A second specific example of Gibbs sampling - canonical Markov chain Monte Carlo method for calculations in Bayesian binary regressions, and a trivial extension of Gibbs in linear regressions with multiple shrinkage.

♠ Probit Model

- Expression level vector $\mathbf{x}_j$ on array (tumor sample, etc) $j = 1, \ldots, n$
- Binary outcome: $z_j = 0$ or 1
- Probit probability model: $\pi_j = Pr(z_j = 1)$ (conditional on chosen predictor variables and model parameters: in full, $\pi_j = Pr(z_j = 1|\mathbf{x}_j, \boldsymbol{\beta})$)
- Linear regression model based on regression function $\mu_j = \mathbf{x}_j'\boldsymbol{\beta}$ (perhaps $\mathbf{x}_j$ is extended with a leading 1 to include an intercept term β_0)
- Probit regression:
$$\pi_j = \Phi(\mu_j)$$
where Φ is standard normal cumulative distribution function

♠ Prior and Posterior

- Example: $\boldsymbol{\beta} \sim N(0, \mathbf{C}^{-1})$ and we'll take the precision as diagonal
- Posterior for regression parameters
$$p(\boldsymbol{\beta}|\mathbf{z}) \propto \exp(-\boldsymbol{\beta}'\mathbf{C}\boldsymbol{\beta}) \prod_{j=1}^{n} \Phi(\mu_j)^{z_j}(1 - \Phi(\mu_j))^{1-z_j}$$
which can be evaluated, numerically optimized (it is unimodal) using standard NR routines,
- *Exercise:* Write code (in C/C++ or other) to implement a Newton-Raphson search for the mode of the posterior density. As ALWAYS in such problems, work with the log-posterior, i.e., maximise the log of the target function for numerical stability
- Under vague reference prior ($\mathbf{C} \to 0$) the posterior is the normalised likelihood function. The MLE and related information can be computer using R/Splus glm functions, Matlab glmfit function, or (quite easily) by user-written code as a special case

♠ Latent Variables and Data Augmentation

- For each sample, recognise an underlying latent variable $\mathbf{y}_j$
- $\pi_j = Pr(y_j > 0)$ when $y_j \sim N(\mu_j, 1) = N(\mathbf{x}_j'\boldsymbol{\beta}, 1)$
- $z_j = 1$ if and only if $y_j > 0$
  - e.g., latent variable is positive for ER+ cases, negative for ER− cases
  - could precisely classify cases if we could observe the latent y_j, but we do not; result is the binary probability model
  - MCMC calculations impute these "missing" values along with values of the parameters $\boldsymbol{\beta}$

♠ Conditional Posteriors

- $p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{z})$
If $\mathbf{y}$ were known, we have a linear regression of the y_j with regression variables $\mathbf{x}_j$, parameter $\boldsymbol{\beta}$ and error variance 1. Notice that this simplifies (it is a special case) of the linear regression with unknown error variance: we simply set the precision parameter ϕ to be 1, and then the analysis is all the same otherwise. In the MCMC, we just skip the simulation of ϕ values at each iteration, since we know $\phi = 1$

here.

The actual value of $\mathbf{z}$ is at this point irrelevant – the information they contain is already there in the (current, imputed or candidate) values of $\mathbf{y}$: Formally, $\boldsymbol{\beta}$ is conditionally independent of $\mathbf{z}$ given $\mathbf{y}$. The posterior (see earlier notes) is multivariate normal

$$\boldsymbol{\beta}|\mathbf{y}, \mathbf{z} \sim N(\mathbf{b}, \mathbf{B}^{-1})$$

with $\mathbf{B} = \mathbf{C} + \mathbf{H}'\mathbf{H}$ and $\mathbf{b} = \mathbf{B}^{-1}\mathbf{H}'\mathbf{y}$ where $\mathbf{H} = \mathbf{X}'$ is the $n \times p$ design matrix.
For multivariate normals, use programmed functions (multivariate normal simulation is very standard - e.g.. rMNorm.m or similar) or use direct Cholesky decomposition: e.g., by hand in matlab

$$\mathbf{b} + chol(inv(\mathbf{B})) * randn(p, 1)$$

for a single draw

- $p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{z})$
If $\boldsymbol{\beta}$ were known, then the y_j are independent normals but subject to the information provided by the z_j - in each case, we just condition the initial normal on the information that y_j must be positive (if $z_j = 1$) or negative (if $z_j = 0$). The result may be written simply in terms of the posterior cumulative distribution function, as

$$P(y_j|z_j = 1) = [\Phi(y_j - \mu_j) - (1 - \pi_j)]/\pi_j, \qquad \text{for } y_j > 0,$$

and

$$P(y_j|z_j = 0) = \Phi(y_j - \mu_j)/(1 - \pi_j), \qquad \text{for } y_j < 0$$

(check that you can derive this). Or, for any value of $z_j = 0, 1$,

$$P(y_j|z_j) = [\Phi(y_j - \mu_j) - z_j(1 - \pi_j)]/[z_j\pi_j + (1 - z_j)(1 - \pi_j)].$$

Simulated values of the y_j are then drawn, independently, via the inverse CDF approach: generate $u_j \sim U(0, 1)$ and solve for y_j in $P(y_j|z_j) = u_j$. It can be written trivially as

$$y_j = \mu_j + \Phi^{-1}\{z_j(1 - \pi_j) + u_j(z_j + (1 - \pi_j)(1 - 2z_j))\}$$

with Φ^{-1} being the normal quantile function (inverse CDF - qnorm function).

♠ Including Additional Parameters

MCMC neatly extends to include other parameters. A key example is the multiple shrinkage prior model that is of general interest, but of particular relevance in factor regression models (see earlier notes on this).

- Suppose $\mathbf{C} = \text{diag}(\gamma_0, \gamma_1, \ldots, \gamma_p)$ with prior variances $\tau_i = \gamma_i^{-1}$ defining element-wise shrinkage parameters for the individual predictor variables. The above discussion all applied now explicitly conditional on values of $\mathbf{C}$, so that the simulation iterations can run with simulations of $\mathbf{C}$ coupled in too. That requires priors on elements of $\mathbf{C}$; if these are independent gamma priors, $\gamma_j \sim Ga(k/2, h/2)$ for each j, say, then the relevant conditional posteriors are also independent gammas, namely

$$\gamma_j \sim Ga((k + 1)/2, (h + \beta_j^2)/2).$$

This allows for learning on differential shrinkage parameters across variables.
- *Exercise:* Develop the MCMC with these priors.
- Think about choices of prior parameters: One general way to think about ranges of relevant parameter values in binary regression models is to consider how variation in μ_j translates through to the probability scale π_j. Absolute values of μ_j bigger than 2 or so lead to probabilities that are already very extreme.