Bayesian and ML estimation

- The usual Bayesian estimate is the mean of the posterior, as this minimizes the mean-squared error of the estimate given the posterior.
- The ML estimate is the mode of the likelihood function.
- If the prior is relatively flat, the likelihood is approximately normal (and symmetric and unimodal), and there are no nuisance parameters, then the two approaches will give similar answers.

Dealing with nuisance parameters

- The two approaches have different ways of dealing with nuisance parameters:
- The likelihood approach typically uses the "concentrated" or "profile likelihood", where the maximum conditional likelihood value of the nuisance parameter is plugged into the joint likelihood (can result in a "chicken or the egg?" problem - you need a conditioning value to plug in, hence iterative methods).
- The Bayesian approach is to integrate the nuisance parameter out of the joint posterior (to obtain the marginal posterior for the parameter(s) of interest).

Bayesian inference

- The unscaled posterior $p(\theta|y) \propto p(\theta)y|\theta)$ has all the shape information about the posterior.
- The exact posterior is found by dividing the unscaled posterior by its integral over the whole range of the parameter θ to obtain the "normalizing constant".
- A closed form for this integral only occurs for a few cases. In other cases the integral has to be found numerically. This may be hard to do, especially in high dimensional problems.
- Computational Bayesian statistics is based on drawing a Monte Carlo random sample from the unscaled posterior. This replaces very difficult numerical calculations with the easier process of drawing random variables. Sometimes, particularly for high dimensional cases, this is the only feasible way to find the posterior.

MCMC inference

- The distribution of a random sample from the posterior approaches the exact posterior distribution as the randomly drawn sample size increases.
- Estimates of parameters can be calculated from statistics calculated from the random sample.
- These estimates can achieve any required degree of accuracy by setting the MCMC sample size large enough.
- Exploratory data analysis (EDA) techniques can be used to explore the posterior distribution given the MCMC sample. This is essentially the overall goal of Bayesian inference.

The posterior can be scaled after we obtain the MCMC sample

- Once an MCMC sample from $p(\theta|y)$, given n observation for y, is obtained, the normalizing constant can be estimated with arbitrary accuracy.
- E.g. Partitioning the range of values of draws into intervals, calculating frequency counts of values in each interval $[\theta_{k-1}, \theta_k]$, then scaling the frequency distribution (histogram), $h_k(\theta_{(i)}) \in [\theta_{k-1}, \theta_k]$ of MCMC draws, $\theta_{(i)}$ by $\sum_{i=1}^{R} h_k(\theta_{(i)})$, with R being the MCMC sample size, approximates the exact (small sample for y) posterior distribution, with accuracy increasing with R.

The key theoretical result

- Equation (1) is known as the 'detailed balance equation' because it equates the rates of moves through states (so balanced) for every possible pair of states (hence detailed).
- $$\pi(x)P(x,y) = \pi(y)P(y,x) \text{ for all } x, y \in S, \qquad (1)$$

  where the state space S is the appropriate subset of $\mathbb{R}^n$ representing the support of x, y.

The key theoretical result

- Equation (1) is known as the ‘detailed balance equation’ because it equates the rates of moves through states (so balanced) for every possible pair of states (hence detailed).
- $$\pi(x)P(x,y) = \pi(y)P(y,x) \,\text{for all}\, x, y \in S, \tag{1}$$
  where the state space S is the appropriate subset of $\mathbb{R}^n$ representing the support of x, y.
- This leads to the key result:
- If there is a distribution p satisfying the detailed balance equation, (1), for an irreducible chain, then the chain is positive recurrent and reversible with stationary distribution π.

The key result for MCMC

- Metropolis et al. (1953) showed that it is then ***always*** possible to construct a Markov chain with stationary distribution π by finding transition probabilities $P(x, y)$ satisfying (1).

The key result for MCMC

- Metropolis et al. (1953) showed that it is then ***always*** possible to construct a Markov chain with stationary distribution π by finding transition probabilities $P(x, y)$ satisfying (1).
- This provides an algorithm for constructing Markov chains that has weak requirements and so has wide applicability.

The key result for MCMC

- Metropolis et al. (1953) showed that it is then ***always*** possible to construct a Markov chain with stationary distribution π by finding transition probabilities $P(x, y)$ satisfying (1).
- This provides an algorithm for constructing Markov chains that has weak requirements and so has wide applicability.
- The above results, in particular, convergence to the limiting distribution, the ergodic theorem and the central limit theorem, all hold for continuous state spaces with only minor technical modifications required (see Gamerman and Lopes, 2006).

The key result for MCMC

- Metropolis et al. (1953) showed that it is then ***always*** possible to construct a Markov chain with stationary distribution π by finding transition probabilities $P(x, y)$ satisfying (1).
- This provides an algorithm for constructing Markov chains that has weak requirements and so has wide applicability.
- The above results, in particular, convergence to the limiting distribution, the ergodic theorem and the central limit theorem, all hold for continuous state spaces with only minor technical modifications required (see Gamerman and Lopes, 2006).
- The above theory provides the means by which sampling from **virtually any** posterior distribution π can be achieved.

MCMC as a useful algorithm

- The basic Metropolis algorithm is to set the posterior distribution of interest, π, as the limiting distribution of an ergodic Markov chain with transition kernel P.

MCMC as a useful algorithm

- The basic Metropolis algorithm is to set the posterior distribution of interest, π, as the limiting distribution of an ergodic Markov chain with transition kernel P.
- The various algorithms that build on this, in particular Gibbs sampling and Metropolis-Hastings (MH), are concerned with various methods of providing proposal distributions P to be sampled from.

MCMC as a useful algorithm

- The basic Metropolis algorithm is to set the posterior distribution of interest, π, as the limiting distribution of an ergodic Markov chain with transition kernel P.
- The various algorithms that build on this, in particular Gibbs sampling and Metropolis-Hastings (MH), are concerned with various methods of providing proposal distributions P to be sampled from.
- So, Markov chain Monte Carlo (MCMC) is the generic algorithm that ensures the Markov chain's stationary distribution *is* the posterior distribution of the associated parameter/unknown quantity.

Metropolis algorithm

1. Choose θ_0, set $t = 0$.
2. Draw θ_c from the proposal distribution $q(\theta_c|\theta_t)$, where q are the transition probabilities$P(x, y)$.
3. Calculate
$$r = \frac{p(\theta_c)}{p(\theta_t)}.$$
4. If $r \geq 1$ set $\theta_{t+1} = \theta_c$, otherwise set $\theta_{t+1} = \theta_c$ with probability r, $\theta_{t+1} = \theta_t$ with probability $1 - r$.
5. Set $t = t + 1$ and repeat the above steps (from 2).

Metropolis acceptance rate

- The probability that a θ_c is accepted is thus

$$p(\theta_{t+1} = \theta_c | \theta_t) = \min \left[\frac{p(\theta_c)}{p(\theta_t)}, 1 \right].$$

- For proof that these draws converge, see Lancaster, p213 and Gamerman and Lopes (2006).
- An important feature of this algorthm is that, since we only calculate the ratio $\frac{p(\theta_c)}{p(\theta_t)}$, we do not need to know the normalizing constant.

Values in the chain can repeat, so the chain can be highly autocorrelated

- We only need the kernel of the target conditional posterior distribution.
- Since we set $\theta_{t+1} = \theta_t$ with probability $1 - r$, the MCMC chain will have repeated values in the chain. If the acceptance rate is low, the chain will get stuck at the same value for many iterations.
- Since r depends on θ_c, which is drawn from $q(\theta_c|\theta_t)$, parameters of q can be adjusted to control the acceptance rate, r. These are called the tuning parameters.

MH algorithm

- Hastings (1970) generalized this algorithm by replacing the acceptance rate, r, with

$$r = \frac{p(\theta_c)q(\theta_t|\theta_c)}{p(\theta_t)q(\theta_c|\theta_t)},$$

- the probability of acceptance is then

$$p(\theta_{t+1} = \theta_c|\theta_t) = \min\left[\frac{p(\theta_c)q(\theta_t|\theta_c)}{p(\theta_t)q(\theta_c|\theta_t)}, 1\right].$$

- If $q(\theta_c|\theta_t)$ is symmetric, so $q(\theta_c|\theta_t) = q(\theta_t|\theta_c)$, then this reduces to the Metropolis algorithm.
- See Gamerman and Lopes (2006) for proofs of convergence to the stationary target distribution $p(\theta)$.
- Two standard choices for $q(\theta_c|\theta_t)$: random walk and independence MH.

The Independence MH Sampler

- Here we let $q(\theta_c|\theta_t) = q(\theta_c)$, i.e. it is independent of the previous draw, so we evaluate $q(\theta_c)$ and $q(\theta_t)$ instead of $q(\theta_c|\theta_t)$ and $q(\theta_t|\theta_c)$. The probability of acceptance is then

$$p(\theta_{t+1} = \theta_c|\theta_t) = \min\left[\frac{p(\theta_c)q(\theta_t)}{p(\theta_t)q(\theta_c)}, 1\right].$$

- In this case, the closer the choice of q is to p, the higher the acceptance rate (if $q = p$ notice that we always accept).
- So we would like the proposal distribution to be a good approximation to the target posterior distribution. The better the approximation, the higher the acceptance rate.

Choosing the proposal/candidate density, q.

- The constraint is that q must be a distribution that we can generate pseudo-random draws from.
- Note that, according to the CLT, the normal distribution ought to be a reasonable approximation in many cases.
- We have to worry about the possibility of fat tails and skewness in the target distribution however, so often a Student-t distribution with small degrees of freedom is used.

The Random Walk MH Sampler

- Let $\theta_c = \theta_t + e_t$, where $e_t \sim q$ and $E(e_t) = 0$ (usually q is a normal distribution).
- Here $q(\theta_c|\theta_t) = q(\theta_c - \theta_t)$, which is symmetric and so this reduces to the Metropolis algorithm,

$$p(\theta_{t+1} = \theta_c|\theta_t) = \min\left[\frac{p(\theta_c)}{p(\theta_t)}, 1\right].$$

- This means that a candidate θ_c that has a higher value of the target density than the target density of the current value, θ_t will always be accepted. The chain will always move 'uphill'.
- To code this, the candidate is accepted if a random draw from a $U[0,1]$ is less than r.

Random walk MH

- A candidate with a lower target density value will only be accepted with probability $r = \frac{p(\theta_c)}{p(\theta_t)} < 1$, so there is a certain probability, r, that the chain will move ‘downhill’.
- This allows a chain the a random walk candidate density to move around the whole parameter space over time.
- In the MH algorithm, the candidate density is centered around the previous value in the chain, so it moves every time a candidate draw is accepted.
- The chain will generally have many accepted candidates, but most moves will be a short distance, so it can take a long time move around the whole parameter space.

Random walk MH

- If the acceptance rate is too high, the algorithm will not move into the tails of the distribution very often, so these will not be accurately approximated.
- The key is to tune the candidate density (typically by changing the variance of q) so that the acceptance/rejection rate is not too high.
- In practice, tuning to obtain rejection rates between 0.3 and 0.5 have been found to work well and so are recommended.
- see **normMixMH.R** and **normmixMHbolstad.R**, **rejrate.R** calculates the rejection rate.
- Try varying the candidate variance (tuning parameter) and observe the effect on the rejection rate, etc.

MH for multiple parameters

- Suppose $\theta = (\theta_1, \ldots, \theta_k)$. Let $q(\theta_c|\theta_t)$ be the candidate density when the chain is at θ_t as above (though θ_t is now a $1 \times k$ vector), and let $p(\theta|y)$ be the posterior density.
- The reversibility condition for all states is

$$p(\theta_c|y)q(\theta_c|\theta_t) = p(\theta_t|y)q(\theta_t|\theta_c).$$

- Most chains won't satisfy the reversibility condition, but the balance can be restored by introducing the probability of moving (the acceptance rate) of

$$p(\theta_{t+1} = \theta_c|\theta_t) = \min\left[\frac{p(\theta_c|y)q(\theta_t|\theta_c)}{p(\theta_t|y)q(\theta_c|\theta_t)}, 1\right].$$

Multivariate and Blocking RW and Independent MH

- For a RW candidate distribution, $q = q(|\theta_c - \theta_t|)$ is symmetric about zero so, as before, q cancels from the acceptance rate.
- For the independence chain, as for the single parameter, $q(\theta_c|\theta_t) = q(\theta_c)$. So everything looks the same, we are just drawing a vector of values instead of a scalar.
- For blocks of parameters, say θ_1 and θ_2, we condition on current values of parameters in other blocks, so we evaluate $p(\theta_{1c}|\theta_{2t}, y)$ and $p(\theta_{1t}|\theta_{2t}, y)$ to calculate the acceptance rate.

Gibbs sampling as a special case of MH

- Suppose we have two blocks of parameters, θ_1 and θ_2.
- If we use the true conditional density as the candidate density at each step, conditioning on the other parameters, then $q(\theta_{1c}|\theta_{1t}) = p(\theta_{1c}|\theta_{2t}, y)$ and $q(\theta_{1t}|\theta_{1c}) = p(\theta_{1t}|\theta_{2t}, y)$, so the acceptance rate is

$$r = \frac{p(\theta_c|y)q(\theta_t|\theta_c)}{p(\theta_t|y)q(\theta_c|\theta_t)} = \frac{p(\theta_{1c}|\theta_{2t}, y)p(\theta_{1t}|\theta_{2t}, y)}{p(\theta_{1t}|\theta_{2t}, y)p(\theta_{1c}|\theta_{2t}, y)} = 1.$$

- So the candidate will always be accepted at every step.
- This is the Gibbs sampler: we draw from the blocks of conditional densities, always accepting the draws.

Use MH and Gibbs summary

- Metropolis-Hastings algorithm is the fundamental method for finding a Markov chain that has the long-run stationary distribution that is the same shape as the posterior distribution of interest.
- There are a number of choices in setting up the algorithm.
- It can be implemented using either random-walk or independent candidate densities.
- An option is to start with the RW MH to get an initial estimate of the posterior, then employ an independent candidate that approximates this posterior.

MCMC general approach

- We set up a Markov chain that has the posterior distribution as its limiting/stationary distribution
- We let the chain run a long time, until it (hopefully) has approached the limiting distribution.
- Any value taken after than initial ‘burn-in’ time approximates a random draw from the posterior distribution.
- Values taken from the Markov chain at time points close to each other are highly correlated. However, values at widely separated time points are approximately independent.

Diagnostics for MCMC output

- To determine if the candidate density is OK and enough draws have been obtained, look at:
- the time plot and the ACF of the chain,
- the histogram and kernel density,
- the rejection rate, n.s.e's and numerical efficiency values.
- For example, **bayesm** in R has a function to calculate n_{eff}, etc., called numEff.

Adjustments available in practice, based on above diagnostics of MCMC chain

- Draws can be made blockwise, each block conditional on the parameters in other blocks.
- Draws can be obtained from parallel runs, with different starting values, or one long run.
- Draws at the beginning "burn-in" period can be dropped to allow the algorithm time to converge to the stationary distribution.
- Thinning the MCMC sample can be used to reduce this autocorrelation (hence the LAG parameter in the R code)., i.e. all draws can be retained in the sample to be analyzed, or only every kth draw retained to reduce autocorrelation in the sample.

Inference from an MCMC sample

- Mean, median, standard deviation, quantiles can all be calculated directly using the MCMC sample
- see **Bayesregexample.R** for a simple example
- Testing a one-sided hypothesis: calculate the posterior probability of the null and compare it to the alternative.
- Suppose $H_0 : \theta \leq \theta_0$ and $H_1 : \theta > \theta_0$, then we calculate the quantile at θ_0 and sum all values below, and all values above this point to get the two probabilities.
- To test a point null hypothesis $H_0 : \theta = \theta_0$ vs. $H_1 : \theta \neq \theta_0$, since $p(\theta = \theta_0) = 0$ we can construct a credible interval and see if 0 lies in this interval. If so, we cannot reject H_0.

Bayesian inference for GLM

- With a uniform prior for β and $\ln\sigma^2$, the conditional posterior density is normal with mean and variance = the GLS estimators,

$$\hat{\beta}_{GLS} = (X_\lambda' X_\lambda)^{-1} X_\lambda' y_\lambda == (X' V^{-1} X)^{-1} X' V^{-1} y.$$

$$\mathrm{var}_{GLS}(\beta) = \sigma^2 (X_\lambda' X_\lambda)^{-1} = \sigma^2 (X' V^{-1} X)^{-1}$$

Bayesian inference for GLM

- With a uniform prior for β and lnσ^2, the conditional posterior density is normal with mean and variance = the GLS estimators,

$$\hat{\beta}_{GLS} = (X_\lambda' X_\lambda)^{-1} X_\lambda' y_\lambda == (X'V^{-1}X)^{-1} X'V^{-1} y.$$

$$\mathrm{var}_{GLS}(\beta) = \sigma^2 (X_\lambda' X_\lambda)^{-1} = \sigma^2 (X'V^{-1}X)^{-1}$$

- Similarly, conditional posterior for σ^2 given V is

$$\sigma^2 | V \sim \mathrm{Gamma}(\nu/2, (y^\lambda - X^\lambda \hat{\beta}_{GLS})'(y^\lambda - X^\lambda \hat{\beta}_{GLS})/2).$$

Conditional posterior for V

- The conditional posterior for V given β and σ^2 and a prior, $p(V)$, has the form

$$p(V|y, X, \beta, \sigma^2) \propto p(V)|V|^{-1/2} \exp\left[-\frac{1}{2\sigma^2}(y - X\beta)' V^{-1}(y - X\beta)\right].$$

Conditional posterior for V

- The conditional posterior for V given β and σ^2 and a prior, $p(V)$, has the form

$$p(V|y,X,\beta,\sigma^2) \propto p(V)|V|^{-1/2} \exp\left[-\frac{1}{2\sigma^2}(y-X\beta)'V^{-1}(y-X\beta)\right].$$

- Given a uniform prior for the elements of $\alpha = (\alpha_1, \alpha_2)'$, in $\sigma_i^2 = \exp(\alpha_1 + \alpha_2 z_i)$, prior becomes part of the normalizing constant, and $|V| = \prod_{i=1}^{n} \sigma_i^2$. The above conditional is not in a known/convenient distributional form from which to draw, so a MH step is required.

Some alternative specifications

- Alternatively, to simplify a little, we can drop the scaling factor in the variance, σ^2, i.e. specify $e \sim N(0, V)$ instead of $e \sim N(0, \sigma^2 V)$, eliminating the need to draw from $\sigma^2 | V$.

Some alternative specifications

- Alternatively, to simplify a little, we can drop the scaling factor in the variance, σ^2, i.e. specify $e \sim N(0, V)$ instead of $e \sim N(0, \sigma^2 V)$, eliminating the need to draw from $\sigma^2 | V$.
- Another alternative that allows GS at all steps is to draw each σ_i^2 independently from a Gamma and construct the diagonal V matrix at each iteration. It would be interesting to evaluate how each of these alternatives performs.

Random walk MH step

- Using a random walk MH algorithm to obtain draws, the above conditional is evaluated each MCMC round, r,

$$p(V^{(r)}|\beta^{(r)}, \sigma^{2(r)}, y, X),$$

and the previous and the candidate draw are used to calculate the acceptance probability as follows.

Random walk HM algorithm

- The idea is to take a random step away from the current location, drawing

$$V^{(r)} = V^{(r-1)} + e, \quad e \sim N(0, s^2\Sigma)$$

where s^2 is the tuning parameter chosen to obtain an acceptance rate in the range 0.3 to 0.5, and either $\Sigma = I$ or $\Sigma =$ an approximate estimate of the posterior covariance matrix, then

Random walk HM algorithm

- The idea is to take a random step away from the current location, drawing

$$V^{(r)} = V^{(r-1)} + e, \;\; e \sim N(0, s^2\Sigma)$$

  where s^2 is the tuning parameter chosen to obtain an acceptance rate in the range 0.3 to 0.5, and either $\Sigma = I$ or $\Sigma =$ an approximate estimate of the posterior covariance matrix, then

- Evaluate the conditional posterior at the old and new locations and accept the new draw with probability

$$p(accept) = \min\{1, \frac{p(V^{(r)}|\beta^{(r)}, \sigma^{2(r)}, y, X)}{p(V^{(r-1)}|\beta^{(r)}, \sigma^{2(r)}, y, X)}.$$

MCMC for GLM with heteroskedasticity of known form

- For the model with the ith diagonal element of V given by $\sigma_i^2 = \exp(\alpha_1 + \alpha_2 z_i)$

```
# Initial values x = X
ols    = lm(y~x-1)
b      = ols$coef
e      = y-x%*%b
q <- log(e^2)
z <- cbind(rep(1,n),x2)
a <- solve(t(z)%*%z)%*%t(z)%*%q

# MCMC
sd = 0.1; burn = 1000; R = 20000; niter = burn+R
bs = matrix(0,niter,p); as = matrix(0,niter,q)

for (iter in 1:niter){
```

MCMC updating draws

```
# new mean and variance for b distribution using GLS tra
A  = exp(z%*%a/2)
y1 = y/A
x1 = x/matrix(A,n,p)
V1 = solve(iV0+t(x1)%*%x1)
b1 = V1%*%(iV0b0+t(x1)%*%y1)

# GS update for regression coefficients b
b  = b1+t(chol(V1))%*%rnorm(p)
```

MH step

```
# MH step for variance coefficients a
e  = y-x%*%b
a1 = rnorm(q,a,sd)
la = sum(dnorm(e,0,exp(z%*%a1/2),log=TRUE))
          -sum(dnorm(e,0,exp(z%*%a/2),log=TRUE))
if (log(runif(1))<la){a = a1}

bs[iter,]    = t(b)
as[iter,]    = t(a)
}
# final MCMC sample
bs = bs[(burn+1):R,]
as = as[(burn+1):R,]
```

- see **heterosked1.R**

Heteroskedasticity of unknown form

- Now suppose we do not know the form of the heteroskedasticity.
- If we are willing to assume instead that the error variances, σ_i^2 are drawn from a common distribution, the Gamma, i.e.

$$p(\sigma_i^2|v) = \prod_{i=1}^{n} \mathrm{Gamma}\left(\frac{v}{2}, \frac{v}{2}\right).$$

Heteroskedasticity of unknown form

- Now suppose we do not know the form of the heteroskedasticity.
- If we are willing to assume instead that the error variances, σ_i^2 are drawn from a common distribution, the Gamma, i.e.

$$p(\sigma_i^2|v) = \prod_{i=1}^{n} \mathrm{Gamma}\left(\frac{v}{2}, \frac{v}{2}\right).$$

- Two approaches:

1. Choose $2 < v < 30$. The closer to 30, the closer to homoskedasticity. A v of around 5 appears to be a reasonable choice (then explore what happens with different values for v).

Heteroskedasticity of unknown form

- Now suppose we do not know the form of the heteroskedasticity.
- If we are willing to assume instead that the error variances, σ_i^2 are drawn from a common distribution, the Gamma, i.e.

$$p(\sigma_i^2|v) = \prod_{i=1}^{n} \mathrm{Gamma}\left(\frac{v}{2}, \frac{v}{2}\right).$$

- Two approaches:

1. Choose $2 < v < 30$. The closer to 30, the closer to homoskedasticity. A v of around 5 appears to be a reasonable choice (then explore what happens with different values for v).
2. Adopt a hierarchical prior for v. The exponential distribution is the usual choice (see Geweke, 1993, and Koop, Poirier and Tobias (2007),

$$p(v) = \mathrm{Exp}(v, 2) = \frac{1}{2}\exp(-v/2).$$

Student-t errors

- If we take either approach, we obtain a neat result - Geweke (1993) shows that with this set up, if we integrate out the σ_i^2s, we get a Student-t error linear regression model with homoskedastic error (i.e. Student-t instead of Normal errors).

Student-t errors

- If we take either approach, we obtain a neat result - Geweke (1993) shows that with this set up, if we integrate out the σ_i^2s, we get a Student-t error linear regression model with homoskedastic error (i.e. Student-t instead of Normal errors).
- Equations (13.8), (13.12), (13.20) and (13.22) in Koop *et al.* (2007) (ch. 13) provide the full set of conditional posteriors for this model.

Student-t errors

- If we take either approach, we obtain a neat result - Geweke (1993) shows that with this set up, if we integrate out the σ_i^2s, we get a Student-t error linear regression model with homoskedastic error (i.e. Student-t instead of Normal errors).
- Equations (13.8), (13.12), (13.20) and (13.22) in Koop *et al.* (2007) (ch. 13) provide the full set of conditional posteriors for this model.
- We can draw sequentially from each of these conditionals to obtain an MCMC sample and conduct inference.

Student-t errors

- If we take either approach, we obtain a neat result - Geweke (1993) shows that with this set up, if we integrate out the σ_i^2s, we get a Student-t error linear regression model with homoskedastic error (i.e. Student-t instead of Normal errors).
- Equations (13.8), (13.12), (13.20) and (13.22) in Koop *et al.* (2007) (ch. 13) provide the full set of conditional posteriors for this model.
- We can draw sequentially from each of these conditionals to obtain an MCMC sample and conduct inference.
- An MH step is required to draw from (13.22) since the density is nonstandard. The other distributions are all of known form and so a Gibbs step can be employed. See Koop *et al.* (2007) for further details.
- see **glmterrs.R**