Preliminaries: Likelihood for the linear regression model

$$y = X\beta + e, \quad e \sim N\left(0, \sigma^2 I_n\right),$$

- The likelihood for the linear model is:

$$p(y_1, y_2, \ldots, y_n | X, \beta, \sigma^2) = p(y|X, \beta, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{Q}{2\sigma^2}\right),$$

with

$$Q = (y - X\beta)'(y - X\beta) = e'e \; (= \sum_{i=1}^{n} e_i^2).$$

Best algorithms of the 20th Century

- Dongarra and Sullivan (2000) *Computing in Science and Engineering*, *2*, 22-23.
- The top 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century (in chronological order)

1. **Metropolis algorithm for Monte Carlo**
2. Simplex method for linear programming
3. Krylov subspace iteration methods
4. The decompositional approach to matrix computations
5. The Fortran optimizing compiler
6. QR algorithm for computing eigenvalues
7. Quicksort algorithm for sorting
8. Fast Fourier transform
9. Integer Relation Detection
10. Fast Multipole Method

A Brief history of MCMC Theory

- While convalescing from an illness in 1946, Stan Ulum was playing solitaire. It occurred to him to try and compute the chances that a particular laid out solitaire of 52 cards would come out successfully.
- After attempting exhaustive combinatorial calculations, he decided to go for a more practical approach, laying out several solitaires at random and then observing and counting the number of successful plays.
- This idea of selecting a statistical sample to approximate a hard combinatorial problem by a much simpler problem is the main idea behind modern Monte Carlo simulation methods.
- Ulum realized that computers could be used in this way to answer questions of neutron diffusion and mathematical physics.

A Brief history of MCMC Theory (cont.)

- He contacted John Von Neumann and they developed many Monte Carlo algorithms that have been rediscovered in 1980s and 90s (importance sampling, rejection sampling, etc.)
- First paper: Metropolis and Ulum (**1949**) The Monte Carlo Method, *JASA.*
- Coworkers at Los Alamos, Metropolis, Teller, Von Neuman, *et al.,* coded these methods for use with the state-of-the-art computer (ENIAC).
- Metropolis *et al.* (**1953**) Equations of state calculations by fast computing machines, *Journal of Chemical Physics,* was the pioneering paper on MCMC, but it was overlooked by statisticians, partly because it was published in a chemistry journal, and partly because of the primitive level of computer technology available at the time.

How long until we recognize a good idea? 1970-1990s

- Hastings (**1970**) *Biometrika* and his student Peskun (1973) *Biometrika* generalized the Metropolis algorithm.
- Geman and Geman (1984) *IEEE Transactions* developed the Gibbs sampler for use in image processing, and Tanner and Wong (1987) *JASA* developed the data augmentation approach and were arguably the first to recognize the potential for Bayesian MCMC inference.
- However, it was the classic expository paper by Gelfand and Smith (**1990**) *JASA*, that brought the Gibbs sampler to the attention of a wider audience.

MCMC methods developed mostly in 1990s

- Widespread recognition of the practical importance of these algorithms occurred among statisticians in the 1990s (following Gelfand and Smith (1990)).
- This led to the rapid development of a generic set of MCMC tools for Bayesian inference and subsequently revolutionized the field of statistics.
- Most of the theoretical developments in MCMC were achieved in the 1990s.
- 2010+: An Econometric Oddity - There are still many "frequentists" in econometrics who refuse to use these methods just "because I'm a frequentist"!

Econometrics for the 21st century

- The posterior distribution for many models is often not available in an analytical form.
- The discovery that MCMC computational methods could be used to simulate from general Bayesian Hierarchical Models (Gelfand and Smith, 1990) revolutionized Statistical Science and made ever more complicated modeling scenarios possible." [Cressie & Wilke, p.31]
- "The emphasis in Statistical Science has moved from being able to derive *analytical* and/or asymptotic expressions for $p(X|D)$ and its properties, to being able to *simulate* from $p(X|D)$.
- Analytical calculations are still important because from them come deep understanding and sharp focus; however, the benefits of simulation are enormous." [CW (2011), p.44]

MCMC - Motivation

- *Simulation* from this distribution allows statistical inference to proceed, often in a relatively straightforward manner.
- Bayesian methods have recently produced some remarkably efficient solutions to complex inference problems.
- The approach is based on a combination of hierarchical prior modeling and MCMC simulation methods.
- This approach is able to tackle estimation and model interpretation in situations that are quite challenging by other means.

MCMC - Motivation

- See Greenberg, Part II - also Lancaster, chapter 4, Chib Handbook of Econometrics article, Gamerman and Lopes book for more details.
- Suppose you have a (posterior) distribution for some unknown quantities $\theta = (\theta_1, \theta_2)$ (two unknowns).
- If you draw (pseudo-) randomly from this distribution, $p(\theta_1, \theta_2|y)$, R times (i.e. R is the number of replications), you will have

$$\theta_R = \begin{bmatrix} \theta_1^{(1)} & \theta_2^{(1)} \\ \theta_1^{(2)} & \theta_2^{(2)} \\ \vdots & \vdots \\ \theta_1^{(R)} & \theta_2^{(R)} \end{bmatrix}.$$

MCMC Motivation (cont.)

- Each row of θ_R represents a randomly selected 'observation'/sampling from the **joint** distribution of θ_1 and θ_2, $p(\theta_1, \theta_2|y)$.
- Each column contains R realizations/'observations' from the **marginal** distribution of θ_j, $p(\theta_j|y)$.
- So we can study the distribution of θ_j simply by ignoring the other column - "it's as simple as that." [Lancaster, p.52]
- "Computer assisted sampling to avoid integration is the key feature of this approach. Increasingly, difficult mathematics is being abandoned in favor of computer power." [Lancaster, p.52]

What to do with a sample of observations

- If we have a sample of n observations of some 'random variable' X_i $(i = 1, 2, \ldots, n)$, we know how to summarize information about this sample (from intro. statistics!).
- We can calculate the mean, variance, standard deviation, etc.
- We can (nonparameterically) estimate the sampling distribution of X_i with a frequency count and plot histograms, etc.
- We can construct confidence intervals, perform hypothesis tests, etc., using the sample.
- If we recognize that with an MCMC sample for θ_j we are in exactly this situation.

Markov Chain Monte Carlo (MCMC): the idea

- The goal of Bayesian computation is to obtain a sample of draws $\theta^{(t)}$, $t = 1, \ldots, M$, from the posterior distribution of the unknown quantity θ, with a large enough sample that quantities of interest can be estimated with reasonable accuracy.
- MCMC simulation is a general method based on drawing values of θ from distributions that result in a sample from the target posterior distribution, $p(\theta|y)$.
- The sample is drawn sequentially, with the tth draw, $\theta^{(t)}$, depending only on the previous draw, $\theta^{(t-1)}$.

Treating the MCMC draws as a random sample

- Further, we can obtain **as large an MCMC sample as we like** just by making more draws using the MCMC algorithm.
- We know, from the law of large numbers, that the sample mean converges on the actual mean.
- so as **the number of draws in the MCMC sample**, $R \to \infty$,

$$\frac{\sum_{i=1}^{R} \theta_j^{(i)}}{R} \to E(\theta_j|D),$$

where D is the data sample used to obtain the posterior distribution.

- So we can just calculate means, standard deviations, etc. in the usual way because **we have a sample drawn from the posterior distribution** of θ_j.

The MCMC chain as a sample of observations

- We use the output of the MCMC algorithm as a sample of observations for the unknown quantity of interest
- It can be shown under remarkably general conditions, that the MCMC chain converges to draws from the marginal posterior distribution for each variable in the chain.
- That is, by drawing iteratively from the conditional posteriors for θ_1 and θ_2, given by $p(\theta_1|\theta_2, y)$ and $p(\theta_2|\theta_1, y)$, we obtain a sample for each that approximates values from the marginal posteriors $p(\theta_1|y)$ and $p(\theta_2|y)$.
- The longer we run the chain, the closer the approximation, so, since **we choose *R***, the number of MCMC draws, we can get **arbitrarily close** by running the chain for long enough.

MCMC convergence

- Typically, especially for the more standard, well understood models, the MCMC converges reasonably quickly to the marginal posteriors.
- We drop the first part of the chain, called the 'burn-in period' because it is likely that the chain hasn't converged at the beginning of the process
- This allows some time for the chain to converge, so that the approximation is better (though theoretically we can keep the entire sample).
- We can evaluate convergence by plotting the chain, looking at numerical standard errors, etc.
- For example, **bayesm** in R has a function to calculate n_{eff}, etc., called numEff.

Using the MCMC sample

- We can thus use the MCMC sample as we would a sample of observations on any observable.
- We can calculate mean, median, mode, standard deviation, confidence intervals, etc.
- A frequency plot of the sample (histogram) is an empirical estimate of the entire marginal posterior density.
- The more observations we have, the more accurate the estimates.
- We can typically run an MCMC chain for tens of thousands of iterations very quickly.
- If we require higher numerical accuracy, we run the algorithm for longer.

The Gibbs sampler

- The simplest MCMC, both to describe and to implement, is the Gibbs sampler (GS).
- Using the GS, we simulate successively from the conditional probability distributions.
- Suppose we have two sets of parameters, say covariate coefficients, θ and variance matrix, Ω.
- For $r = 1, \ldots R$, we choose starting values $\theta^{(0)}$ and $\Omega^{(0)}$ and sample iteratively from

$$p(Y^{(r)}|\theta^{(r-1)}, \Omega^{(r-1)}, Z),$$

$$p(\theta^{(r)}|, Y^{(r)}\Omega^{(r-1)}, Z),$$

$$p(\Omega^{(r)}|Y^{(r)}, \theta^{(r)}, Z).$$

Output from the Gibbs sampler

- At each step , the latest values obtained from the previous steps are used in the conditioning arguments. This defines a Markov chain with stationary distribution that is the posterior distribution $p(Y, \theta, \Omega|Z)$.
- The final result is a simulation of R observations for each of the elements of Y, θ and Ω.
- Hence the optimal predictor $E(Y|Z)$ and marginal posteriors $p(\theta|Z)$, $p(\Omega|Z)$, etc. can be approximated by (6).
- Another measure of uncertainty is the Bayesian credible interval [see CW, p.47].

MCMC with Normal-gamma prior

- Likelihood for n observations

$$y \sim N_n(X\beta, \sigma^2 I_n)$$

- Prior distribution for β **independent of** σ^2:

$$\beta \sim N_k(\beta_0, V_0)$$

- Prior for the precision τ,

$$\tau \sim Gamma(v_0/2, \delta_0/2)$$

or, if you prefer to specify the variance σ^2,

$$\sigma^2 \sim IG(v_0/2, \delta_0/2).$$

- Notation: some authors use δ_0, others use $n_0 S_0$ or some other notation.

Conditional posterior densities

$$\beta|\sigma^2, y, X \sim N_k(\beta_1, V_1),$$

$$\sigma^2|\beta, y, X \sim IG(v_1/2, \delta_1/2),$$

where

$$V_1 = (\sigma^{-2}X'X + V_0^{-1})^{-1},$$

$$\beta_1 = V_1(\sigma^{-2}X'y + V_0^{-1}\beta_0),$$

$$v_1 = v_0 + n,$$

$$\delta_1 = \delta_0 + (y - X'\beta)'(y - X'\beta).$$

MCMC Gibbs algorithm

- This leads to the following MCMC (Gibbs) algorithm [Greenberg Algorithm 8.1, p.112]
- a) Choose a starting value $\sigma^{2(0)}$,
- b) At the gth iteration, draw

$$\beta^{(g)}|\tau\sigma^2 \sim N_k(\beta_1^{(g)}, V_1^{(g)}),$$

$$\sigma^{2(g)}|\beta \sim IG(v_1/2, \delta_1^{(g)}/2).$$

- c) Repeat step b) until $g = B + G$ where B is the burn-in sample and G is the desired sample size.

Conditional distributions parameters

$$V_1^{(g)} = (\sigma^{-2(g-1)} X'X + V_0^{-1})^{-1},$$

$$\beta_1^{(g)} = V_1(\sigma^{-2(g-1)} X'y + V_0^{-1}\beta_0),$$

$$v_1 = v_0 + n,$$

$$\delta_1^{(g)} = \delta_0 + (y - X'\beta^{(g)})'(y - X'\beta^{(g)}).$$

- When using R or Matlab (or a different package), care must be taken to set the correct value for δ_0 and δ_1 when drawing from the Gamma distribution using rgamma or gam_rnd.

MCMC Gibbs sampler examples

- **Bayesregexample.R** provides examples using the bayesm in R.
- **linregmcmceg1.R** does essentially the same as above, just a shorter version.
- **mcmclinreg.R** is a "DIY" version to demonstrate the nuts and bolts of the algorithm, i.e. what's going on 'under the hood' (too many idioms?).
- **Koop-hprice-ch4.R** uses the data HPRICE.txt to reproduce the house prices example in Koop, chapter 4
- **chapter4a.m** is the matlab code from the Koop website to reproduce the above example.
- More examples with heteroskedastic errors below.