
nu =  3  ssq=  1

MCMC parms:
R=  1000  keep=  1
```

R session (continued)

```
MCMC Iteration (est time to end - min)
  100  ( 0 )
  200  ( 0 )
  300  ( 0 )
  400  ( 0 )
  500  ( 0 )
  600  ( 0 )
  700  ( 0 )
  800  ( 0 )
  900  ( 0 )
  1000  ( 0 )
  Total Time Elapsed:  0.01
```


Draws of Beta
out$betadraw
0
1
2
3
4
5
6
0
200
400
600
800
1000


Draws of Sigma Squared
out$sigmasqdraw
0.8
1.0
1.2
1.4
1.6
0
200
400
600
800
1000

R session (continued)

```
> mat=apply(out$betadraw,2,quantile,probs=c(.01,.05,.5,.95,.99))
> mat=rbind(beta,mat); rownames(mat)[1]="beta"; print(mat)
          [,1]     [,2]     [,3]     [,4]
beta 1.0000000 2.000000 3.000000 4.000000
1%   0.2712523 1.006115 2.299525 3.228547
5%   0.4738900 1.280155 2.558585 3.455370
50%  1.0169590 1.886477 3.177056 4.156238
95%  1.5678797 2.560761 3.755547 4.810680
99%  1.7563197 2.799983 4.036885 5.043213


> quantile(out$sigmasqdraw,probs=c(.01,.05,.5,.95,.99))
       1%        5%       50%       95%       99%
0.7737674 0.8386931 1.0346436 1.3199761 1.4385439
>
```

Data Augmentation-Probit Ex

Consider the Binary Probit model:

$$z_i = x_i'\beta + \varepsilon_i \quad \varepsilon_i \sim N(0,1)$$

$$y_i = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Z is a latent, unobserved variable

$$p(y|x,\beta) = \int p(y,z|x,\beta)dz = \int p(y|z,x,\beta)p(z|x,\beta)dz$$

$$= \int f(z)p(z|x,\beta)dz$$

$$\Pr(y=1) = \int_0^\infty p(z|x,\beta)dz = \Pr(\varepsilon > -x'\beta) = \Phi(x'\beta)$$

$$\Pr(y=0) = \Phi(-x'\beta)$$

Integrate out z to obtain likelihood

Data augmentation

All unobservables are objects of inference, including parameters and latent variables.

For Probit, we desire the joint posterior of latents and β.

$$p(z,\beta \mid y) = p\left(z \middle| \beta, y\right) p\left(\beta \middle| z, y\right) = p\left(z \middle| \beta, y\right) p\left(\beta \middle| z\right)$$

Conditional independence of y,β.

Gibbs Sampler: $z \middle| \beta, y$

$\beta \middle| z$

Probit conditional distributions

$[z|\beta, y]$

This is a truncated normal distribution:

if y = 1, truncation is from below at $-x'\beta$

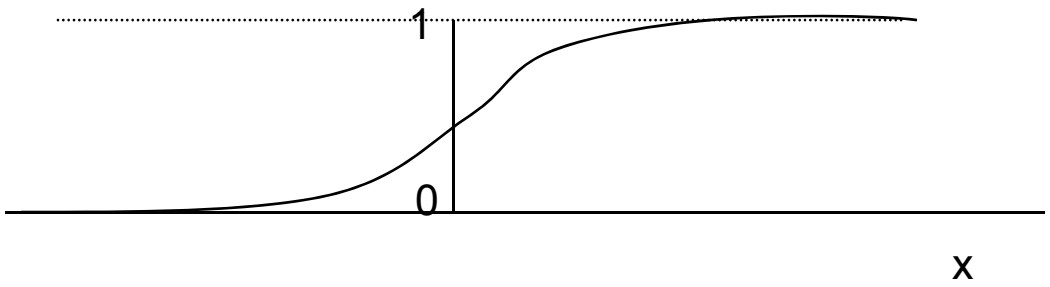
if y = 0, truncation is from above

How do we make these draws? We use the inverse CDF method.

Inverse cdf

If $X \sim F$

$U \sim \text{Uniform}[0,1]$ Then $F^{-1}(U) = X$



Let G be the cdf of X truncated to [a,b]

$$G(x) = \frac{F(x) - F(a)}{F(b) - F(a)}$$

Inverse cdf

what is G^{-1}? solve $G(x) = y$

$$\frac{F(x)-F(a)}{F(b)-F(a)} = y$$

$$F(x) = y(F(b)-F(a))+F(a)$$

$$x = F^{-1}(y(F(b)-F(a))+F(a))$$

$\Rightarrow$ Draw $u \sim U(0,1)$

$$x = F^{-1}(u(F(b)-F(a))+F(a))$$

rtrun

```
rtrun=
function(mu,sigma,a,b){
# function to draw from univariate truncated norm
# a is vector of lower bounds for truncation
# b is vector of upper bounds for truncation
#
FA=pnorm(((a-mu)/sigma))
FB=pnorm(((b-mu)/sigma))
mu+sigma*qnorm(runif(length(mu))*(FB-FA)+FA)
}
```

Probit conditional distributions

$$[\beta|z,X] \propto [z|X,\beta]\,[\beta]$$

$$[\beta\,|\bar{\beta},A^{-1}] \sim N(\bar{\beta},A^{-1})$$

$$[\beta\,|y,X] = \text{Normal}(\tilde{\beta},(X'X+A)^{-1})$$

$$\tilde{\beta} = (X'X+A)^{-1}(X'X\hat{\beta}+A\bar{\beta})$$

$$\hat{\beta} = (X'X)^{-1}X'z$$

rbprobitGibbs

```
rbprobitGibbs=
function(Data,Prior,Mcmc)
{
#
# purpose:
#   draw from posterior for binary probit using Gibbs Sampler
#
# Arguments:
#   Data - list of X,y
#     X is nobs x nvar, y is nobs vector of 0,1
#   Prior - list of A, betabar
#     A is nvar x nvar prior preci matrix
#     betabar is nvar x 1 prior mean
#   Mcmc
#     R is number of draws
#     keep is thinning parameter
#
# Output:
#   list of betadraws
# Model:   y = 1 if  w=Xbeta + e   > 0  e ~N(0,1)
#
# Prior:   beta ~ N(betabar,A^-1)
```

rbprobitGibbs (continued)

```
# define functions needed
#
breg1=
function(root,X,y,Abetabar)
{
# Purpose: draw from posterior for linear regression, sigmasq=1.0
#
# Arguments:
#  root is chol((X'X+A)^-1)
#  Abetabar = A*betabar
#
# Output:  draw from posterior
#
# Model: y = Xbeta + e  e ~ N(0,I)
#
# Prior:  beta ~ N(betabar,A^-1)
#
cov=crossprod(root,root)
betatilde=cov%*%(crossprod(X,y)+Abetabar)
betatilde+t(root)%*%rnorm(length(betatilde))
}
.
. (error checking part of code)
.
```

rbprobitGibbs (continued)

```
betadraw=matrix(double(floor(R/keep)*nvar),ncol=nvar)
beta=c(rep(0,nvar))
sigma=c(rep(1,nrow(X)))
root=chol(chol2inv(chol((crossprod(X,X)+A))))
Abetabar=crossprod(A,betabar)
        a=ifelse(y == 0,-100, 0)
        b=ifelse(y == 0, 0, 100)
#
#     start main iteration loop
#
itime=proc.time()[3]
cat("MCMC Iteration (est time to end - min) ",fill=TRUE)
flush()
```

rbprobitGibbs (continued)

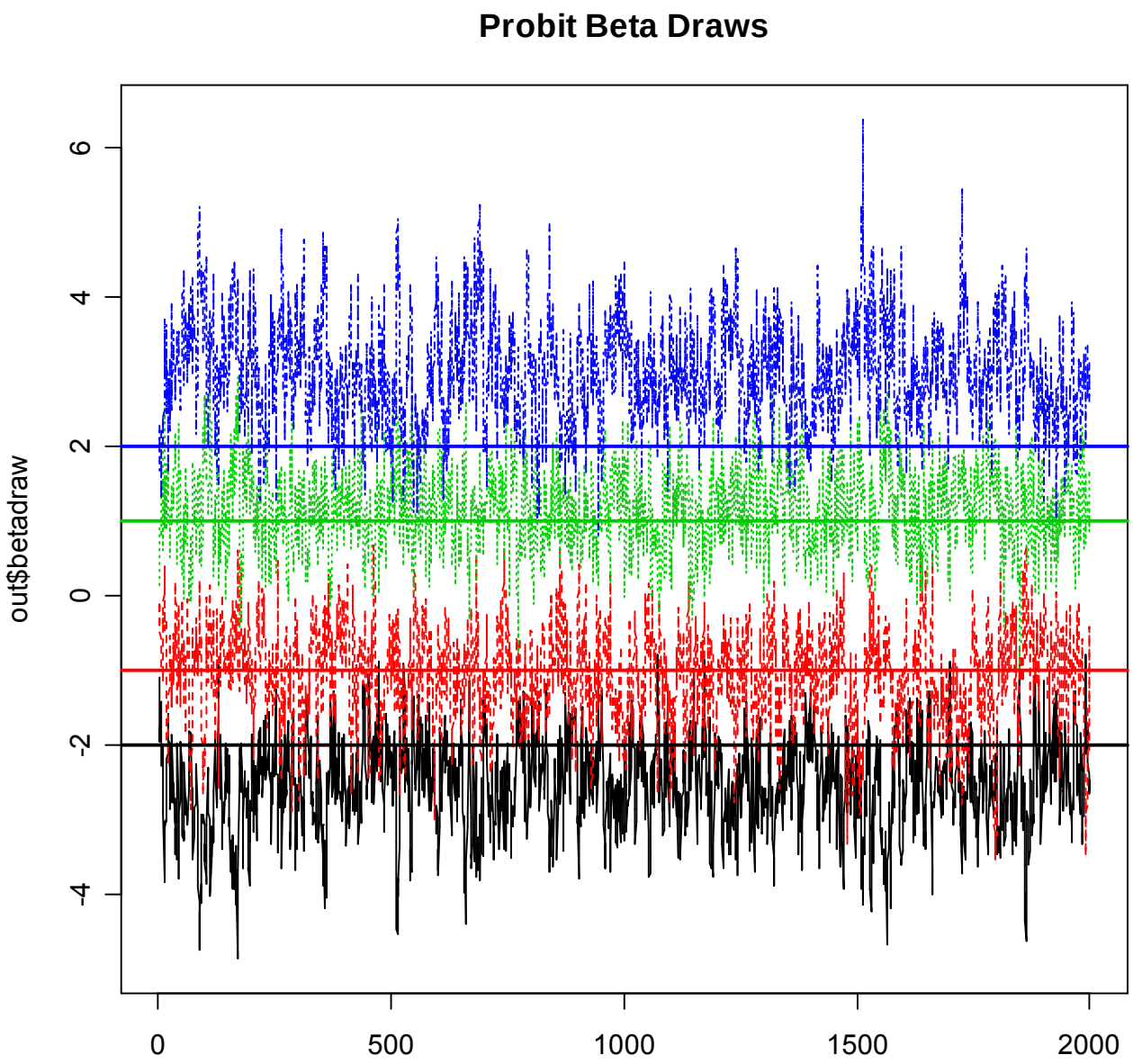
```
for (rep in 1:R)
{
  mu=X%*%beta
  z=rtrun(mu,sigma,a,b)
  beta=breg1(root,X,z,Abetabar)
}
```

Binary probit example

```
## rbprobitGibbs example
##
set.seed(66)
simbprobit=
function(X,beta) {
##  function to simulate from binary probit including x variable
y=ifelse((X%*%beta+rnorm(nrow(X)))<0,0,1)
list(X=X,y=y,beta=beta)
}
```

Binary probit example

```
nobs=100
X=cbind(rep(1,nobs),runif(nobs),runif(nobs),runif(nobs))
beta=c(-2,-1,1,2)
nvar=ncol(X)
simout=simbprobit(X,beta)

Data=list(X=simout$X,y=simout$y)
Mcmc=list(R=2000,keep=1)

out=rbprobitGibbs(Data=Data,Mcmc=Mcmc)

cat(" Betadraws ",fill=TRUE)
mat=apply(out$betadraw,2,quantile,probs=c(.01,.05,.5,.95,.99))
mat=rbind(beta,mat); rownames(mat)[1]="beta"; print(mat)
```

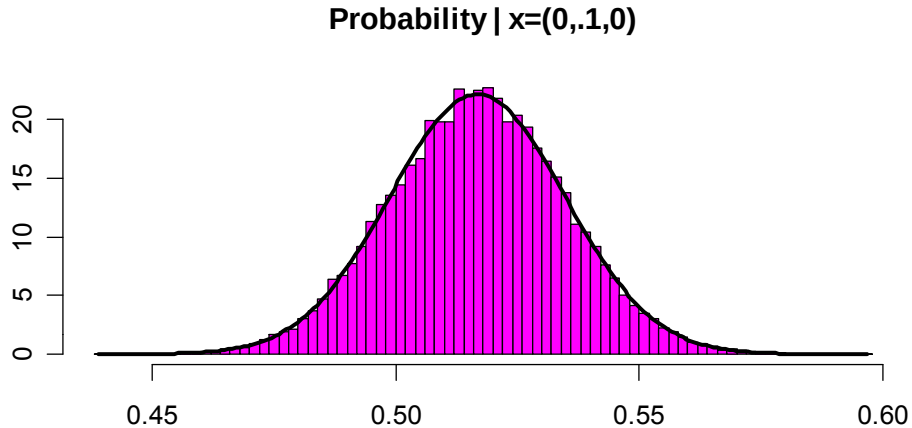

Probit Beta Draws
out$betadraw
6
4
2
0
-2
-4
0
500
1000
1500
2000

Summary statistics

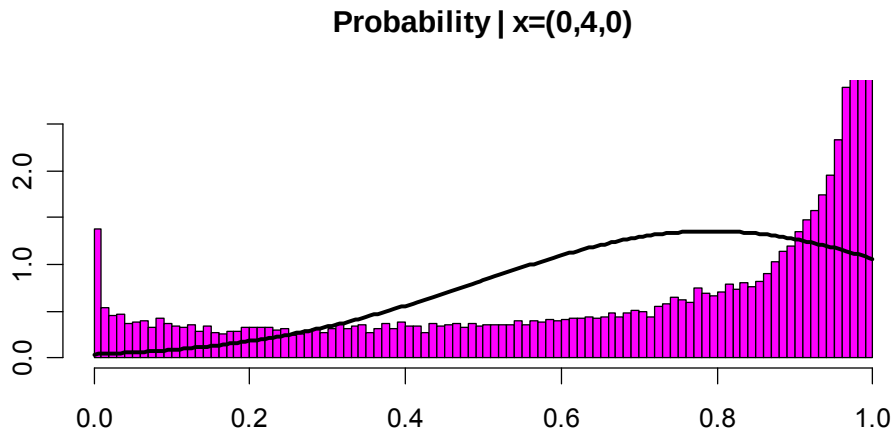
```
 Betadraws
          [,1]        [,2]        [,3]     [,4]
beta -2.000000 -1.00000000  1.00000000 2.000000
1%   -4.113488 -2.69028853 -0.08326063 1.392206
5%   -3.588499 -2.19816304  0.20862118 1.867192
50%  -2.504669 -1.04634198  1.17242924 2.946999
95%  -1.556600 -0.06133085  2.08300392 4.166941
99%  -1.233392  0.34910141  2.43453863 4.680425
```

Binary probit example

Example from BSM:



$$\Pr\left(y = 1 \middle| x, \beta\right) = \Phi\left(x'\beta\right)$$



Basic GS strategy- latent var models

1. draw $p(z|y,\theta)$
2. draw $p(\theta|y,z)$
3. repeat

$\Rightarrow$ yields: $p(z,\theta|y)$

discard draws of z to obtain: $p(\theta|y)$

Mixtures of normals

$$y_i \sim \sum_k \phi_k N(\mu_k, \Sigma_k)$$

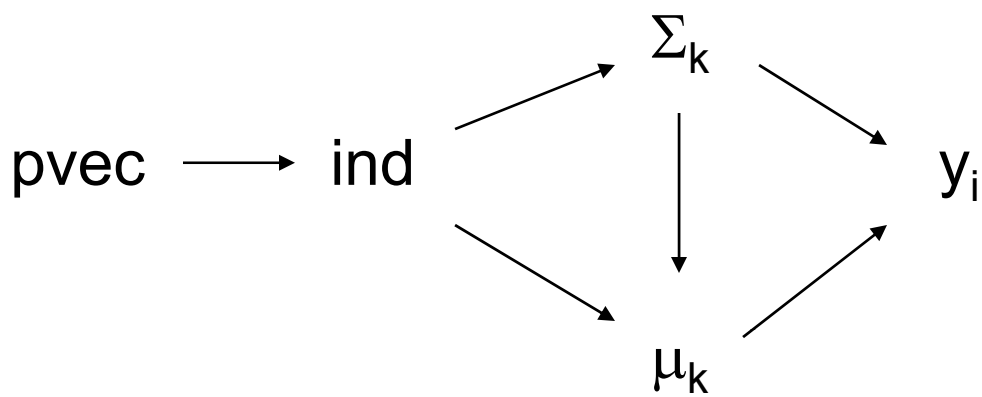
$$y_i \sim N(\mu_{ind}, \Sigma_{ind})$$

$$ind_i \sim \text{Multinomial}(\pi = \text{pvec})$$

A general flexible model or a non-parametric method of density approximation?

ind_i is a augmented variable that points to which normal distribution is associated with observation i. ind is an indicator variable that classifies observations one of the length(pvec) components.

Model hierarchy



Model

$[\text{pvec}]$
$[\text{ind}|\text{pvec}]$
$[\Sigma_k|\text{ind}]$
$[\mu_k|\text{ind},\Sigma_k]$
$[Y|\mu_k,\Sigma_k]$

Priors :

$\text{pvec} \sim \text{Dirichlet}(\alpha)$

$\mu_k \sim N\left(\bar{\mu},\Sigma_k \otimes a_\mu^{-1}\right)$

$\Sigma_k \sim IW(\upsilon, V)$

$k = 1,\ldots,K$

Conditionals

$[\text{pvec}|\text{ind},\text{priors}]$
$[\text{ind}|\text{pvec},\{\mu_k,\Sigma_k\},y]$
$[\{\mu_k,\Sigma_k\}|\text{ind},y,\text{priors}]$

Gibbs Sampler for Mixture of Normals

<u>Conditionals</u>

[pvec|ind,priors]

$$\text{pvec} \sim \text{Dirichlet}\left(\tilde{\alpha}\right)$$

$$\alpha_k = n_k + \alpha_k;\ n_k = \sum_{i=1}^{n} I\left(\text{ind}_i = k\right)$$

[ind|pvec,{μ_k,Σ_k},y]

$$\text{ind}_i \sim \text{multinomial}\left(\pi_i\right);\ \pi' = \left(\pi_{i,1},\ldots,\pi_{i,K}\right)$$

$$\pi_{i,k} = \text{pvec}_k \frac{\varphi\left(y_i \middle| \mu_k, \Sigma_k\right)}{\sum_k \varphi\left(y_i \middle| \mu_k, \Sigma_k\right)}$$

φ( ) is the multivariate normal density

Gibbs Sampler for Mixtures of Normals

$[\{\mu_k,\Sigma_k\}|\text{ind},y,\text{priors}]$

$$Y_k = \iota\mu_k' + U;\;\; U = \begin{bmatrix} u_1' \\ \vdots \\ u_{n_k}' \end{bmatrix};\; u_i \sim N\left(0,\Sigma_k\right)$$

given ind (classification), this is just a MRM!

$$\Sigma_k \left| \Theta_k^*, \upsilon, V \sim IW\left(\upsilon + n_k, V + S\right)\right.$$

$$\mu_k \left| \Theta_k^*, \Sigma_k, \bar{\mu}, a_\mu \sim N\left(\tilde{\mu}_k, {}^{1}\!/_{\left(n_k + a_\mu\right)}\Sigma_k\right)\right.$$

$$S = \left(\Theta_k^* - \iota\tilde{\mu}_k'\right)'\left(\Theta_k^* - \iota\tilde{\mu}_k'\right)$$

$$\tilde{\mu}_k = \left(n_k + a_\mu\right)^{-1}\left(n_k\bar{\theta}_k^* + a_\mu\bar{\mu}\right)$$

$$\bar{\theta}_k^* = \left(\Theta_k^{*'}\iota / n_k\right)'$$

Identification for Mixtures of Normals

Likelihood for mixture of K normals can have up to K! modes of equal height!

So-called “label” switching problem: I can permute the labels of each component without changing likelihood.

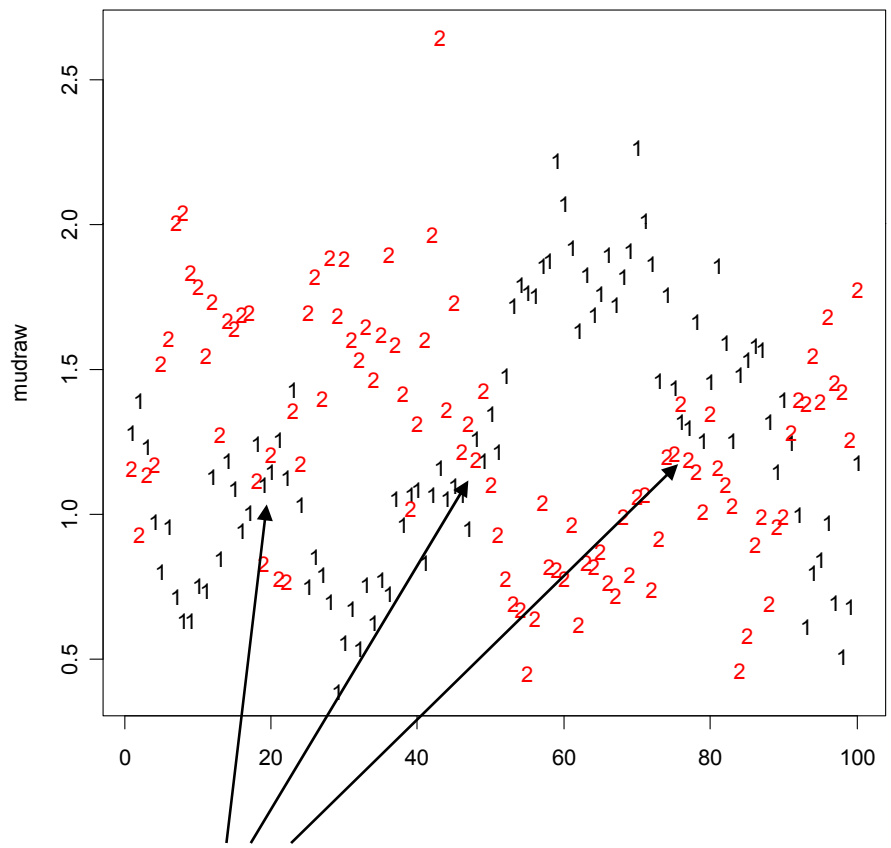
Implies the Gibbs Sampler may not navigate all modes! Who cares?

Joint density or any function of this is identified!

Label-Switching Example

Consider a mixture of two univariate normals that are not very "separated" and with a relatively small amount of data. Density of y is unimodal with mode a 1.5

$$y = .5N(1,1) + .5N(2,1)$$

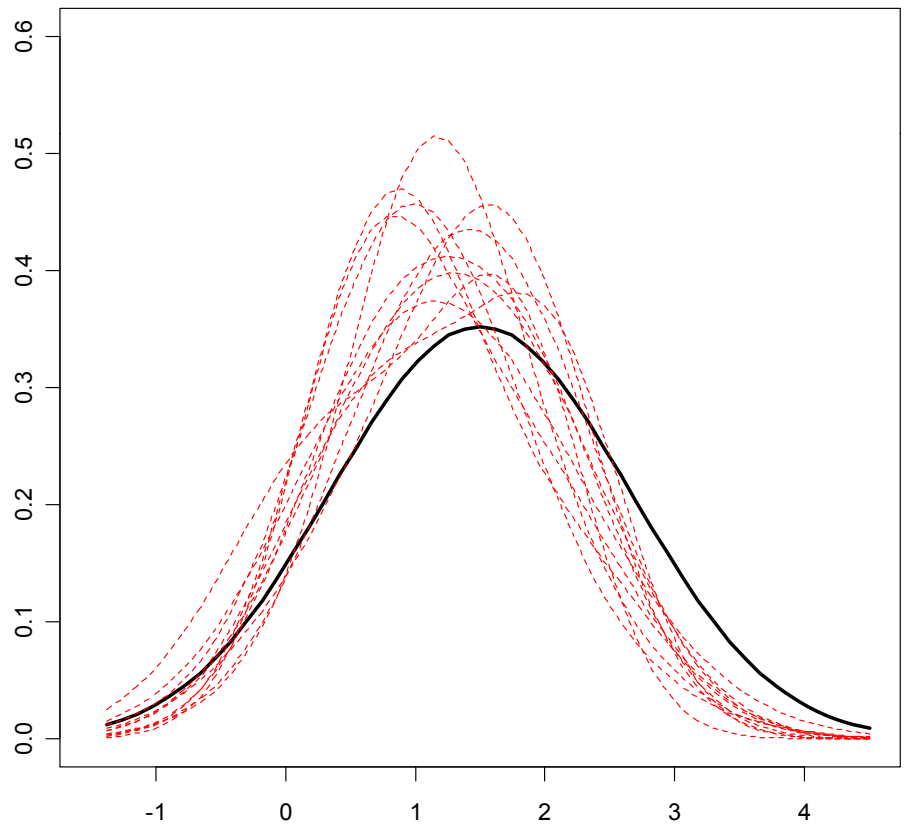


Label-switches

Label-Switching Example

$$p(y) = p\varphi(y|\mu_1,\sigma_1) + (1-p)\varphi(y|\mu_2,\sigma_2)$$

Density of y is identified. Using Gibbs Sampler, we get R draws from posterior of joint density



Identification for Mixtures of Normals

We use unconstrained Gibbs Sampler (rnmixGibbs). Others advocate restrictions or post-processing of draws to identify components

Pros:

- superior mixing
- focuses attention on identified quantities

Cons:

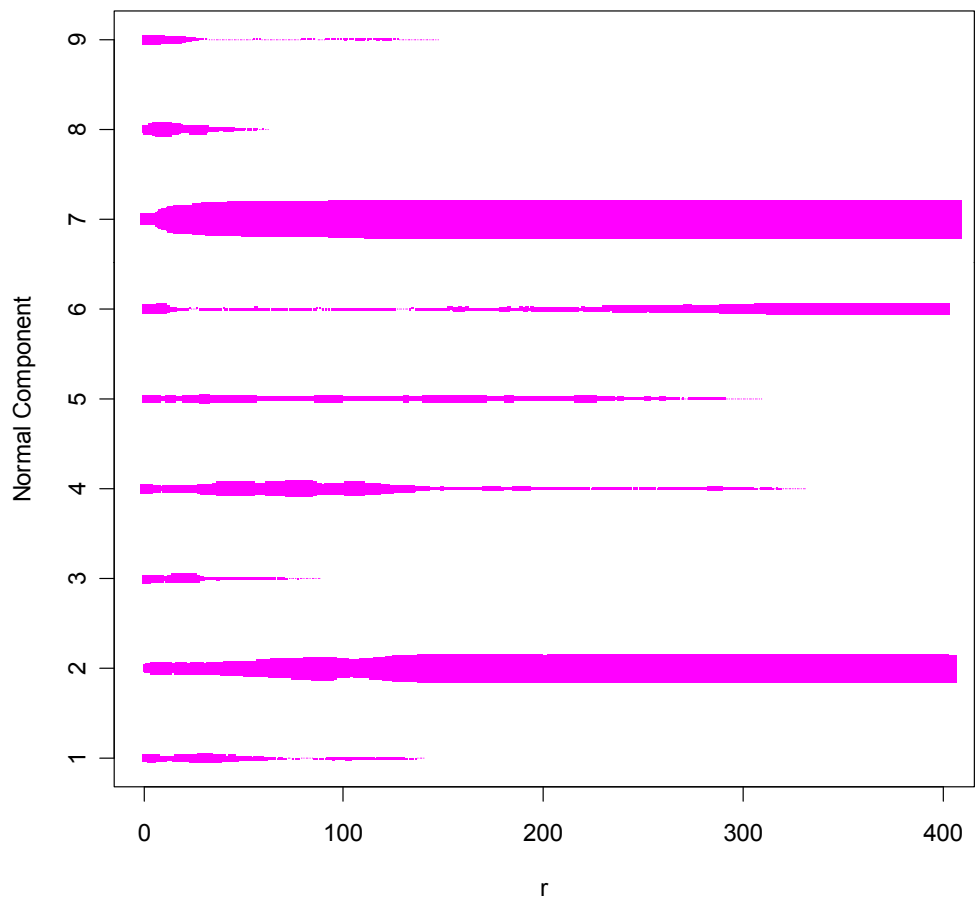
- can’t make inferences about component parms
- must summarize posterior of joint density!

Multivariate Mix of Norms Ex

$$\mu_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}; \ \mu_2 = 2\mu_1; \mu_3 = 3\mu_1;$$

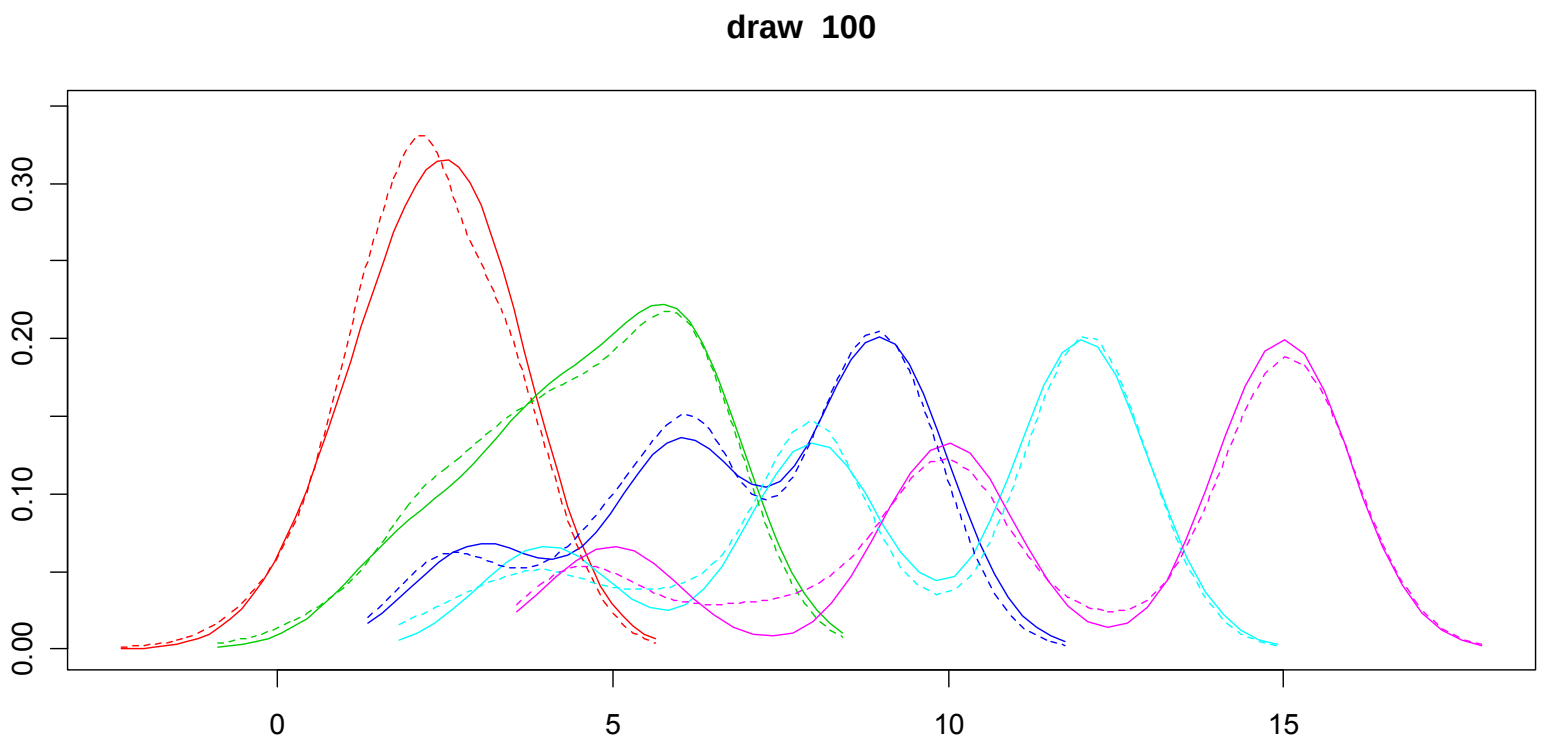
$$\Sigma_k = \begin{bmatrix} 1 & .5 & \cdots & .5 \\ .5 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & .5 \\ .5 & \cdots & .5 & 1 \end{bmatrix}$$

$$\text{pvec} = \begin{pmatrix} 1/2 \\ 1/3 \\ 1/6 \end{pmatrix}$$

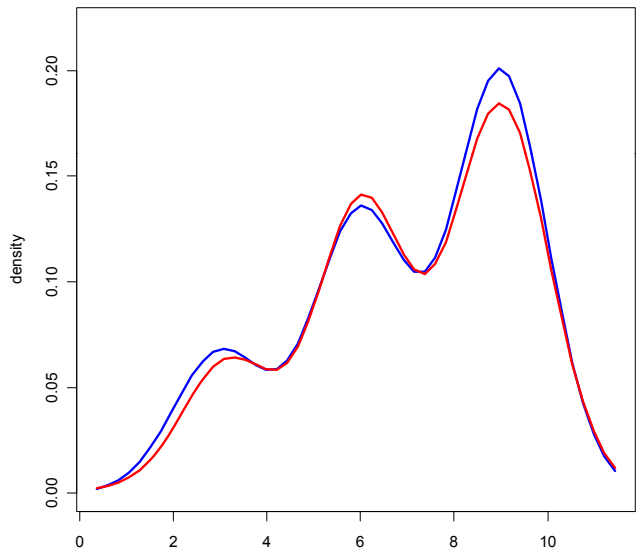


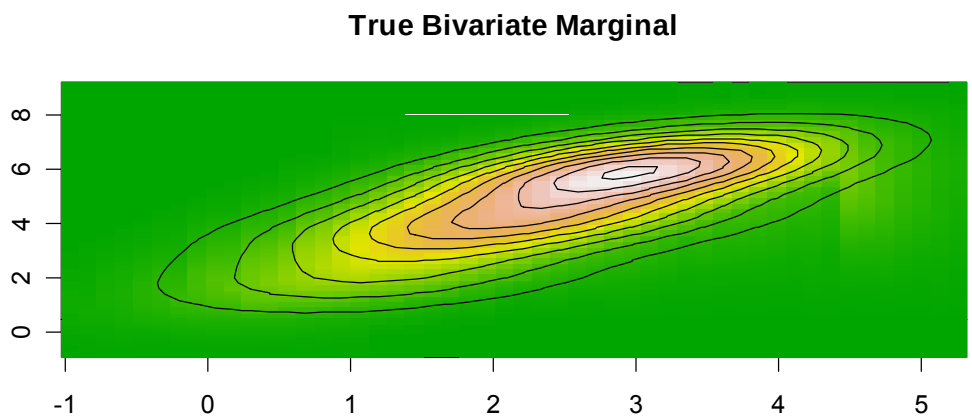
Multivariate Mix of Norms Ex

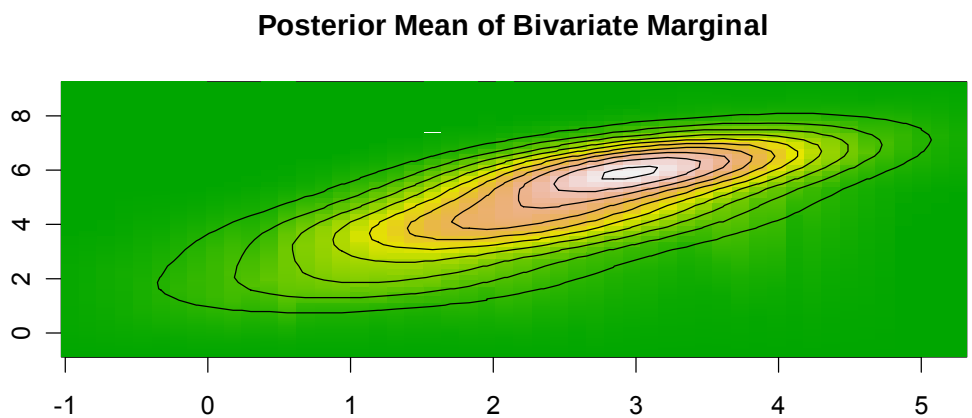
$$\hat{p}(y) = \frac{1}{R}\sum_{r=1}^{R}\hat{p}^{r}\left(y \middle| \left\{\mu_k^r, \Sigma_k^r\right\}, p^r\right) = \frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{K} p_k^r \varphi\left(y \middle| \mu_k^r, \Sigma_k^r\right)$$



Bivariate Distributions and Marginals







Multinomial probit model

$$y_i = f(z_i)$$

$$f(z_i) = \sum_{j=1}^{p} j \times I\left(\max(z_i) = z_{ij}\right)$$

$$z_i = X_i\delta + \upsilon_i, \quad \upsilon_i \sim \text{iid } N(0,\Omega)$$

$$\text{e.g.,} \quad X_i = \begin{bmatrix} & \text{price}_{i1} \\ I_p, & \vdots \\ & \text{price}_{ip} \end{bmatrix}$$

Identification Problem: I can add a scalar to z vector and change y!

Differenced system

$$y_i = f(w) = \sum_{j=1}^{p-1} j \times I\left(\max\left(w_i\right) = w_{ij} \text{ and } w_{ij} > 0\right) + p \times I\left(w < 0\right)$$

$$w_i = X_i^d \beta + \varepsilon_i, \qquad \varepsilon_i \sim N\left(0, \Sigma\right)$$

$$w_{ij} = z_{ij} - z_{ip}, \quad X_i = \begin{bmatrix} x'_{i1} \\ \vdots \\ x'_{ip} \end{bmatrix}, \qquad X_i^d = \begin{bmatrix} x'_{i1} - x'_{ip} \\ \vdots \\ x'_{i,p-1} - x'_{ip} \end{bmatrix}, \qquad \varepsilon_{ij} = \upsilon_{ij} - \upsilon_{ip},$$

$$\text{e.g.,} \quad X_i^d = \begin{bmatrix} & \text{price}_{i1} - \text{price}_{ip} \\ I_{p-1}, & \vdots \\ & \text{price}_{i,p-1} - \text{price}_{ip} \end{bmatrix}$$

Note: if X contains intercepts, we have “set” one to zero.

Identification Problems in Differenced System

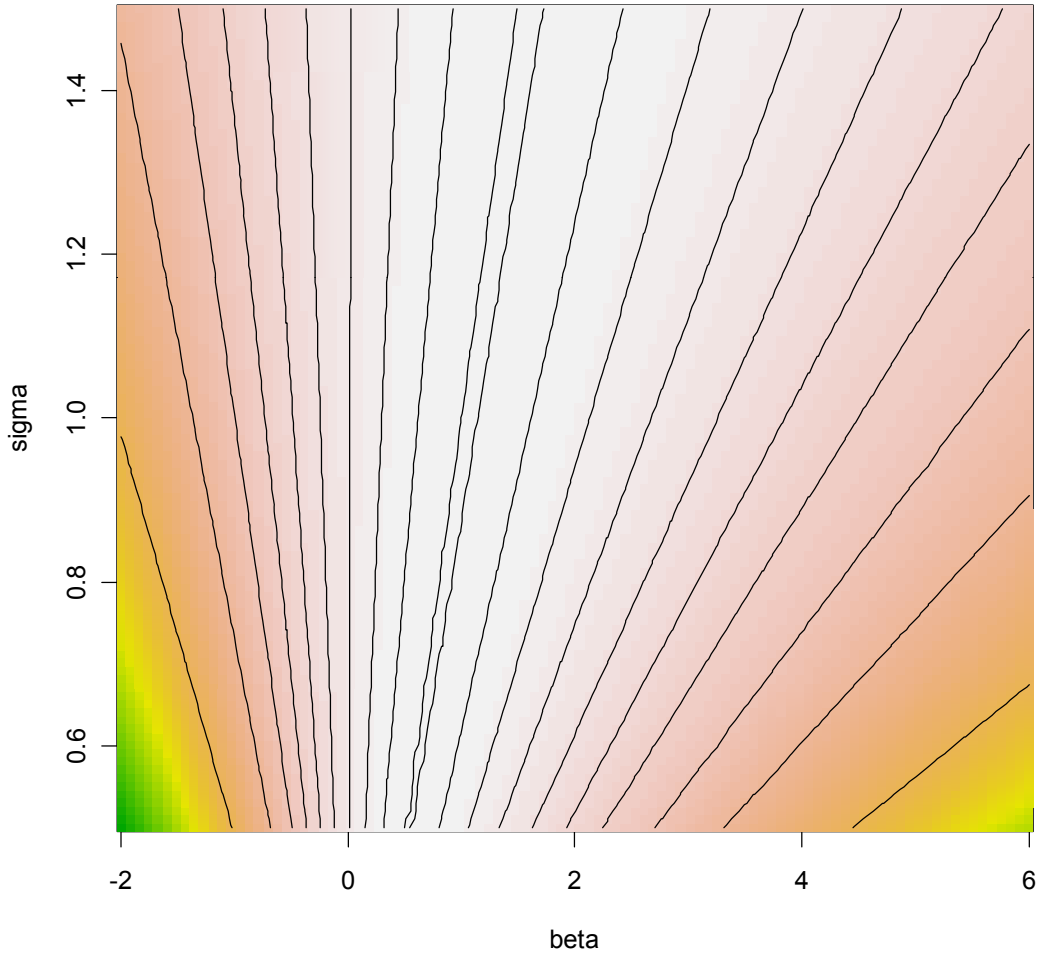
Given that the differenced system has a full covariance matrix, I can still multiply by a positive constant and leave y unchanged!

Thus, the identified parameters are given by:

$$\tilde{\beta} = \beta / \sqrt{\sigma_{11}}; \tilde{\Sigma} = \Sigma / \sigma_{11}$$

This implies that the likelihood function is constant over any direction in which $\tilde{\beta} = \beta / \sqrt{\sigma_{11}}$ is constant

Identification in Probit



Likelihood for Binary Probit Example

McCulloch and Rossi ’94 Approach

$$\begin{array}{ccccc} \Sigma & \searrow & & & \\ & & w & \rightarrow & y \\ \beta & \nearrow & & & \end{array}$$

GS :

$$w \mid \beta, \Sigma, y, X_i^d$$

$$\beta \mid \Sigma, w$$

$$\Sigma \mid \beta, w$$

Two “problems” to solve:

1. identification
2. draw of w | rest

M&R "Solution" to identification

Put a prior on the full, unidentified parameter space – induces a prior over the identified parms.

Gibbs in the unidentified space and "margin" down on the identified parms.

$$\beta \sim N\left(\overline{\beta}, A^{-1}\right) \quad \Sigma \sim IW\left(\upsilon, V_0\right)$$

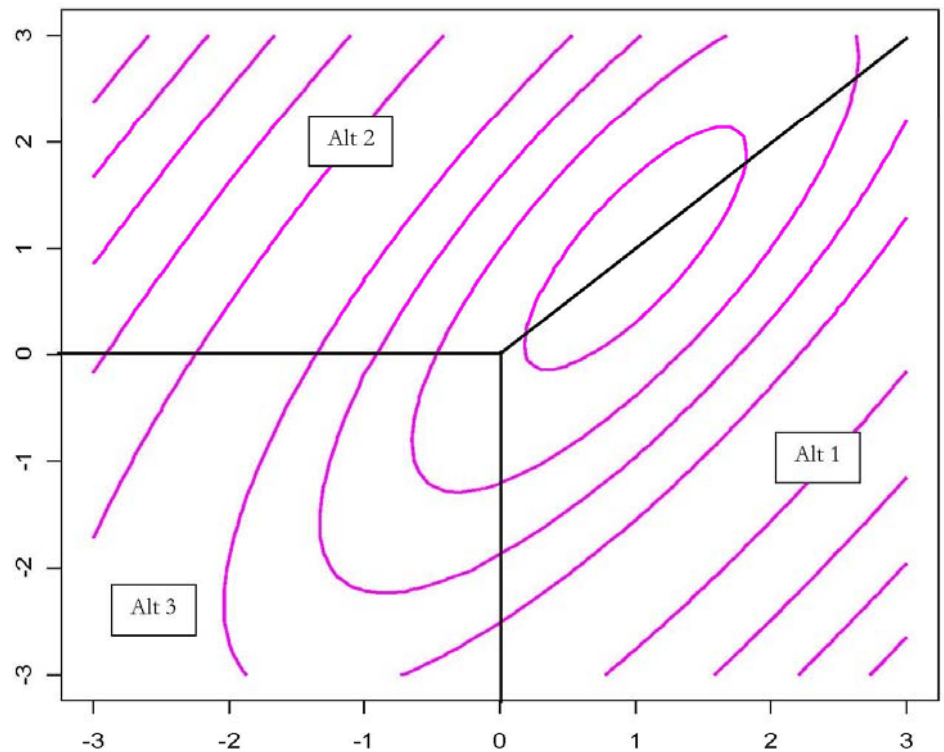
$$\tilde{\beta}^r = \beta^r / \sqrt{\sigma_{11}^r}; \tilde{\Sigma}^r = \Sigma^r / \sigma_{11}^r$$

"cost" – check prior. Works fines for relatively diffuse priors.

Likelihood for mnp model

Latents avoid evaluation of likelihood – integrals of MVN density over cones!

$$\Pr\left(y_i \mid X_i^d, \beta, \Sigma\right) = \int_{R_{y_i}} \varphi\left(w \mid X_i^d, \beta, \Sigma\right) dw$$



Conditional normal distribution

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

$$\Sigma^{-1} = V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$$

$$x_1 \mid x_2 \sim N\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\left(x_2 - \mu_2\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \right)$$

$$x_1 \middle| x_2 \sim N\left(\mu_1 - V_{11}^{-1}V_{12}\left(x_2 - \mu_2\right), V_{11}^{-1} \right)$$

Drawing w- "Gibbs thru"

$$w_{ij} \mid w_{i,-j}, y_i, \beta, \Sigma \sim N\left(m_{ij}, \tau_{jj}^2\right) \times I_{\text{truncation pts.}}$$

$$I_{\text{truncation pts.}} = \begin{cases} I\left(w_{ij} > \max\left(w_{i,-j}, 0\right)\right) & \text{if } j = y_i \\ I\left(w_{ij} < \max\left(w_{i,-j}, 0\right)\right) & \text{if } j \neq y_i \end{cases}$$

$$m_{ij} = x_{ij}'^{d}\beta + F'\left(w_{i,-j} - X_{i,-j}^{d}\beta\right), \quad F = -\sigma^{jj}\gamma_{j,-j}, \quad \tau_{jj}^2 = 1/\sigma^{jj}$$

where σ^{ij} denotes the (i,j)th element of Σ^{-1} and $\Sigma^{-1} = \begin{bmatrix} \gamma_1' \\ \vdots \\ \gamma_{p-1}' \end{bmatrix}$

createX

Usage:

```
createX(p, na, nd, Xa, Xd, INT = TRUE, DIFF =
FALSE, base = p)
```

Arguments:

p: integer - number of choice alternatives
na: integer - number of alternative-specific vars in Xa
nd: integer - number of non-alternative specific vars
Xa: n x p*na matrix of alternative-specific vars
Xd: n x nd matrix of non-alternative specific vars
INT: logical flag for inclusion of intercepts
DIFF: logical flag for differencing wrt to base alternative
base: integer - index of base choice alternative
note: na,nd,Xa,Xd can be NULL to indicate lack of Xa or Xd variables.

Value:
X matrix - n*(p-DIFF) x [(INT+nd)*(p-1) + na] matrix.

Examples:

```
na=2; nd=1; p=3
vec=c(1,1.5,.5,2,3,1,3,4.5,1.5)
Xa=matrix(vec,byrow=TRUE,ncol=3)
Xa=cbind(Xa,-Xa)
Xd=matrix(c(-1,-2,-3),ncol=1)
createX(p=p,na=na,nd=nd,Xa=Xa,Xd=Xd)
createX(p=p,na=na,nd=nd,Xa=Xa,Xd=Xd,base=1)
createX(p=p,na=na,nd=nd,Xa=Xa,Xd=Xd,DIFF=TRUE)
createX(p=p,na=na,nd=NULL,Xa=Xa,Xd=NULL)
createX(p=p,na=NULL,nd=nd,Xa=NULL,Xd=Xd)
```

Estimating the differenced system

Model: $[y|w]\ [w|X,\beta,\Sigma]\ [\beta]\ [\Sigma]$

Draw:	$w\|y,\beta,\Sigma$	(truncated normals)
	$\beta\|w,\Sigma$	(Bayes regression after standardization)
	$\Sigma\|w,\beta$	(Inverted Wishart)

Implemented in **`rmnpGibbs`** in **`bayesm`**

Example

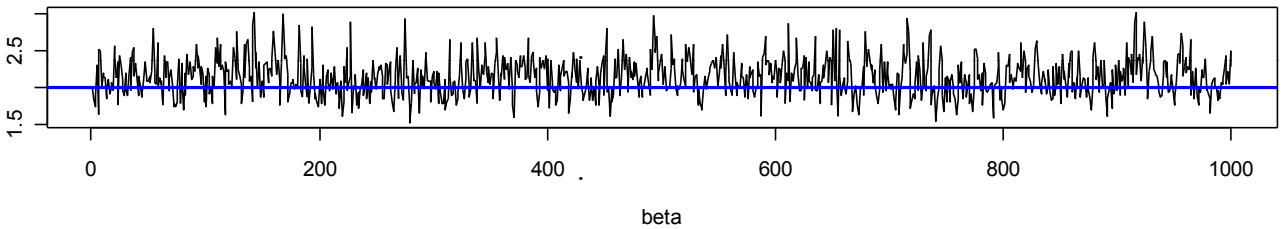
N=1600, p=6. X~iidUnif(-2,2).

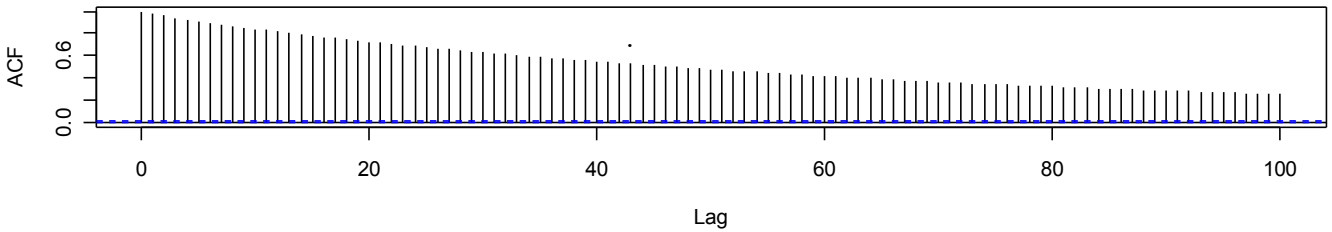
$\rho = .5$

$\beta = 2$

and

$$\Sigma = \text{diag}(\sigma)\left(\rho\iota\iota' + (1-\rho)I_{p-1}\right)\text{diag}(\sigma) \qquad \sigma' = (1,2,3,4,5)^{.5}$$





Hard Ex and f = 110

Multivariate probit model (`rmvpGibbs`)

$$y_{ij} = \begin{cases} 1, & \text{if } w_{ij} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$w_i = X_i\beta + \varepsilon_i \quad , \quad \varepsilon_i \sim N(0,\Sigma)$$

Select m of n brands; multiperiod, multicategory situations

$$(\beta,\Sigma) \rightarrow (\tilde{\beta},R) \text{ where } \tilde{\beta}_j = \beta_j / \sqrt{\sigma_{jj}}$$

$$\text{and } R = \Lambda\Sigma\Lambda \text{ and } \Lambda = \begin{bmatrix} 1/\sqrt{\sigma_{11}} & & \\ & \ddots & \\ & & 1/\sqrt{\sigma_{pp}} \end{bmatrix}$$

Metropolis algorithms, logit model estimation

Markov Chain Monte Carlo

Goal:

construct a Markov Chain whose invariant distribution is the posterior.

Implementation:

Start the chain from a point in the parameter space

Simulate “forward” until the initial conditions have worn off

Use the draws from the chain to estimate any posterior quantity of interest, appealing to ergodicity.

We are using asymptotics but sample sizes can be huge and under our control – more like the inventors of asymptotics had in mind

Review of Markov Chains

Discrete time, space

Put probability distribution on

$\{x_n\}$ n=1,2,3,…

x_n=i: Process is in state i at time n

$$p(x_{n+1} = j \mid x_n = i, x_{n-1} = i_{n-1}, \ldots, x_0 = i_0)$$

$$= p(x_{n+1} = j \mid x_n = i)$$

$$= p_{ij}$$

$$\text{require}: \sum_j p_{ij} = 1$$

MC

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kk} \end{bmatrix}$$

Each row gives the conditional distribution of the next x.
The column corresponds to the values of the current x

MC

If $p(x_0 = i) = \pi_{0i}$

then $x_1 \sim \pi_0 P$ where π_0 is a row vector

$$\text{Prob}[x_1=j] = p(x_0=1)p_{1j} + p(x_0=2)p_{2j} + \ldots$$

$$x_1 \sim \pi_0 P$$
$$x_n \sim \pi_0 P^n$$

MC

Assume all states communicate

i.e., can eventually get from i to j.

(aperiodic, irreducible)

Then we have a unique stationary distribution

$$\pi_0 P \rightarrow \pi$$

$$\pi P = \pi$$

Stationary distribution: Ex

$$\text{Let } P = \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} \qquad \pi = \begin{bmatrix} 1/3 & 2/3 \end{bmatrix}$$

$$\pi P = \begin{bmatrix} 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix} = \begin{bmatrix} 1/3 & 2/3 \end{bmatrix}$$

Time reversible chain

If we “reverse” time order, and ask what are the properties of the chain in reverse order, we find:

The “reversed” chain is a Markov chain.

The transition probabilities of the “reversed” chain are given by

$$p_{ij}^{*} = \frac{\pi_j p_{ji}}{\pi_i}$$

Time Reversible Chains

A Markov chain is time reversible if $p_{ij}^{*} = p_{ij}$

$$p_{ij} = \frac{\pi_j p_{ji}}{\pi_i} \text{ or } \pi_i p_{ij} = \pi_j p_{ji}$$

The chance of seeing a $i \to j$ transition

is the same as seeing a $j \to i$ transition

Some say the chain is “reversible” wrt π

Stationary and Time Reversibility

Suppose we have a time reversible chain:

$$\omega_i > 0 \quad \text{such that} \sum_i \omega_i = 1$$

$$\text{and } \omega_i p_{ij} = \omega_j p_{ji}$$

then:

$$\sum_i \omega_i p_{ij} = \omega_j \sum_i p_{ji} = \omega_j$$

$$\Rightarrow \omega P = \omega$$

$$\Rightarrow \omega = \pi \qquad (\omega \text{ is the stationary dist.})$$

Example

$$P = \begin{bmatrix} 1/4 & 3/4 \\ 3/8 & 5/8 \end{bmatrix} \qquad \omega = \begin{bmatrix} 1/3 & 2/3 \end{bmatrix}$$

$$\left.\begin{aligned} \omega_1 p_{12} &= (1/3)(3/4) = 3/12 \\ \omega_2 p_{21} &= (2/3)(3/8) = 6/24 = 3/12 \end{aligned}\right\} \text{ Time reversible}$$

check: does $\omega P = \omega$? If so, then $\omega = \pi$

Metropolis-Hastings algorithm

Construct a MC whose stationary distribution is π (the posterior distribution).

Know: π_i/π_j for any i, j

Let: $Q=\{q_{ij}\}$ be a proposed transition matrix

Define a new MC based on $Q=\{q_{ij}\}$ and π_i/π_j as follows:

Metropolis-Hastings algorithm

Start with a Markov Chain with transition probs given q. Modify this chain to get the correct stationary dist.

$$\text{Compute } \alpha(i,j) = \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\}$$

with probability α go to j,

with probability 1- α stay at i (repeat i).

then $p_{ij} = q_{ij}\, \alpha(i,j)$ yields a time reversible Markov chain

Proof

$$p_{ij} = q_{ij}\,\alpha(i,j)$$

generating candidate j given i

accept with some probability

$$\pi_i p_{ij} = \pi_i q_{ij} \min\left\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right\}$$

$$= \min\{\pi_i q_{ij}, \pi_j q_{ji}\}$$

$$\pi_j p_{ji} = \min\{\pi_j q_{ji}, \pi_i q_{ij}\}$$

P is reversible with stationary distribution π: $\pi_i p_{ij} = \pi_j p_{ji}$

Metropolis-Hastings algorithm example

$$\pi = \begin{bmatrix} 1/3 & 2/3 \end{bmatrix} \qquad q_{ij} = 1/2 \qquad Q = \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix}$$

$$p_{12} = .5 \min\left\{1, \frac{2/3}{1/3}\right\} = .5(1) = .5$$

$$p_{21} = .5 \min\left\{1, \frac{1/3}{2/3}\right\} = .5(.5) = .25 \quad P = \begin{bmatrix} 1/2 & 1/2 \\ 1/4 & 3/4 \end{bmatrix}$$

check: does $\pi_1 p_{12} = \pi_2 p_{21}$? does $\pi P = \pi$? (yes!)

can construct a MC whose stationary dist. is π knowing only π_i/π_j for any i and j.

Continuous Metropolis-Hastings

discrete: $i \to j$
continuous: $\theta \to \vartheta$

Q is a Markov chain. Given θ, $q(\theta, \vartheta)$ is the conditional density of the "next one." π is the desired stationary distribution.

1. Generate $\vartheta \sim q(\theta, \vartheta)$

2. $\alpha(\theta, \vartheta) = \min\left\{1, \dfrac{\pi(\vartheta)q(\vartheta, \theta)}{\pi(\theta)q(\theta, \vartheta)}\right\}$

3. With prob α, move to ϑ, else stay at θ

Independence chain

$$\text{Let } q(\theta, \vartheta) = q_{imp}(\theta)$$

$$\text{Then } \alpha(\theta, \vartheta) = \min\left\{1, \frac{\pi(\vartheta)\, q_{imp}(\theta)}{\pi(\theta)\, q_{imp}(\vartheta)}\right\}$$

$$= \min\left\{1, \frac{\pi(\vartheta) / q_{imp}(\vartheta)}{\pi(\theta) / q_{imp}(\theta)}\right\}$$

$q_{imp}()$ should have fatter tails than π to avoid the need to reject draws to build up tail mass.

Random walk chains

At θ, draw $\varepsilon \sim q$ independent of x.

$$\vartheta = \theta + \varepsilon$$

$$q(\theta, \vartheta) = q(\vartheta, \theta) \quad \text{if } q \text{ is a symmetric dist}$$

Then $$\alpha(\theta, \vartheta) = \min\left\{1, \frac{\pi(\vartheta)q(\vartheta, \theta)}{\pi(\theta)q(\theta, \vartheta)}\right\}$$

$$= \min\left\{1, \frac{\pi(\vartheta)}{\pi(\theta)}\right\}$$

Indep Vs. RW Chains

Independence Chains:

- requires a good approximation to posterior
- (similar to Importance Sampling)
- implies some sort of optimizer
- more efficient than RW

RW Chains:

- will explore parameter space – no location required!
- for low dimensions will work even with “dumb” choices of increment Cov matrix
- may not work well in high dimensional spaces unless increment Cov closely approximates posterior

Choosing a step size for the RW chain

At θ, draw $\varepsilon \sim q_{imp}$ independent of θ.

candidate $= \theta + \varepsilon$

ε small leads to small steps, higher acceptance, higher autocorrelation.

ε large leads to large steps, lower acceptance, lower autocorrelation.

Pick $\varepsilon \sim N(0,s^2\Sigma)$, choosing s to maximize information content.

Choosing a step size for the RW chain

Choice of Σ:

I

Asymptotic Var-Cov for Posterior or Likelihood

Choice of scaling constant (s):

maximize information content (numerical efficiency) of draw sequence

$$\hat{f}_R = 1 + \sum_{j=1}^{m} \left(\frac{m+1-j}{m+1} \right) \hat{\rho}_j$$

get the "right" acceptance rate (30-50%)

R&R:

$$s = 2.3 \Big/ \sqrt{d = \dim(\text{state space})}$$

The Gibbs sampler

Draws from full conditional distribution: $\theta' = (\theta_j, \theta_{-j})$

$$q(\theta^{t-1}, \theta^t) = \left\{ \begin{array}{ll} p(\theta_j^t \mid \theta_{-j}^{t-1}, y) & \text{if } \theta_{-j}^t = \theta_{-j}^{t-1} \\ 0 & \text{otherwise} \end{array} \right\}$$

Only update θ_j

$$\alpha(\theta_j^{t-1}, \theta_j^t) = \min\left\{1, \frac{\pi(\theta^t \mid y)\, p(\theta_j^{t-1} \mid \theta_{-j}^{t-1}, y)}{\pi(\theta^{t-1} \mid y)\, p(\theta_j^t \mid \theta_{-j}^{t-1}, y)}\right\}$$

The Gibbs sampler

But:

$$p(\theta^t \mid y) = p(\theta_{-j}^{t-1}, \theta_j^t \mid y) = p(\theta_{-j}^{t-1} \mid y)\, p(\theta_j^t \mid \theta_{-j}^{t-1}, y)$$

$$p(\theta^{t-1} \mid y) = p(\theta_{-j}^{t-1}, \theta_j^{t-1} \mid y) = p(\theta_{-j}^{t-1} \mid y)\, p(\theta_j^{t-1} \mid \theta_{-j}^{t-1}, y)$$

So:

$$\alpha(\theta_j^{t-1}, \theta_j^t) = \frac{p(\theta_{-j}^{t-1} \mid y)}{p(\theta_{-j}^{t-1} \mid y)}$$

$$= 1$$

(always accept!)

Logit model

$$\text{Prior}: \; p(\beta) = \text{Normal}(\bar{\beta}, A^{-1})$$

Likelihood:

$$\ell\left(\beta \middle| X, y\right) = \prod_{i=1}^{n} \Pr\left(y_i = j \middle| X_i, \beta\right)$$

$$\Pr\left(y_i = j \middle| X_i, \beta\right) = \frac{\exp\left(x_{i,j}^{'}\beta\right)}{\sum_{j=1}^{J} \exp\left(x_{i,j}^{'}\beta\right)}$$

$$X_i = \begin{bmatrix} x_{i,1}^{'} \\ \vdots \\ x_{i,J}^{'} \end{bmatrix}$$

Logit model-Hessian

Both Indep and RW Metropolis chains rely on an asymptotic approximation to the posterior

$$\pi\left(\beta \middle| X, y\right) \dot{\propto} \left|H\right|^{1/2} \exp\left\{\tfrac{1}{2}\left(\beta - \hat{\beta}\right)' H\left(\beta - \hat{\beta}\right)\right\}$$

For the logit model, we will use the expected sample information matrix:

$$H = -E\left[\frac{\partial^2 \log \ell}{\partial\beta\partial\beta'}\right] = \sum_i X_i A_i X_i'$$

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}; \quad A_i = \mathrm{Diag}\left(p_i\right) - p_i p_i'$$

Logit model MCMC Algorithms

1. Pick an arbitrary starting value β^{old}

2. Generate candidate realization:

   random walk chain: $\beta^{cand} = \beta^{old} + \varepsilon; \quad \varepsilon \sim N\left(0, s^2 H^{-1}\right)$

   independence chain: $\beta^{cand} \sim MSt\left(\upsilon, \hat{\beta}, H^{-1}\right)$

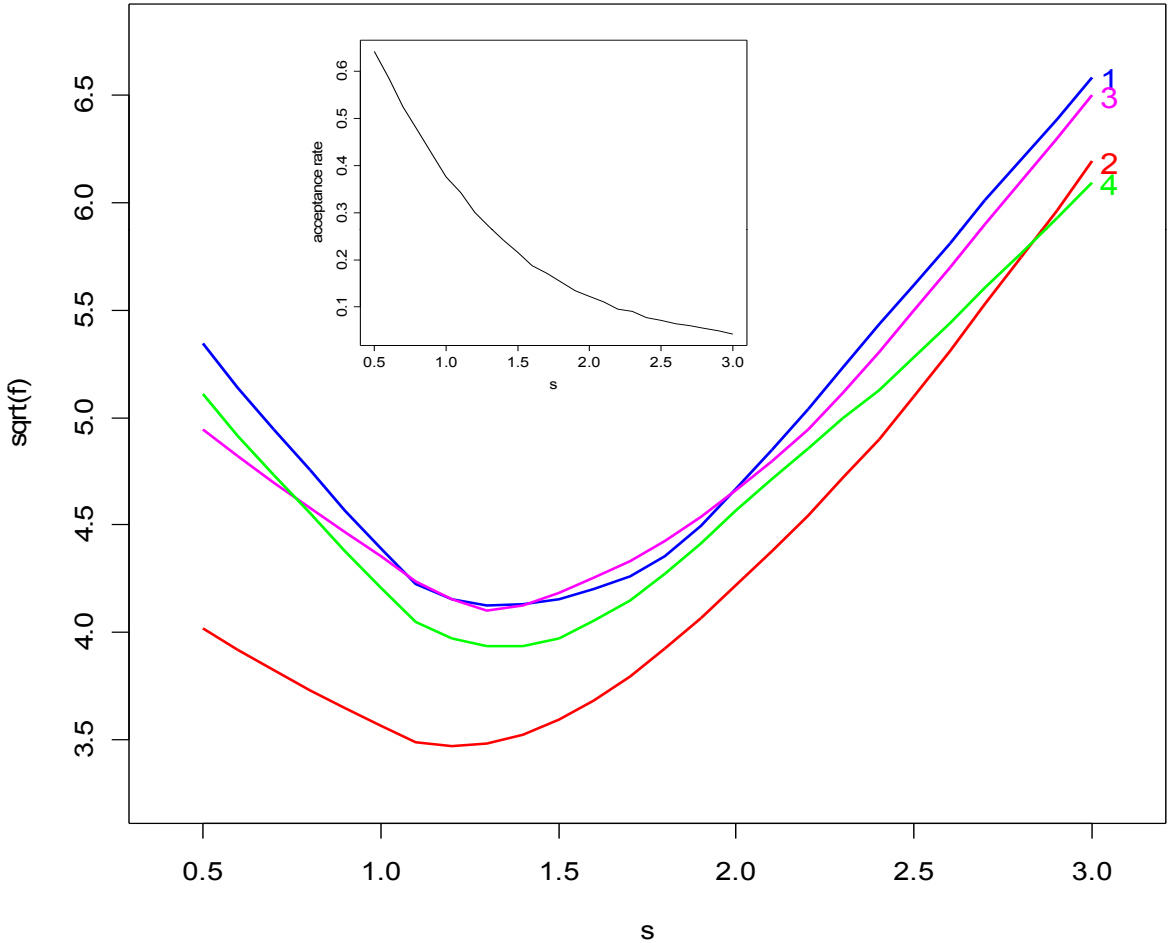
3. Accept β^{new} with probability α

$$\alpha = \min\left\{1, \frac{\ell\left(\beta^{new} \middle| y, X\right)\pi(\beta^{new})}{\ell\left(\beta^{old} \middle| y, X\right)\pi(\beta^{old})} \times \frac{q\left(\beta^{new}, \beta^{old}\right)}{q\left(\beta^{old}, \beta^{new}\right)}\right\}$$

4. Repeat

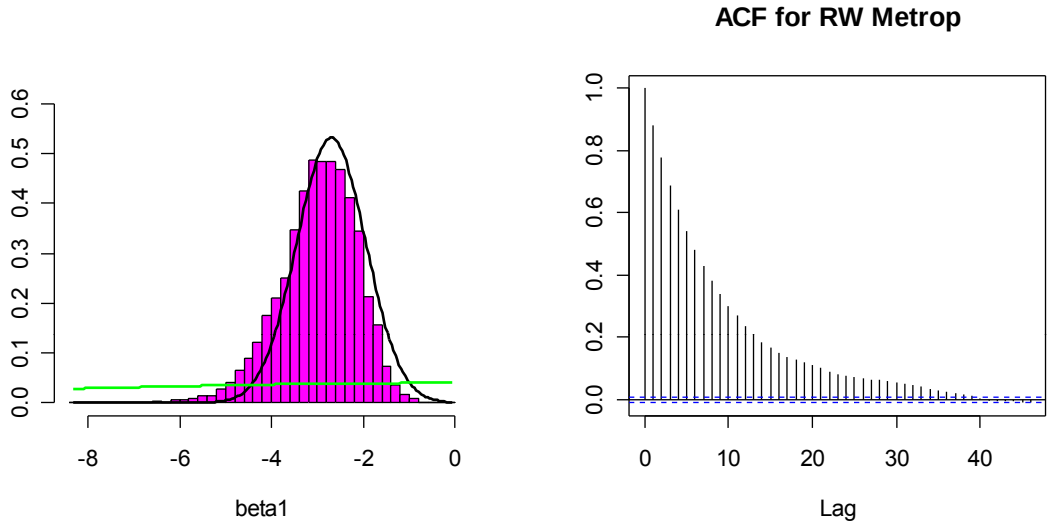
Scaling RW Metropolis

$\sqrt{f}$

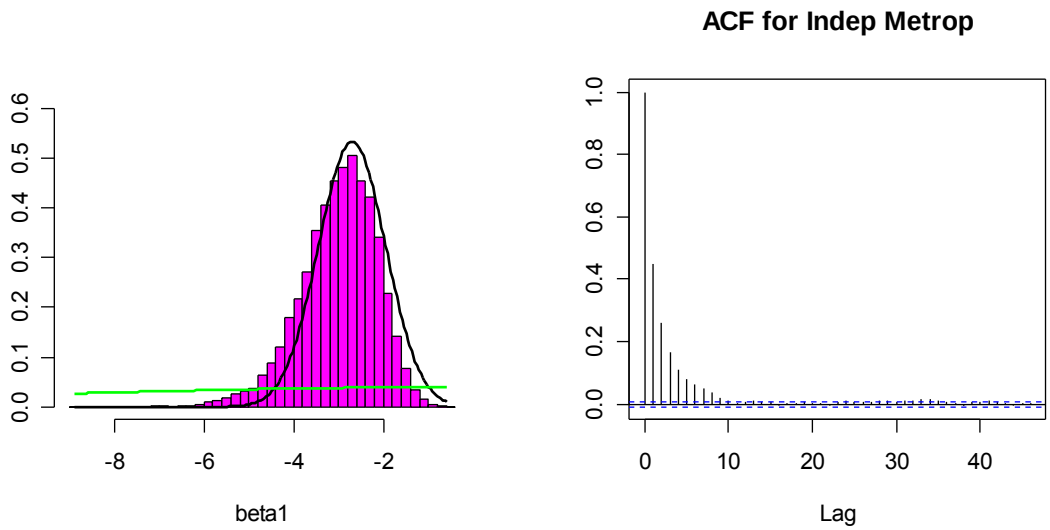


Comparison of Indep/RW Metropolis

f=16



f=2



rmnlIndepMetrop

```
rmnlIndepMetrop= function(Data,Prior,Mcmc)
{
#
# purpose:
#   draw from posterior for MNL using Independence Metropolis
#
# Arguments:
#   Data - list of m,X,y
#     m is number of alternatives
#     X is nobs*m x nvar matrix
#     y is nobs vector of values from 1 to m
#   Prior - list of A, betabar
#     A is nvar x nvar prior preci matrix
#     betabar is nvar x 1 prior mean
#   Mcmc
#     R is number of draws
#     keep is thinning parameter
#     nu degrees of freedom parameter for independence
#        sampling density
#
```

rmnIndepMetrop (continued)

```
# Output:
betadraw=matrix(double(floor(R/keep)*nvar),ncol=nvar)
#
# compute required quantities for indep candidates
#
beta=c(rep(0,nvar))
mle=optim(beta,llmnl,X=X,y=y,method="BFGS",hessian=TRUE,control=list(fnscale=-1))
beta=mle$par
betastar=mle$par
mhess=mnlhess(y,X,beta)
candcov=chol2inv(chol(mhess))
root=chol(candcov)
rooti=backsolve(root,diag(nvar))
priorcov=chol2inv(chol(A))
rootp=chol(priorcov)
rootpi=backsolve(rootp,diag(nvar))
```

rmnIndepMetrop (continued)

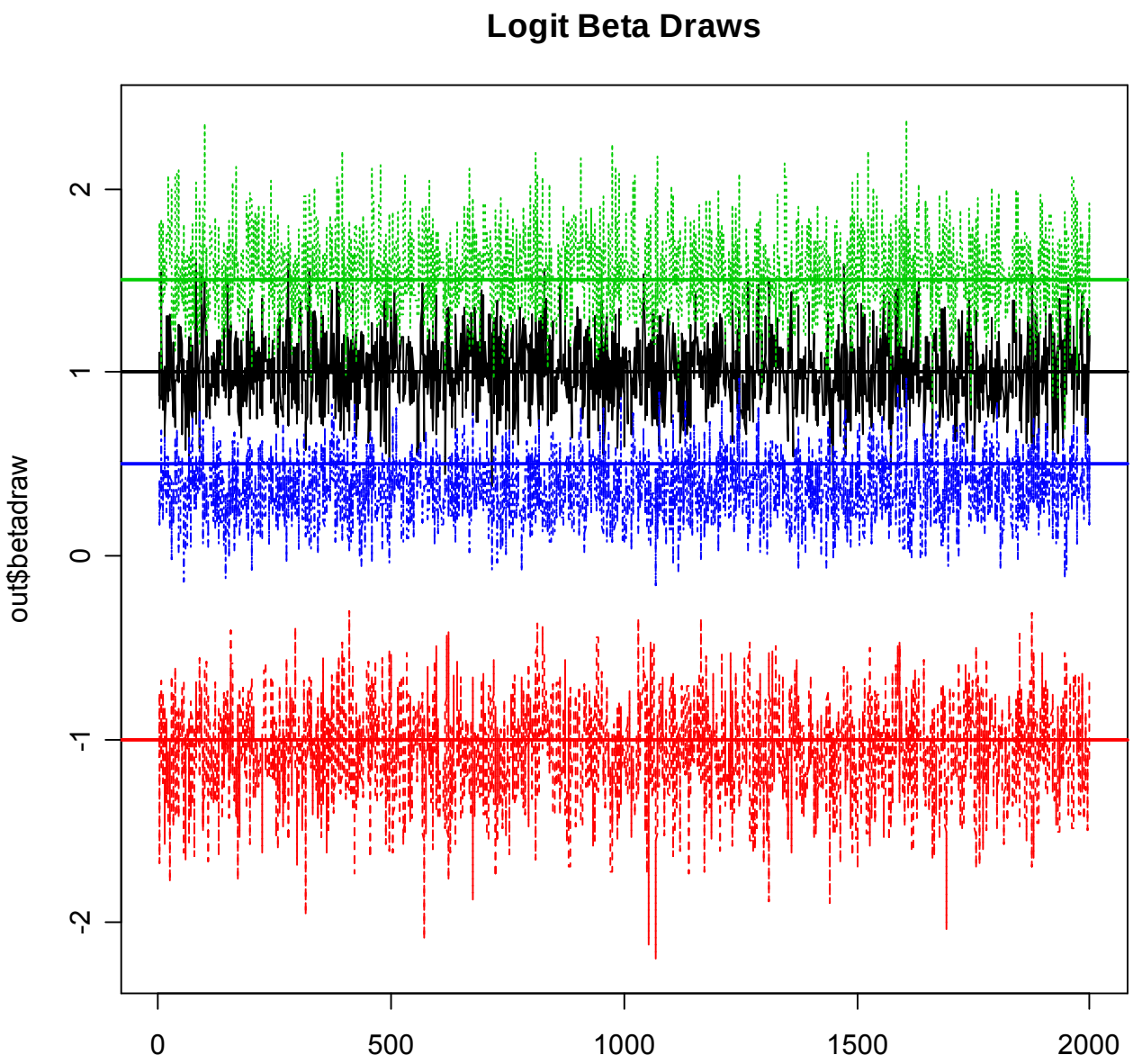
```
#
#    start main iteration loop
#
itime=proc.time()[3]
cat("MCMC Iteration (est time to end - min) ",fill=TRUE)
flush()

oldlpost=llmnl(y,X,beta)+lmvn(beta,betabar,rootpi)
oldlimp=lmvst(beta,nu,betastar,rooti)
#       note: we don't need the determinants as they cancel in
#       computation of acceptance prob
naccept=0
```

rmnIndepMetrop (continued)

```
for (rep in 1:R)
{
   betac=rmvst(nu,betastar,root)
   clpost=llmnl(y,X,betac)+lmvn(betac,betabar,rootpi)
   climp=lmvst(betac,nu,betastar,rooti)
   ldiff=clpost+oldlimp-oldlpost-climp
   alpha=min(1,exp(ldiff))
   if(alpha < 1) {unif=runif(1)} else {unif=0}
   if (unif <= alpha)
      { beta=betac
        oldlpost=clpost                    accept!
        oldlimp=climp
        naccept=naccept+1}

  if(rep%%keep == 0)
    {mkeep=rep/keep; betadraw[mkeep,]=beta}
}
list(betadraw=betadraw,acceptr=naccept/R)
}
```

rmnIndepMetrop (continued)

```
set.seed(66)
n=200; m=3; beta=c(1,-1,1.5,.5)
simout=simmnl(m,n,beta)
A=diag(c(rep(.01,length(beta)))); betabar=rep(0,length(beta))

R=2000
Data=list(y=simout$y,X=simout$X,m=m); Mcmc=list(R=R,keep=1) ; Prior=list(A=A,betabar=betabar)
out=rmnlIndepMetrop(Data=Data,Prior=Prior,Mcmc=Mcmc)
cat(" Betadraws ",fill=TRUE)
mat=apply(out$betadraw,2,quantile,probs=c(.01,.05,.5,.95,.99))
mat=rbind(beta,mat); rownames(mat)[1]="beta"; print(mat)
...
Betadraws
          [,1]       [,2]     [,3]        [,4]
beta 1.0000000 -1.0000000 1.500000  0.50000000
1%   0.5815957 -1.7043037 1.030927 -0.02822091
5%   0.6975339 -1.5333176 1.190716  0.07802924
50%  1.0020766 -1.0534945 1.533220  0.36624089
95%  1.3385280 -0.6466890 1.907956  0.67010696
99%  1.4804503 -0.4880313 2.077941  0.79372107
```


Logit Beta Draws
out$betadraw
2
1
0
-1
-2
0
500
1000
1500
2000

Heterogeneity

Panel Structures

Disaggregate data often comes with a panel structure:

- conjoint surveys with 10-20 questions and many respondents
- “key account” data
- a panel of consumers

Unit Likelihoods: $p\left(y_i \middle| \theta_i\right)$

Prior? It will matter!

Heterogeneity and priors

$$p(\theta_1,\ldots,\theta_m \mid y_1,\ldots,y_m) \propto \left[\prod_i p(y_i \mid \theta_i)\right] \times p(\theta_1,\ldots,\theta_m \mid \tau)$$

$$p(\theta_1,\ldots,\theta_m \mid \tau) = ?$$

$$p(\theta_1,\ldots,\theta_m \mid \tau) = \prod_i p(\theta_i \mid \tau)$$

Some call this a random effects model

$$p(\theta_1,\ldots,\theta_m,\tau \mid h) \propto \left[\prod_i p(\theta_i \mid \tau)\right] \times p(\tau \mid h)$$

Multistage Prior/ Multi-level Model:
$[y_i|\theta_i]\ [\theta_i|\tau]\ [\tau|h]$

$$\tau \rightarrow \{\theta_i\} \rightarrow y_i$$

Heterogeneity and priors

$$p(\theta_1,\ldots,\theta_m,\tau \mid h) \propto \left[\prod_i p(\theta_i \mid \tau)\right] \times p(\tau \mid h)$$

Induces a highly dependent prior on the collect of unit-level parameters, esp. if "top" prior is diffuse

$$p(\theta_1,\ldots,\theta_m \mid h) = \int \prod_i p(\theta_i \mid \tau)\, p(\tau \mid h)\, d\tau$$

τ is the common component!

Marginalizing the likelihood

In a Bayesian analysis, we do <u>not</u> "marginalize" the likelihood:

$$\ell(\tau) = \prod_i \int p(y_i \mid \theta_i)\, p(\theta_i \mid \tau)\, d\theta_i$$

Instead, we derive the joint distribution of all model parameters.

$$p(\theta_1, \ldots, \theta_m, \tau \mid y_1, \ldots, y_m, h) \propto \left[\prod_i p(y_i \mid \theta_i)\, p(\theta_i \mid \tau) \right] \times p(\tau \mid h)$$

Hierarchical Linear Model

Consider m regressions:

$$y_i = X_i\beta_i + \varepsilon_i \qquad \varepsilon_i \sim \text{iidN}\left(0, \sigma_i^2 I_{n_i}\right) \quad i = 1, \ldots, m$$

$$\beta_i = \Delta' z_i + v_i \quad v_i \sim \text{iidN}\left(0, V_\beta\right)$$ Tie together via Prior

or

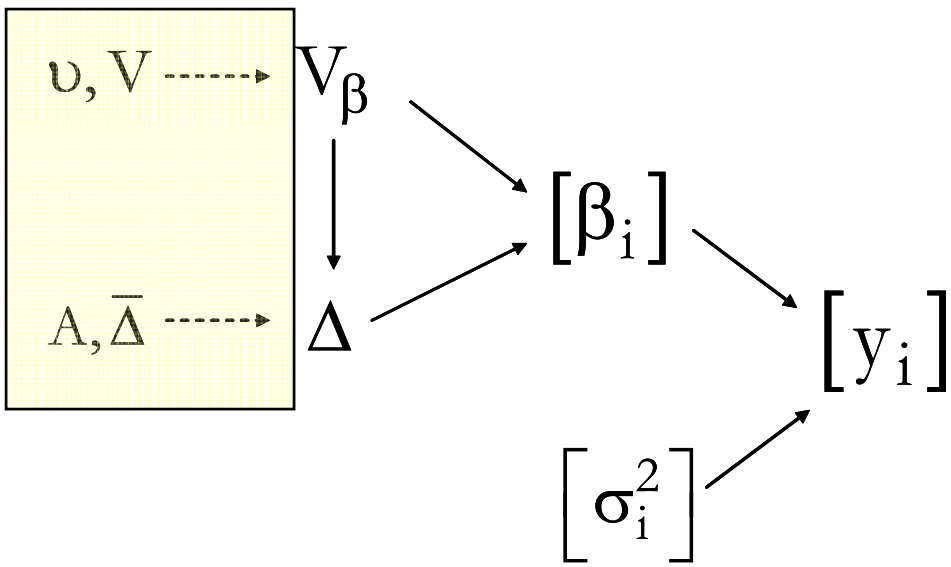
$$B = Z\Delta + V \quad B = \begin{bmatrix} \beta_1' \\ \vdots \\ \beta_m' \end{bmatrix} \quad Z = \begin{bmatrix} z_1' \\ \vdots \\ z_m' \end{bmatrix} \quad \Delta = \begin{bmatrix} \delta_1 & \cdots & \delta_k \end{bmatrix} \quad v_i' \sim N\left(0, V_\beta\right)$$

Priors

$$V_\beta \sim IW(\nu, V)$$

$$\mathrm{vec}(\Delta) \,|\, V_\beta \sim N\left(\mathrm{vec}(\bar{\Delta}), V_\beta \otimes A^{-1}\right)$$

$$\sigma_i^2 \sim \nu_i s_{0,i}^2 \Big/ \chi_{\nu_i}^2$$



GS for the Hierarchical Linear Model

$$\beta_i \left| \sigma_i^2, \Delta, V_\beta, y_i, X_i \right.$$

Univariate Regression with an *informative* prior!

$$\sigma_i^2 \left| \beta_i, y_i, X_i \right.$$

note independence from hierarchical parms!

$$\Delta, V_\beta \left| \{\beta_i\}, Z \right.$$

rmultireg with $\{\beta_i\}$ as data and Z as "X"

implemented in rhierLinearModel

Adaptive Shrinkage

With fixed values of Δ, V_β, we have m independent Bayes regressions with informative priors.

In the hierarchical setting, we “learn” about the location and spread of the $\{\beta_i\}$.

The extent of shrinkage, for any one unit, depends on dispersion of betas across units and the amount of information available for that unit.

An Example – Key Account Data

y= log of sales of a “sliced cheese” product at a “key” account – market retailer combination

X:

log(price)

display (dummy if on display in the store)

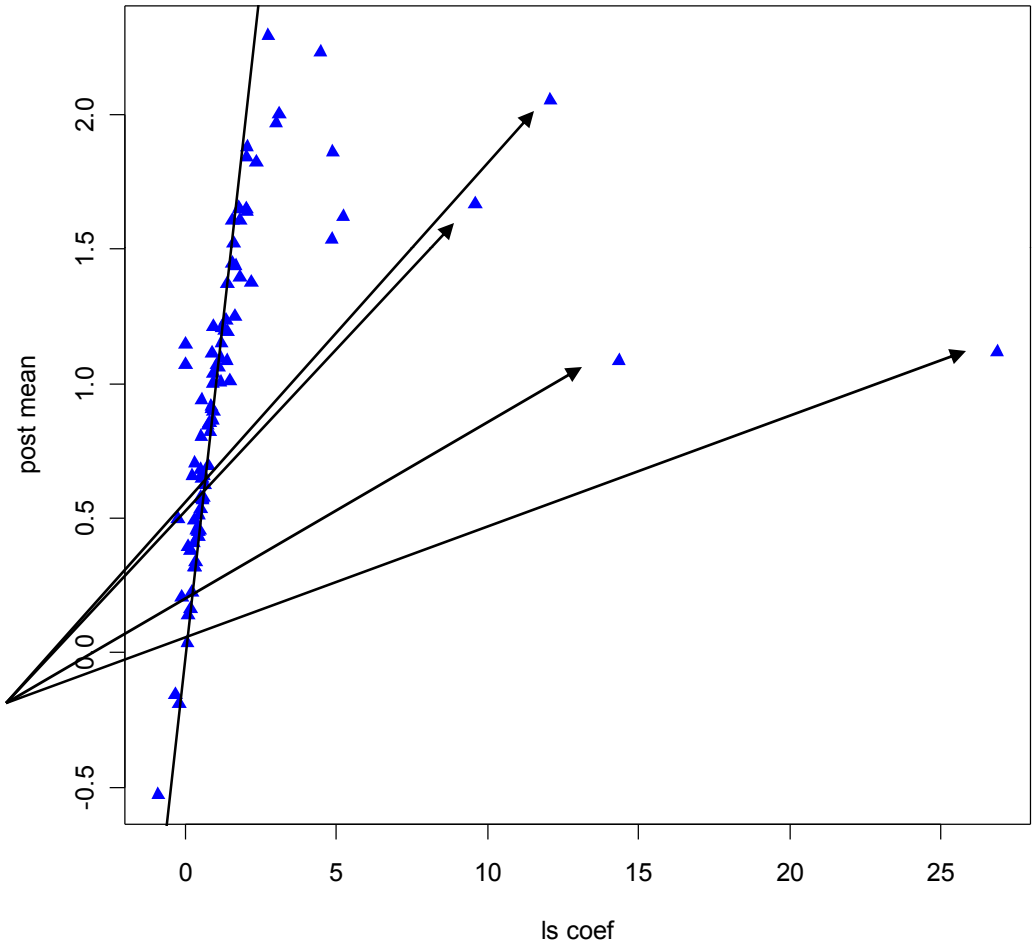
weekly data on 88 accounts. Average account has 65 weeks of data.

See data(cheese)

An Example – Key Account Data

Failure of Least Squares

- some accounts have no displays!
- some accounts have absurd coefs



Shrinkage

Prior on V_β is key.

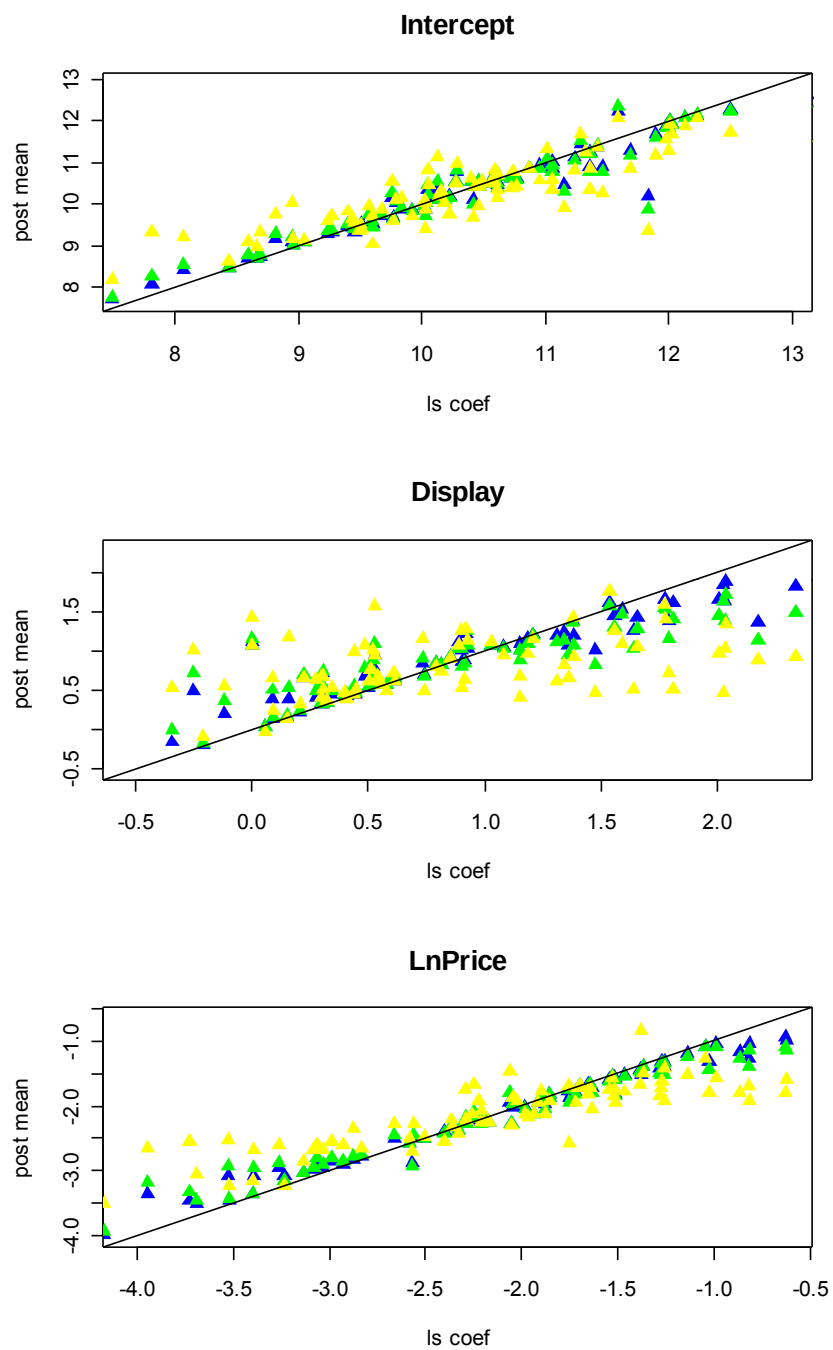
$$V_\beta \sim IW(\upsilon, .1I)$$

blue : $\upsilon = k + 3$

green : $\upsilon = k + .5n$

yellow : $\upsilon = k + 2n$

Greatest Shrinkage for Display, least for intercepts



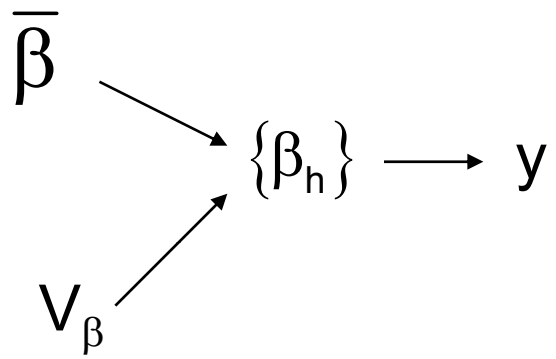
Heterogeneous logit model

Priors:

$$\beta_h \sim N(\bar{\beta}, V_\beta)$$

$$\bar{\beta} \sim N(\bar{\bar{\beta}}, A)$$

$$V_\beta \sim IW(\upsilon, \upsilon I)$$



GS:

$$\beta_h \,|\, \bar{\beta}, V_\beta$$

$$\bar{\beta}, V_\beta \,|\, \{\beta_h\}$$

Heterogeneous logit model

Assume T_h observations per respondent

$$\Pr(y_{it})_h = \frac{\exp[x_{it}'\beta_h]}{\sum_j \exp[x_{jt}'\beta_h]}$$

The posterior:

$$p(\{\beta_h\},\bar{\beta},V_\beta \mid \text{Data}) \propto \prod_{h=1}^{H}\left(\prod_{t=1}^{T_h}[y_{iht} \mid X_{ht},\beta_h]\right)[\beta_h \mid \bar{\beta},V_\beta]\,[\bar{\beta}][V_\beta]$$

logit model

normal heterogeneity

priors

Drawing β_h

Use RW Metropolis:

$$\beta_h^{new} = \beta_h^{old} + \varepsilon,\ \varepsilon \sim N(0, ?)$$

V_β

Increment Cov matrix: One simple idea is just to use the prior – assumes unit likelihoods are relatively uninformative

$$\alpha(\beta_h^{new}, \beta_h^{old}) = \min\left\{1, \frac{\pi(\beta_h^{new})}{\pi(\beta_h^{old})}\right\}$$

$$\pi(\beta^{new}) = \left(\prod_{t=1}^{T_h} \frac{\exp[x_{iht}'\beta^{new}]}{\sum_j \exp[x_{jht}'\beta^{new}]}\right) \times \exp\left(\frac{-1}{2}(\beta^{new} - \bar{\beta})' V_\beta^{-1} (\beta^{new} - \bar{\beta})\right)$$

Random effects with regressors

$\beta_h = \Delta' z_h + u_i$ or $B = Z\Delta + U$ $U \sim \text{Normal}(0, V_\beta)$

Δ is a matrix of regression coefficients related covariates (Z) to mean of random-effects distribution.

z_h are covariates for respondent h

Heterogeneous logit with R

```
 rhierBinLogit=
function(Data,Prior,Mcmc){
# Arguments:
#   Data contains a list of (Dat[[i]],Demo)
#      Dat[[i]]=list(y,X)
#         y is index of brand chosen, y=1 is exp[X'beta]/(1+exp[X'beta])
#         X is a matrix that is n_i x by nxvar
# Demo is a matrix of demographic variables nhh*ndvar that
#            have been mean centered so that the intercept is
#            interpretable
#   Prior contains a list of (nu,V0,deltabar,Adelta)
#      beta_i ~ N(delta,Vbeta)
#      delta ~ N(deltabar,Adelta^-1)
#      Vbeta ~ IW(nu,V0)
#   Mcmc is a list of (sbeta,R,keep)
#      sbeta is scale factor for RW increment for beta_is
#      R is number of draws
#      keep every keepth draw
#
# Output:
#      a list of deltadraw (R/keep x nxvar x ndvar),
#         Vbetadraw (R/keep x nxvar**2),
#         llike (R/keep),  betadraw is a nunits x nxvar x
#          ndvar x R/keep array of draws of betas
#         nunits=length(Data)
```

Heterogeneous logit with R (cont.)

```
loglike=
function(y,X,beta) {
# function computer log likelihood of data for bin. logit model
# Pr(y=1) = 1 - Pr(y=0) = exp[X'beta]/(1+exp[X'beta])
prob = exp(X%*%beta)/(1+exp(X%*%beta))
prob = prob*y + (1-prob)*(1-y)
sum(log(prob))
}
# extract needed information
#
Demo=as.matrix(Data$Demo)
Data=Data$Dat
nhh=length(Data)
nxvar=ncol(Data[[1]]$X)
ndvar=ncol(Demo)
deltabar=Prior$deltabar
Adelta=Prior$Adelta
V0=Prior$V0
nu=Prior$nu
R=Mcmc$R
keep=Mcmc$keep
sbeta=Mcmc$sbeta
```

Heterogeneous logit (cont.)

```
#
# initialize storage for draws
#
Vbetadraw=matrix(double(floor(R/keep)*nxvar*nxvar),ncol=nxvar*nxvar)
betadraw=array(double(floor(R/keep)*nhh*nxvar),dim=c(nhh,nxvar,floor(R/keep)))
deltadraw=matrix(double(floor(R/keep)*nxvar*ndvar),ncol=nxvar*ndvar)
oldbetas=matrix(double(nhh*nxvar),ncol=nxvar)
oldVbeta=diag(rep(1,nxvar))
oldVbetai=diag(rep(1,nxvar))
olddelta=matrix(double(nxvar*ndvar),ncol=nxvar)

betad = array(0,dim=c(nxvar))
betan = array(0,dim=c(nxvar))
reject = array(0,dim=c(R/keep))
llike=array(0,dim=c(R/keep))
```

Heterogeneous logit (cont.)

```
#
# set up fixed parm for the draw of Vbeta, Delta=delta
#
Fparm=init.rmultiregfp(Demo,Adelta,deltabar,nu,V0)

itime=proc.time()[3]
cat("MCMC Iteration (est time to end - min)",fill=TRUE)
flush.console()

for (j in 1:R) {
      rej = 0
      logl = 0
      sV = sbeta*oldVbeta
      root=t(chol(sV))
```

Heterogeneous logit (cont.)

```
# Draw B-h|B-bar, V
 for (i in 1:nhh) {
   betad = oldbetas[i,]
   betan = betad + root%*%rnorm(nxvar)
# data
   lognew = loglike(Data[[i]]$y,Data[[i]]$X,betan)
   logold = loglike(Data[[i]]$y,Data[[i]]$X,betad)
# heterogeneity
   logknew = -.5*(t(betan)-Demo[i,]%*%olddelta) %*% oldVbetai
                %*% (betan-t(Demo[i,]%*%olddelta))

   logkold = -.5*(t(betad)-Demo[i,]%*%olddelta) %*% oldVbetai
                 %*% (betad-t(Demo[i,]%*%olddelta))
# MH step
alpha = exp(lognew + logknew - logold - logkold)
      if(alpha=="NaN") alpha=-1
      u = runif(n=1,min=0, max=1)
      if(u < alpha) {
              oldbetas[i,] = betan
              logl = logl + lognew } else {
              logl = logl + logold
              rej = rej+1  }
      }
```

candidate beta

Heterogeneous logit (cont.)

```
# Draw B-bar and V as a multivariate regression
    out=rmultiregfp(oldbetas,Demo,Fparm)
    olddelta=out$B
    oldVbeta=out$Sigma
    oldVbetai=chol2inv(chol(oldVbeta))
```

Heterogeneous logit (cont.)

```
    if((j%%1000)==0)
      {
       ctime=proc.time()[3]
       timetoend=((ctime-itime)/j)*(R-j)
       cat(" ",j," (",round(timetoend/60,1),")",fill=TRUE)
       flush.console() }
    mkeep=j/keep
    if(mkeep*keep == (floor(mkeep)*keep))
      {deltadraw[mkeep,]=as.vector(olddelta)
       Vbetadraw[mkeep,]=as.vector(oldVbeta)
       betadraw[,,mkeep]=oldbetas
       llike[mkeep]=logl
       reject[mkeep]=rej/nhh
      }
}
ctime=proc.time()[3]
cat(" Total Time Elapsed: ",round((ctime-itime)/60,2),fill=TRUE)

list(betadraw=betadraw,Vbetadraw=Vbetadraw,deltadraw=deltadraw,llike=llike,reject=reject)
}
```

Running rhierBinLogit

```
z=read.table("bank.dat",header=TRUE)
d=read.table("bank demo.dat",header=TRUE)

# center demo data so that mean of random-effects
# distribution can be interpretted as the average respondents
d[,1]=rep(1,nrow(d))
d[,2]=d[,2]-mean(d[,2])
d[,3]=d[,3]-mean(d[,3])
d[,4]=d[,4]-mean(d[,4])
hh=levels(factor(z$id))
nhh=length(hh)

Dat=NULL

for (i in 1:nhh) {
      y=z[z[,1]==hh[i],2]
      nobs=length(y)
      X=as.matrix(z[z[,1]==hh[i],c(3:16)])
      Dat[[i]]=list(y=y,X=X)
                }
Data=list(Dat=Dat,Demo=d)
```

Running rhierBinLogit (continued)

```
cat("Finished Reading data",fill=TRUE)
flush.console()

nxvar=14
ndvar=4
nu=nxvar+5
Prior=list(nu=nu,V0=nu*diag(rep(1,nxvar)),
     deltabar=matrix(rep(0,nxvar*ndvar),
      ncol=nxvar),
     Adelta=.01*diag(rep(1,ndvar)))

Mcmc=list(R=20000,sbeta=0.2,keep=20)

out=rhierBinLogit(Data=Data,Mcmc=Mcmc)
```

data(bank)

Pairs of proto-type credit cards were offered to respondents. The respondents were asked to choose between cards as defined by “attributes.”

Each respondent made between 13 and 17 paired comparisons.

Sample Attributes (14 in all):

Interest rate, annual fee, grace period, out-of-state or in-state bank, …

data(bank)

Not all possible combinations of attributes were offered to each respondent. Logit structure (independence of irrelevant alternatives makes this possible).

14,799 comparisons made by 946 respondents.

$$\Pr(\text{card 1 chosen}) = \frac{\exp[x'_{h,i,1}\beta_h]}{\exp[x'_{h,i,1}\beta_h] + \exp[x'_{h,i,2}\beta_h]}$$

$$= \frac{\exp[(x_{h,i,1} - x_{h,i,2})'\beta_h]}{1 + \exp[(x_{h,i,1} - x_{h,i,2})'\beta_h]}$$

differences in attributes is all that matters

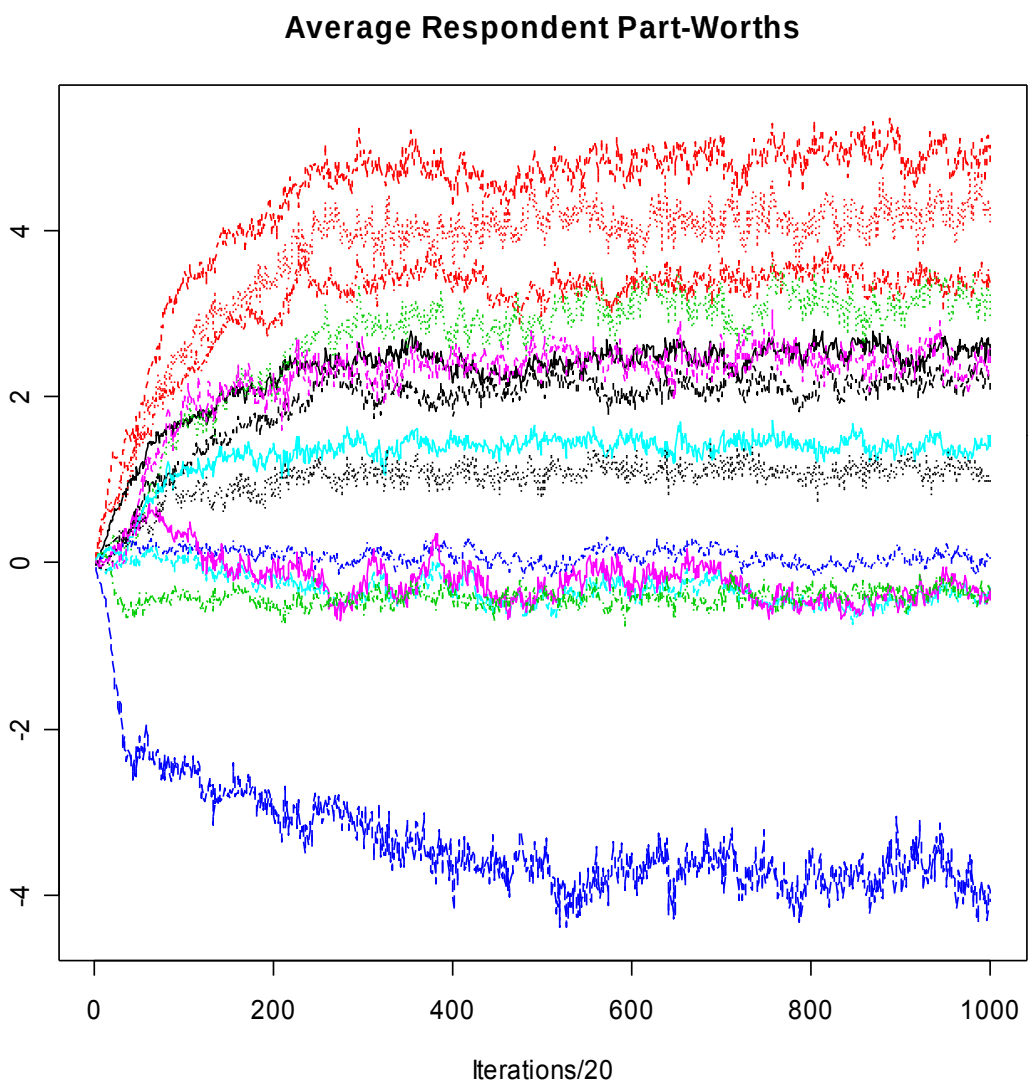
Sample observations

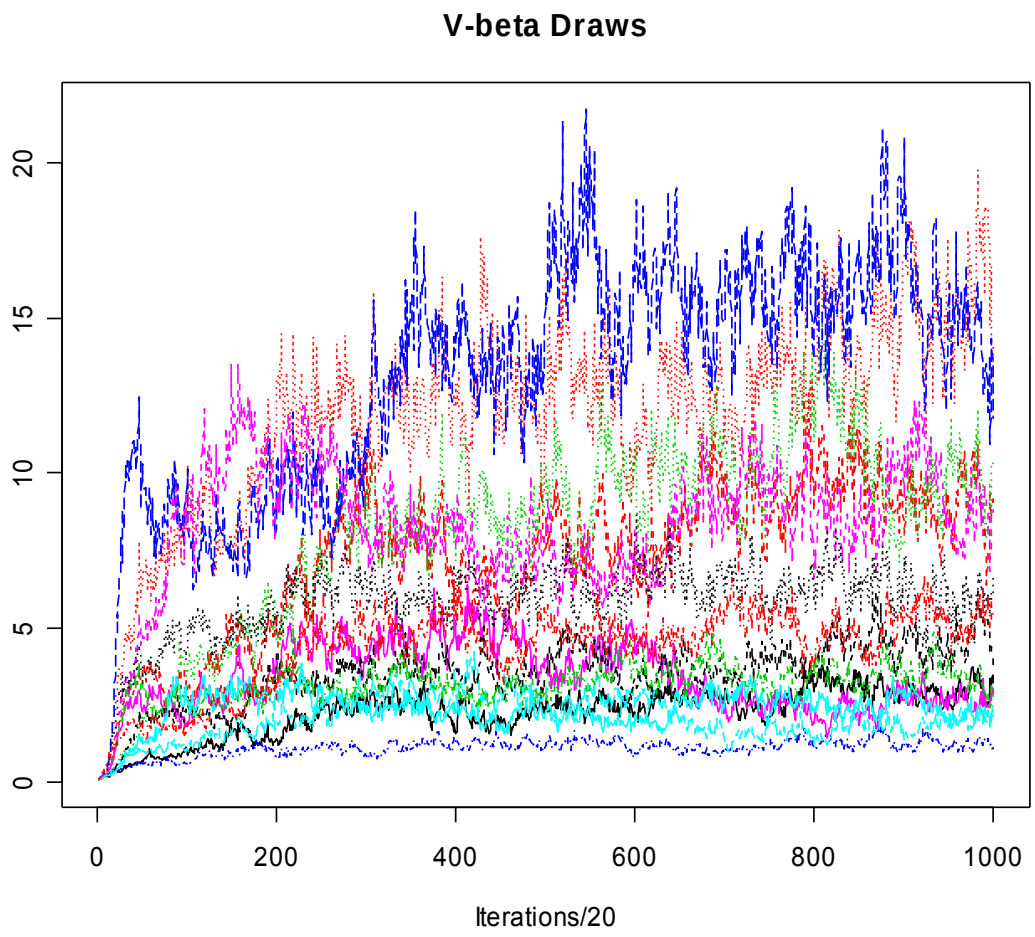
respondent 1 choose first card on first pair. Card chosen at attribute 1 on. Card not chosen had attribute 4 on.

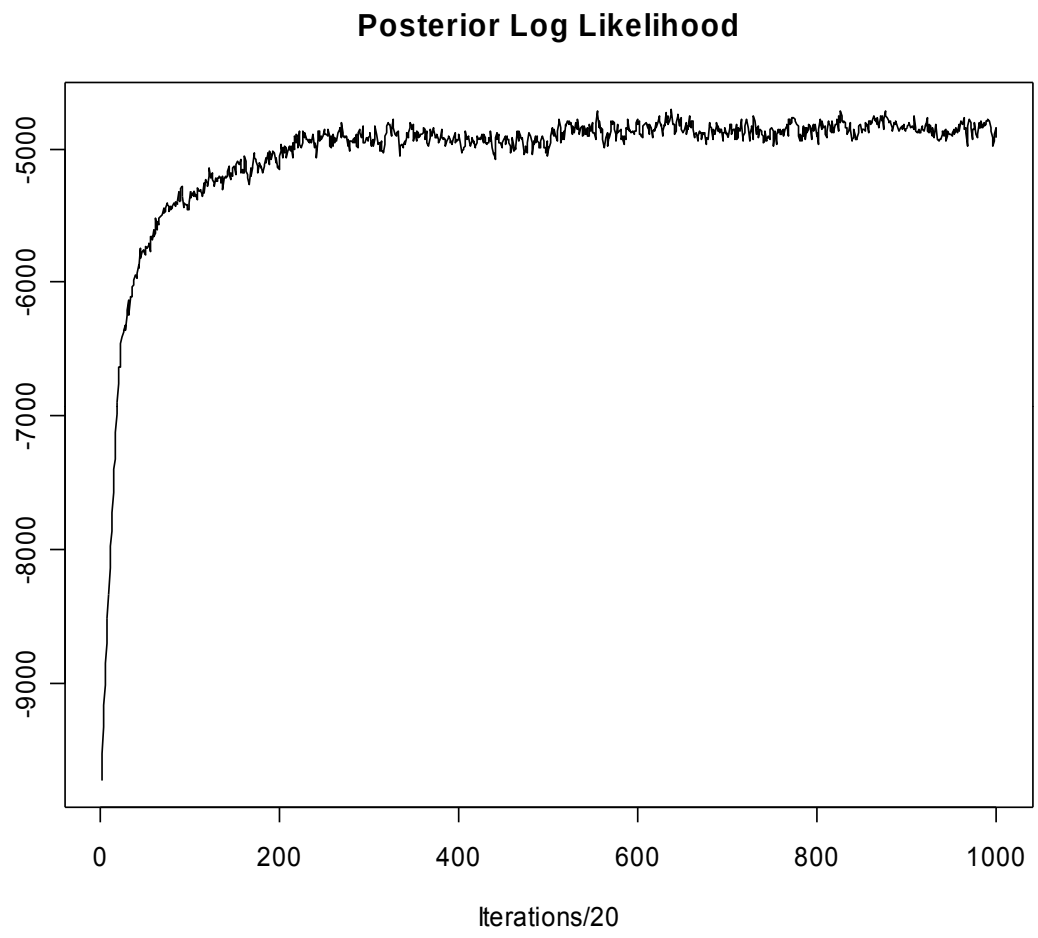
id	choice	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14
1	1	1	0	0	-1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	-1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	-1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	-1	0	0	0	0	0
1	1	0	0	0	0	0	0	1	0	1	-1	0	0	0	0
1	1	0	0	0	-1	0	0	0	0	0	0	1	-1	0	0
1	1	0	0	0	0	0	0	0	0	-1	0	0	0	-1	0
1	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0
2	1	1	0	0	-1	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	1	-1	0	0	0	0	0	0	0	0	0

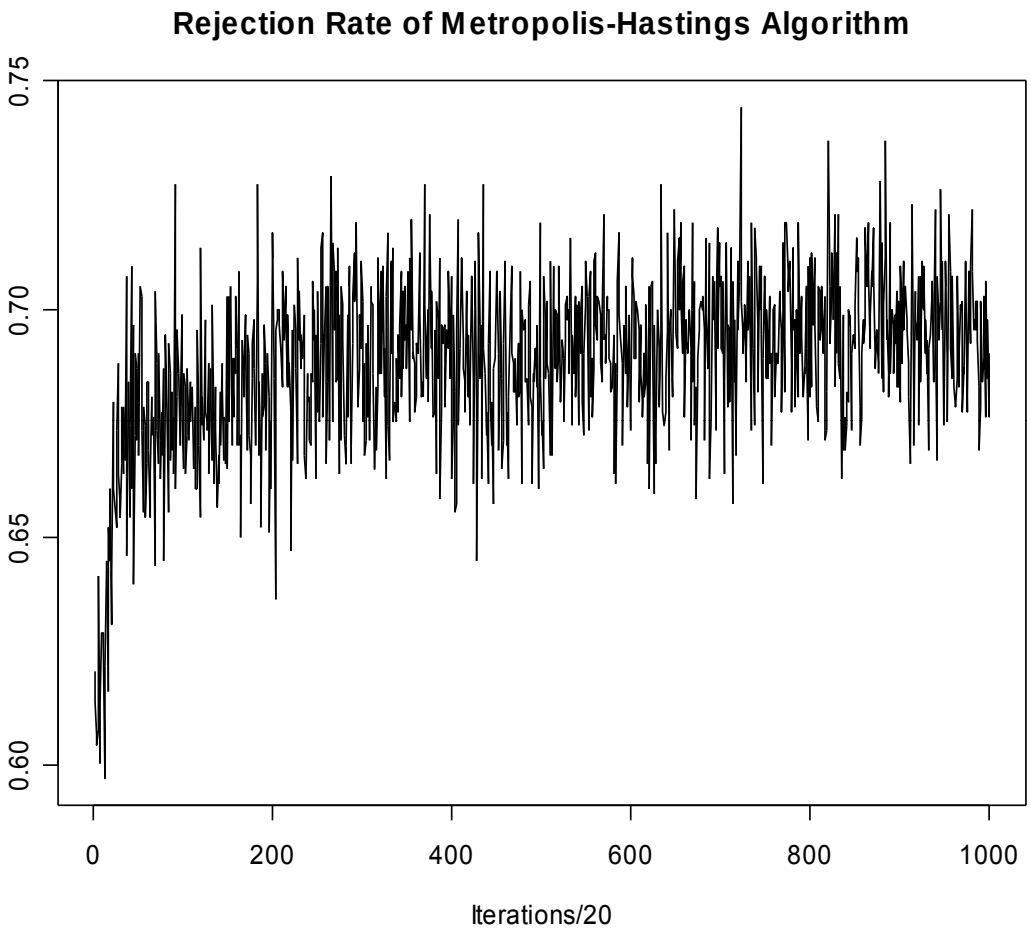
Sample demographics

id	age	income	gender
1	60	20	1
2	40	40	1
3	75	30	0
4	40	40	0
6	30	30	0
7	30	60	0
8	50	50	1
9	50	100	0
10	50	50	0
11	40	40	0
12	30	30	0
13	60	70	0
14	75	50	0

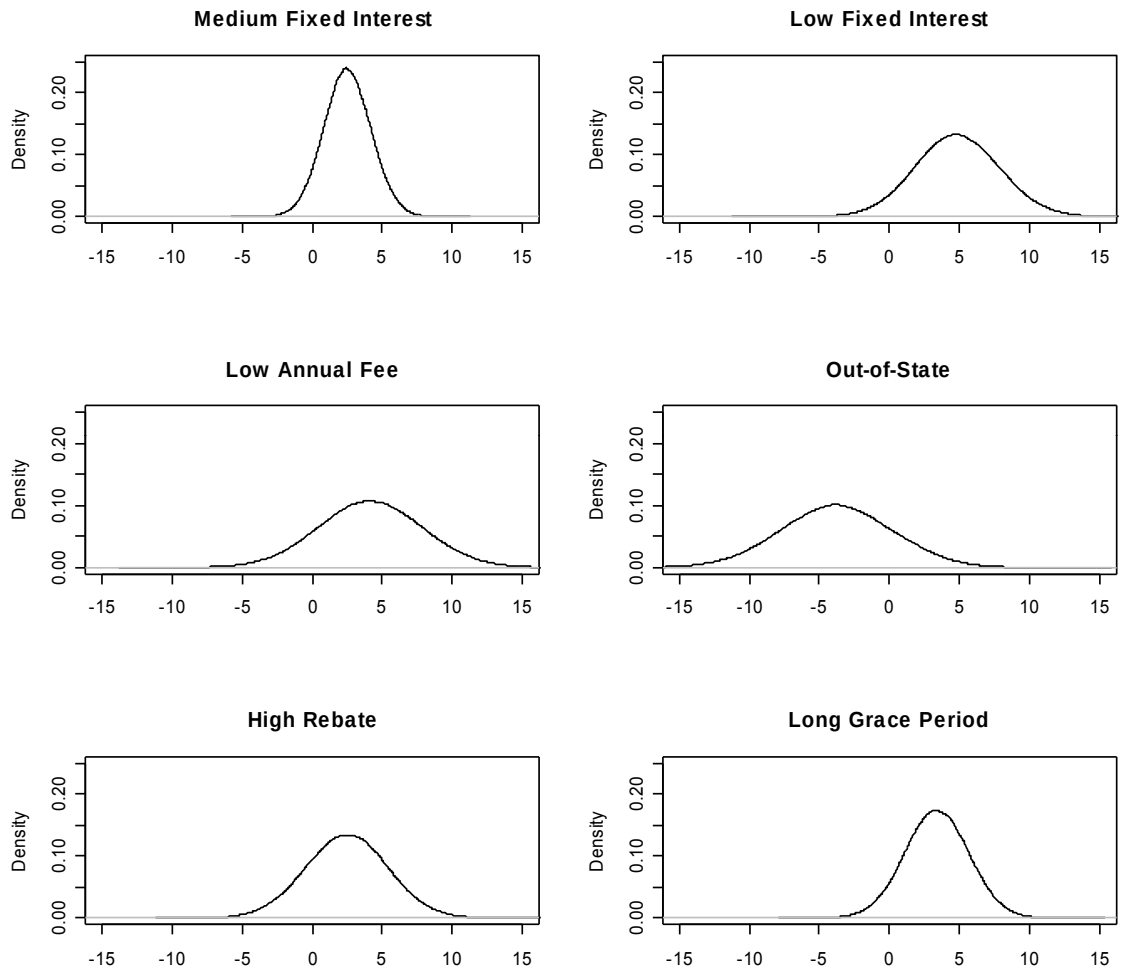

Average Respondent Part-Worths
4
2
0
-2
-4
0
200
400
600
800
1000
Iterations/20

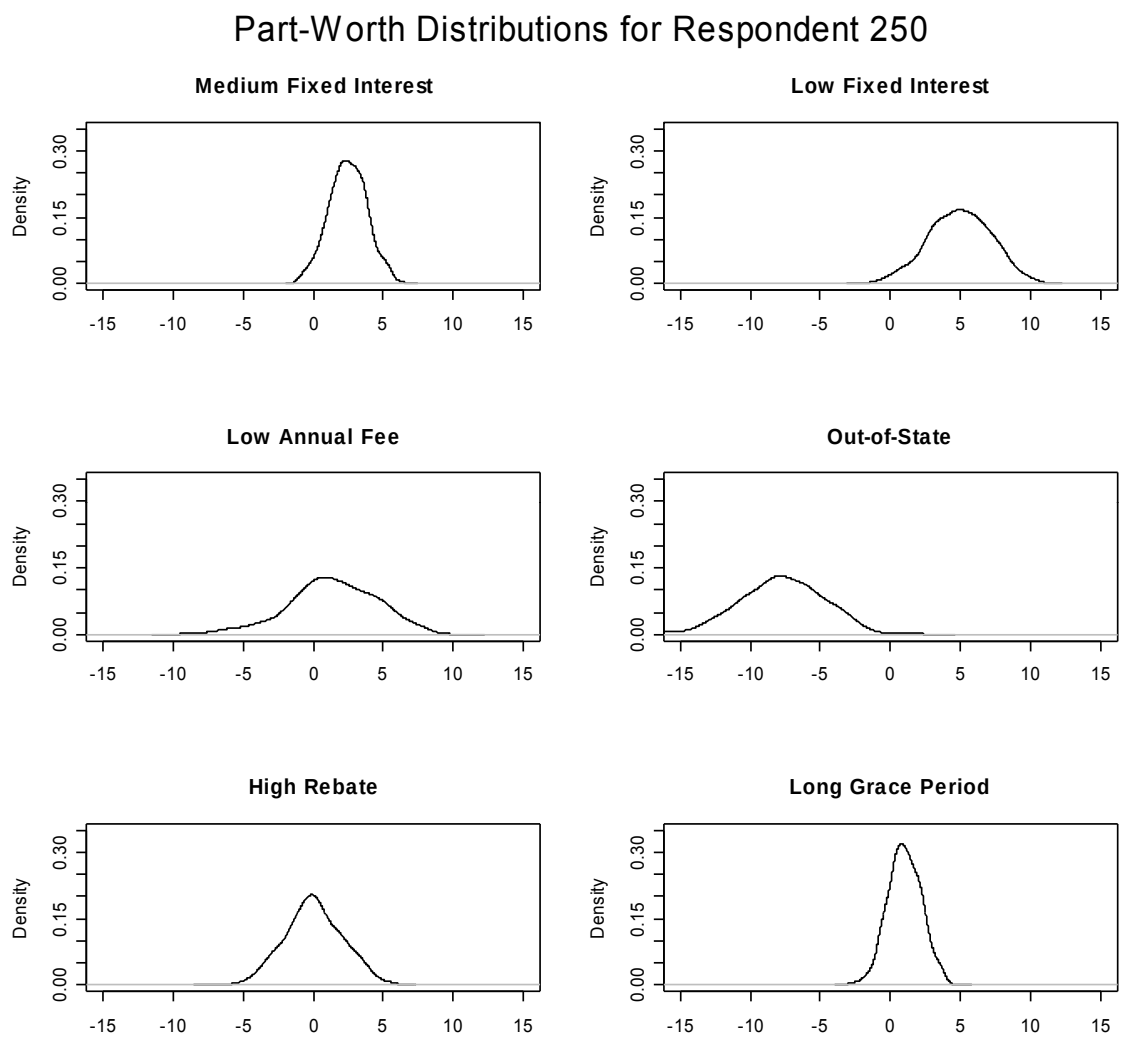

V-beta Draws
0
5
10
15
20
0
200
400
600
800
1000
Iterations/20


Posterior Log Likelihood
-5000
-6000
-7000
-8000
-9000
0
200
400
600
800
1000
Iterations/20


Rejection Rate of Metropolis-Hastings Algorithm
0.75
0.70
0.65
0.60
0
200
400
600
800
1000
Iterations/20

Distribution of Heterogeneity for Selected Part-Worths




Part-Worth Distributions for Respondent 250
Medium Fixed Interest
Low Fixed Interest
Low Annual Fee
Out-of-State
High Rebate
Long Grace Period
Density
0.00
0.15
0.30
-15
-10
-5
0
5
10
15

Extensions

Mixture of normals: **`rhierMnlRwMixture`**

Structural heterogeneity:

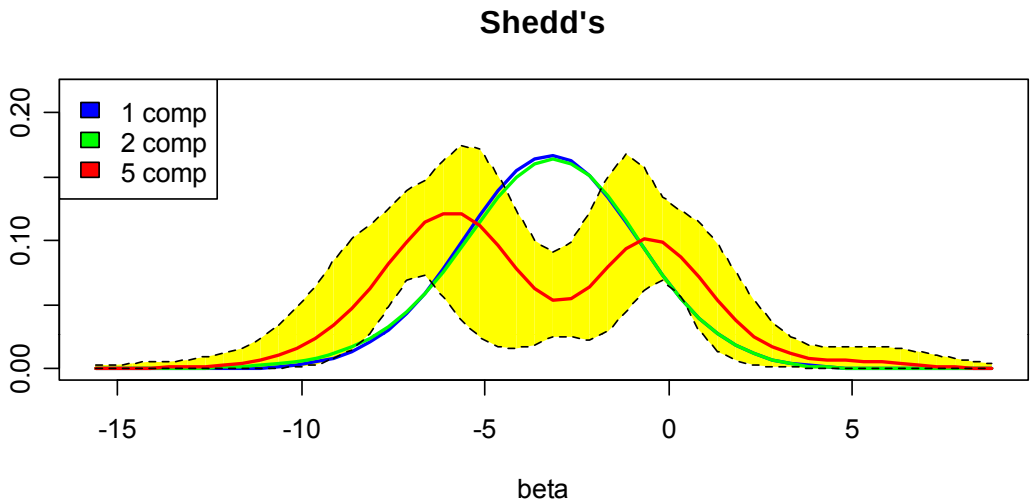
$$p(y|\theta) = r_1 p_1(y|\theta_1) + \ldots + r_k p_k(y|\theta_k)$$

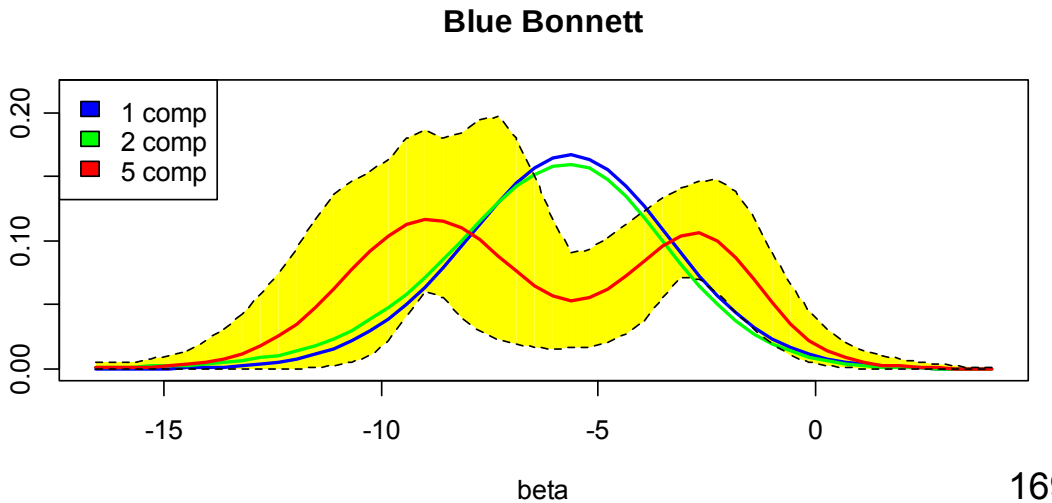
Interdependent preferences: non-iid draws

Scale use heterogeneity

Mixture of Normals

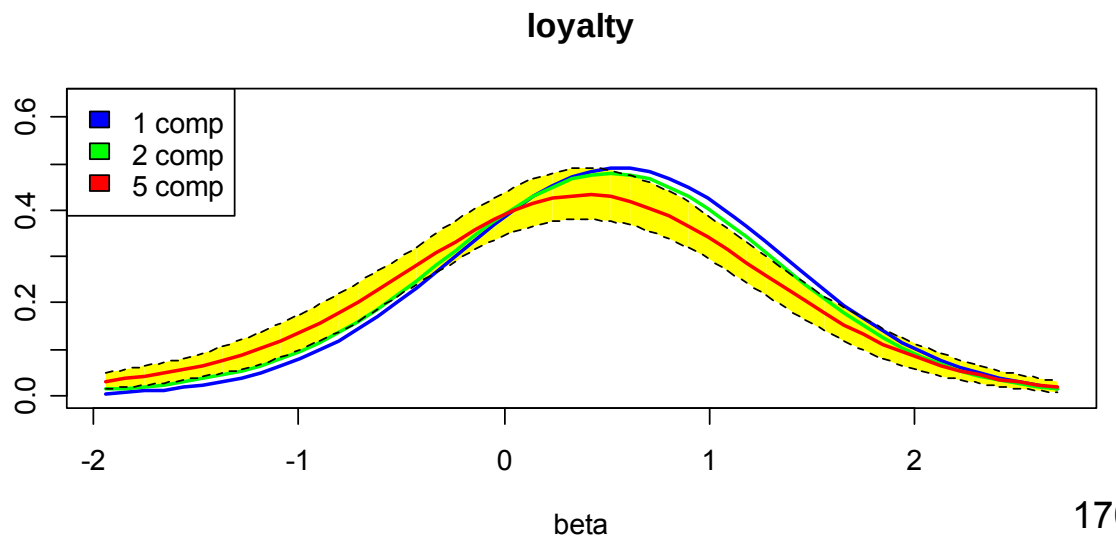
Logit model with log-price and lagged choice (called a state dependent model) as well as brand intercepts



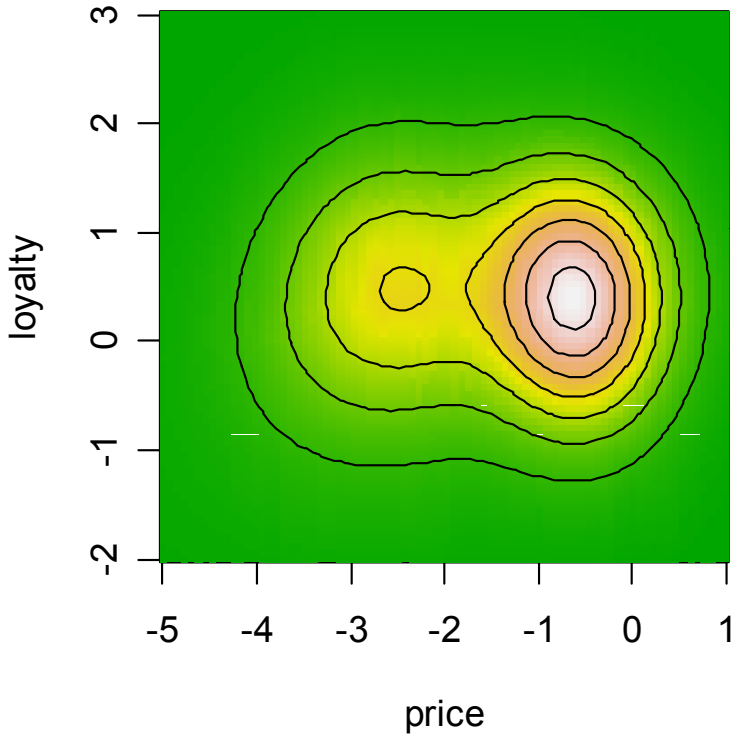


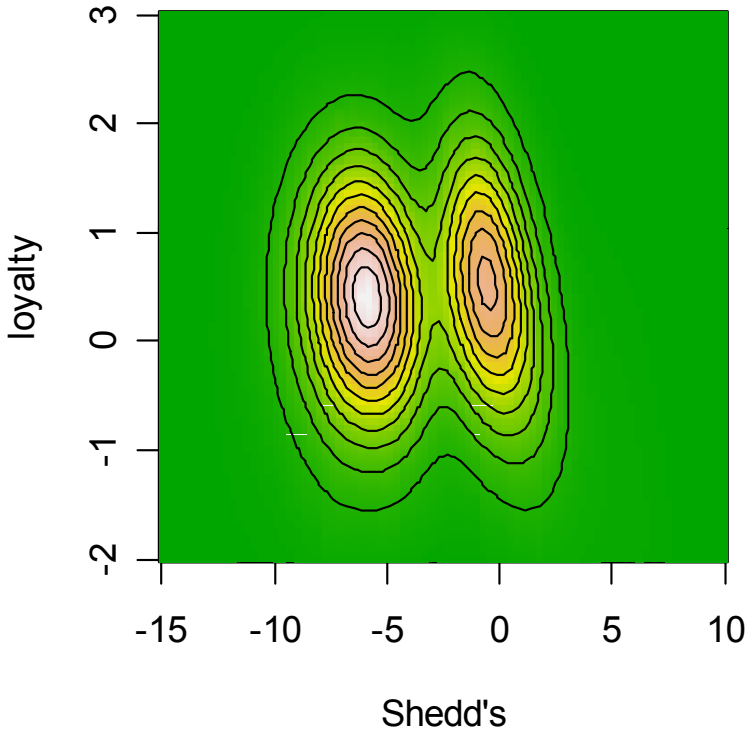
Mixture of Normals

loyalty distribution pretty normal but everything else non-normal!



Mixture of Normals





Model choice and decision theory

Decision theory

Loss: $L(a,\theta)$ where a=action; θ=state of nature

Bayesian decision theory:

$$\min_a \left\{ \bar{L}(a) = E_{\theta|D}\left[L(a,\theta)\right] = \int L(a,\theta)p(\theta \mid D)d\theta \right\}$$

note separation of Loss function from posterior/likelihood!

Profit function is the natural loss for marketing applications!

Model Selection

We are often faced with the problem of selection from a set of models. The Bayes solution is to compute the posterior probability of each model.

For the set of models: $M_1, \ldots, M_k$

compute:

$$p(M_i|y) = \frac{p(y|M_i)p(M_i)}{p(y)}$$

Posterior odds Ratio

$$\frac{p(M_1|y)}{p(M_2|y)} = \frac{p(y|M_1)}{p(y|M_2)} \times \frac{p(M_1)}{p(M_2)}$$

$$= \text{Bayes Factor} \times \text{Prior Odds}$$

Model Probabilities cont.

For parametric models,

$$p\left(y \middle| M_i\right) = \int p\left(y \middle| \theta, M_i\right) p\left(\theta \middle| M_i\right) d\theta$$

Depends on the prior! It should. One interpretation is that the model prob is the average of the "likelihood" wrt to the prior.

$$\ell^*\left(y \middle| M_i\right) = E_{\theta|M_i}\left[\ell\left(\theta \middle| y, M_i\right)\right]$$

No Improper priors. As prior becomes more diffuse, model prob declines! Implies when comparing models, diffuseness of priors can matter!

Model Probabilities cont.

The marginal density of the data is also the normalizing constant for the posterior.

$$p\left(\theta \middle| y, M_i\right) = \frac{\ell\left(\theta \middle| y, M_i\right) p\left(\theta \middle| M_i\right)}{p\left(y \middle| M_i\right)}$$

The numerator above is the un-normalized posterior. This we can always evaluate. The marginal density of the data is not always easy!

$$p\left(y \middle| M_i\right) = \int \tilde{p}\left(\theta \middle| y, M_i\right) d\theta = \frac{\tilde{p}\left(\theta \middle| y, M_i\right)}{p\left(\theta \middle| y, M_i\right)}$$

Savage-Dickey Conjugate setting

$$M_0 : \phi_1 = \phi_1^h, \quad M_1 : \text{unrestricted} \quad \text{where } \phi' = \left(\phi_1', \phi_2'\right).$$

$$p\left(\phi_2 \mid \phi_1 = \phi_1^h\right) = \left.\frac{p\left(\phi_1, \phi_2\right)}{\int p\left(\phi_1, \phi_2\right) d\phi_2}\right|_{\phi_1 = \phi_1^h}$$

$$\mathrm{BF} = \frac{\left.\int \ell\left(\phi_1, \phi_2 \mid y\right)\left(p\left(\phi_1, \phi_2\right) / p\left(\phi_1\right)\right) d\phi_2\right|_{\phi_1 = \phi_1^h}}{\int\int \ell\left(\phi_1, \phi_2 \mid y\right) p\left(\phi_1, \phi_2\right) d\phi_1 d\phi_2}$$

$$= \left.\frac{\int p\left(\phi_1, \phi_2 \mid y\right) d\phi_2}{p\left(\phi_1\right)}\right|_{\phi_1 = \phi_1^h}$$

← Marginal Posterior

← Prior

Asymptotic methods (BIC)

$$p(y \mid M_i) = \int \exp\left(\Gamma(\theta)\right) d\theta \qquad \Gamma(\theta) = \log\left(\tilde{p}(\theta \mid y)\right)$$

$$\approx \int \exp\left(\Gamma(\tilde{\theta}) - \frac{1}{2}(\theta - \tilde{\theta})' H(\tilde{\theta})(\theta - \tilde{\theta})\right) d\theta$$

$$= \exp\left(\Gamma(\tilde{\theta})\right)(2\pi)^{p_i/2} \left|H(\tilde{\theta})\right|^{-1/2} \quad \text{where } H(\tilde{\theta}) = -\frac{\partial^2 \exp\left(\Gamma(\theta)\right)}{\partial\theta\,\partial\theta'}$$

$$\approx \exp\left(\Gamma(\tilde{\theta})\right)(2\pi)^{p_i/2} n^{-p_i/2} \left|\inf_i(\tilde{\theta})/n\right|^{-1/2}$$

$$\approx p\left(y \mid \hat{\theta}_{MLE}, M_i\right) n^{-p_i/2} \qquad \text{as } n \to \infty$$

Computing Model Probs

For non-conjugate problems, there are three approaches:

1. Importance Sampling
2. Use of MCMC draws (NR)
3. Chib's Method (useful for latent var models)

BF using MCMC draws

$$\int \frac{q(\theta)}{\tilde{p}(\theta \mid y, M_i)} p(\theta \mid y, M_i)\, d\theta = \int \frac{q(\theta)}{p(\theta \mid M_i)\, p(y \mid \theta, M_i)} p(\theta \mid y, M_i)\, d\theta$$

$$= \frac{1}{p(y \mid M_i)} \int q(\theta)\, d\theta$$

$$= \frac{1}{p(y \mid M_i)}$$

$$E_{\theta \mid y, M_i}\left[\frac{q(\theta)}{\tilde{p}(\theta \mid y, M_i)}\right] = \frac{1}{p(y \mid M_i)}$$

Special case (Newton-Raftery)

If $q(\theta) = p(\theta \mid M_i)$,

$$p(y \mid M_i) = \frac{1}{E_{\theta|y,M_i}\left[\frac{1}{\ell(\theta \mid M_i)}\right]}$$

$$\hat{p}(y \mid M_i) = \frac{1}{\frac{1}{R}\sum_{r=1}^{R}\frac{1}{\ell(\theta^r \mid M_i)}}$$

`logMargDenNR`

Appeal: uses MCMC draws and likelihood

Problems: extreme sensitivity to outliers.
Distribution of likelihood values is non-uniform!

Simultaneity

Exogeneity

$$\pi(y,x|\theta_y,\theta_x,\alpha) = \pi(y|x,\theta_y,\alpha)\ \pi(x|\theta_x,\alpha)$$

The variable x is exogenous to y if the joint distribution can be factored, if α does not exist, and cross-restrictions between θ_y, and θ_x, do not exist.

Otherwise, x and y are endogenous i.e., dependent variables from the system of study.

Endogeneity

$$\pi(y,x|\theta) \neq \pi(y|x,\theta_y)\ \pi(x|\theta_x)$$

For example:

$$\pi(\text{demand},\text{price}|\text{preferences},\text{sensitivities})$$
$$\neq \pi(\text{demand}|\text{price},\theta_d)\ \pi(\text{price}|\theta_p)$$

Instrumental variables

$$x = \delta z + \varepsilon_1 \qquad\qquad x = \delta z + \alpha_x w + u_1$$

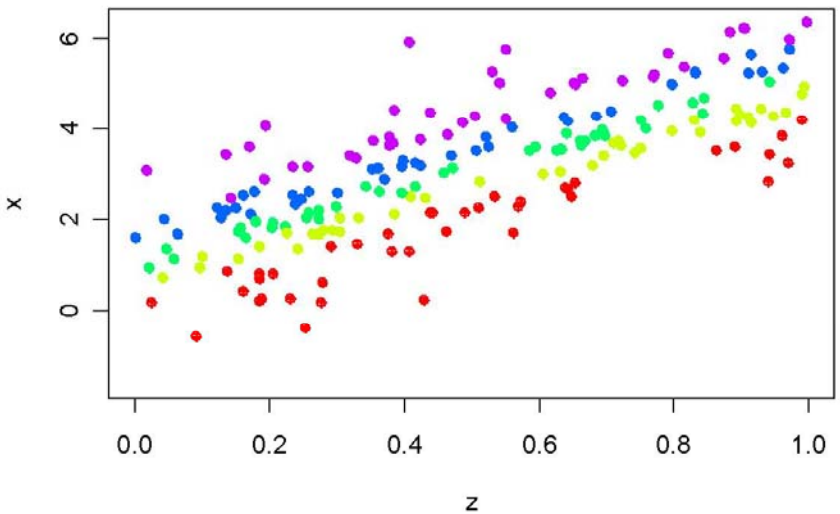
$$\Longleftrightarrow$$

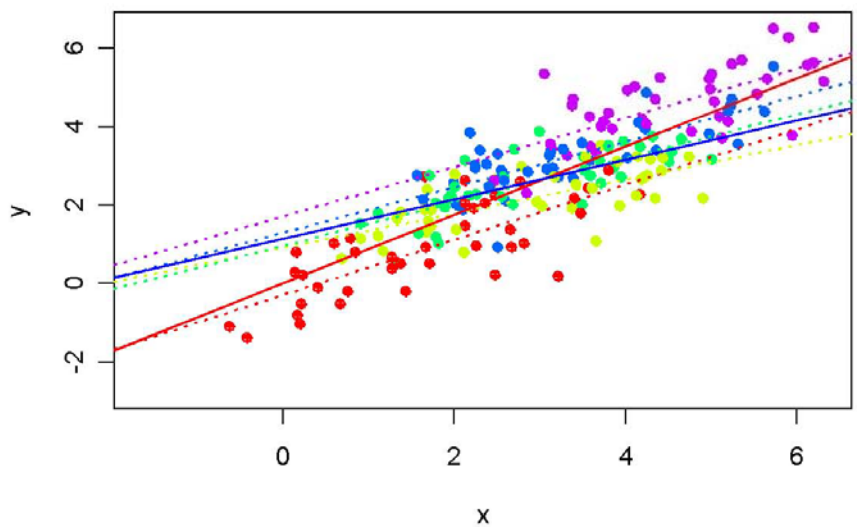
$$y = \beta x + \varepsilon_2 \qquad\qquad y = \beta x + \alpha_y w + u_2$$

Models that omit *w* are mis-specified and cannot be written in the form $\pi(y,x|\theta) = \pi(y|x,\theta_y)\ \pi(x|\theta_x)$ because $E[\varepsilon_2|x]=f(x)$ if ε_1 and ε_2 are correlated.

Examples of x depending on ε_2 include omitted factors (*w*) that are demand shocks or unobserved characteristics.

Correlation (ε_1,ε_2) = 0.8





Color groups correspond to ε_1 realizations

Likelihood for x and y

$$x = \delta z + \varepsilon_1$$
$$y = \beta\delta z + \left(\beta\varepsilon_1 + \varepsilon_2\right)$$

$$\pi\left(x, y\right) = \pi\left(\varepsilon_1, \varepsilon_2\right)\left|J_{(\varepsilon_1,\varepsilon_2)\to(x,y)}\right|$$

$$\left|J_{(\varepsilon_1,\varepsilon_2)\to(x,y)}\right| = \begin{vmatrix} \dfrac{\partial\varepsilon_1}{\partial x} & \dfrac{\partial\varepsilon_1}{\partial y} \\ \dfrac{\partial\varepsilon_2}{\partial x} & \dfrac{\partial\varepsilon_2}{\partial y} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ \beta & 1 \end{vmatrix} = 1$$

Estimation: i) Gibbs; ii) Metropolis-Hastings

Gibbs estimation (`rivGibbs`)

$$x = z'\delta + \varepsilon_1$$

$$y = \beta x + w'\gamma + \varepsilon_2$$

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} \sim N(0, \Sigma)$$

$$\beta, \gamma \mid \delta, \Sigma, x, y, w, z$$

$$\delta \mid \beta, \gamma, \Sigma, x, y, w, z$$

$$\Sigma \mid \beta, \gamma, \delta, x, y, w, z$$

Given δ, solve for ε_1 and use **`breg`**

Isolate x in both eqns, stack and transform for **`breg`**.

Standard inverted Wishart draw

Bayesian statistics and marketing

BSM is a self-contained text for marketing researchers.

Bayesm package includes:

- Linear regression, multivariate regression, logit, binary probit and multinomial probit models.
- Models for count data, instrumental variables, mixture distributions for heterogeneity, and much more.