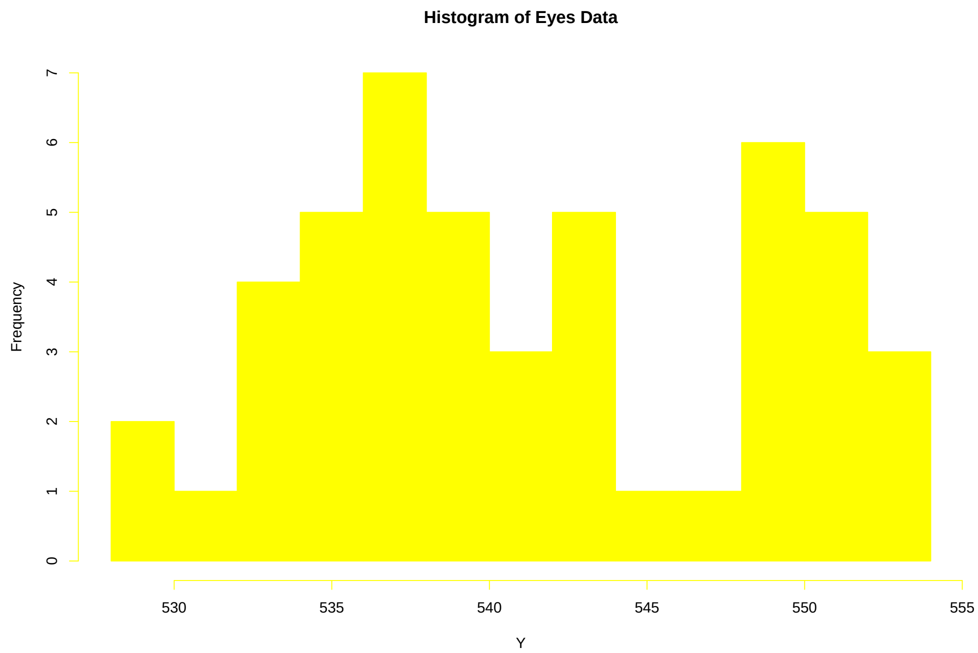
Mixture Models and Gibbs Sampling

February 22, 2010

Readings: Hoff CHapter 6

Eyes Exmple

Bowmaker et al (1985) analyze data on the peak sensitivity wavelengths for individual microspectophotometric records on a small set of monkey's eyes. WinBUGs Examples Volume II gives the data for one monkey.



Mixture Model

Model the data using a Mixture of 2 Normals:

$$Y_i \mid \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \pi_1, \pi_2 \overset{ind}{\sim} \pi_1 \mathsf{N}(\mu_1, \sigma_1^2) + \pi_2 \mathsf{N}(\mu_2, \sigma_2^2)$$

Which is equivalent to

$$\begin{aligned} Y_i \mid T_i, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2 &\overset{ind}{\sim} \mathsf{N}(\mu_{T_i}, \sigma_{T_i}^2) \\ T_i &\overset{iid}{\sim} \mathsf{Cat}(T, \pi) \end{aligned}$$

where T_i is a latent variable indicating which group observation i belongs to i.e. $T_i \in \{1, 2\}$ and $\mathsf{P}(T_i = j) = \pi_j$, and $\sum_j \pi_j = 1$

Prior Distributions

Based on WinBUGS example, adopt noninformative prior distributions

$$\mu_j \overset{iid}{\sim} \mathsf{N}(0, 1.0 \times 10^6)$$

$$1/\sigma_j^2 \overset{iid}{\sim} \mathsf{G}(0.001, 0.001)$$

$$(\pi_1, \pi_2) \sim \text{Dirichlet}(1, 1) \Leftrightarrow \pi_1 \sim \text{Beta}(1, 1))$$

Proper prior distributions are necessary for Mixture Models; if prior on μ or σ^2 is improper, then the posterior will also be improper if all observations are in one group! False sense of security with vague but proper priors...

Single Component Gibbs Sampler

Find full conditional distributions for

- $\mu_1 \mid \mu_2, \sigma_1^2, \sigma_2^2, \pi_1, \pi_2, T_1, \ldots, T_N, Y$ (normal)
- $\mu_2 \mid \mu_1, \sigma_1^2, \sigma_2^2, \pi_1, \pi_2, T_1, \ldots, T_N, Y$ (normal)
- $\sigma_1^2 \mid \mu_1, \mu_2, \sigma_2^2, \pi_1, \pi_2, T_1, \ldots, T_N, Y$ (gamma)
- $\sigma_2^2 \mid \mu_1, \mu_2, \sigma_1^2, \pi_1, \pi_2, T_1, \ldots, T_N, Y$ (gamma)
- $T_i \mid \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \pi_1, \pi_2, T_{(i)}, Y$ (Categorical)
- $(\pi_1, \pi_2) \mid \mu_1, \mu_2, \sigma_1^2, \sigma_2^2, T_1, \ldots, T_N, Y$ (Dirichlet)

Easy to find and sample!

Programs

BUGS: Bayesian inference Using Gibbs Sampling

- WinBUGS is the Windows implementation
  - can be called from R with `R2WinBUGS` package
  - can be run on any intel-based computer using VMware, wine
- OpenBUGS open source version of WinBUGS
- LinBUGS is the Linux implementation of OpenBUGS.
- JAGS: Just Another Gibbs Sampler is an alternative program that uses the same model description as BUGS (Linux, MAC OS X, Windows)

Include more than just Gibbs Sampling

BUGS

Need to specify

- Model
- Data
- Initial values

May do this through ordinary text files or use the functions in `R2WinBUGS` to specify model, data, and initial values then call WinBUGS.

Model Specification via R2WinBUGS

```
mixmodel=function() {
  for( i in 1 : N ) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- lambda[T[i]]
    T[i] ~ dcat(pi[]) }
  pi[1:2] ~ ddirch(alpha[])
  theta ~ dnorm(0.0, 1.0E-6)%_%I(0.0, )
  lambda[1] ~ dnorm(0.0, 1.0E-6)
  lambda[2] <- lambda[1] + theta
  tau ~ dgamma(0.001,0.001)
  sigma <- 1 / sqrt(tau)
}
```

Notes on Models

- Distributions of stochastic “nodes” are specified using $\sim$
- Assignment of deterministic “nodes” uses `<-` (NOT $=$)
- Cannot put expressions as arguments in distributions
- Normal distributions are parameterized using precisions, so `dnorm(0, 1.0E-6)` is a $N(0, 1.0 \times 10^6)$
- uses `for` loop structure as in R

Alternative Parameterization

- With vague prior distributions, the Gibbs sampler may get stuck with all observations assigned to one component (hard to escape)
- Label switching Problem
- Robert suggested parameterizing means

$$\lambda_1 \sim \mathsf{N}(0, 1.0 \times 10^6)$$

$$\theta \sim \mathsf{N}_+(0, 1.0 \times 10^6) \quad \theta > 0$$

$$\lambda_2 = \lambda_1 + \theta$$

Constrains Group 2 mean to be larger than Group 1.

Function to Return Initial Values as a List

```
inits = function() {
  lambda1 = mean(eyesdata$y[1:30]) +rnorm(1,0,.
  theta =  mean(eyesdata$y[31:48])-lambda1
  sigma2 = var(eyesdata$y[1:30])
  return(list(lambda = c(lambda1, NA),
              theta = theta,
              tau = 1/sigma2,
              pi = c(30, 48-30)/48))
}
```

- λ_2 is not random, so no initial value is specified (it is determined by λ_1 and θ
- If no initial value is given, BUGS will generate values given the other values, model and priors

Data

A list or rectangular data structure for all data and summaries of data used in the model

```
eyesdata= list(
  y = c(529.0, 530.0, 532.0, 533.1, 533.4,  ...
    535.3, 535.4, 535.9, 536.1, 536.3, 536.4, .
    538.3, 538.5, 538.6, 539.4, 539.6, 540.4, .
    543.5, 543.8, 543.9, 545.3, 546.2, 548.8, .
    549.9, 550.6, 551.2, 551.4,  ... 552.9,553.
  N = 48,
  alpha = c(1, 1),
  T = c(1, NA, NA, NA, NA, NA, NA, NA, NA, ...
    NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
    NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
    NA, 2))
```

Notes

- The variable T is treated as part of the data, rather than “prior”
- With the data sorted, assign the smallest observation to group 1, and the largest to group 2.
- any fixed hyperparameters can be given here

Specifying which Parameters to Save

The parameters to be monitored and returned to R are specified with the variable `parameters`

```
parameters = c("lambda", "theta", "sigma",
               "pi" )
```

- To save a whole vector (for example all lambdas, just give the vector name)
- May save stochastic or deterministic nodes

Running WinBUGS from R

Write the model out as a text file, then call `bugs()`

```
path = getwd()
model.file = paste(path,"model.txt", sep="")
write.model(mixmodel, model.file)

sim = bugs(eyesdata, inits, parameters, model.f
           n.chains=2, n.iter=5000,
           bugs.dir=BUGS.DIR, # for use with MA
           WINE=WINE,            #for use with MAC
           WINEPATH=WINEPATH, #for use with MAC
           debug=T, DIC=F)
```

`debug=T` keeps WinBUGS open – very useful for debugging BUGS!

Output

```
> sim
 2 chains, each with 5000 iterations
(first 2500 discarded), n.thin = 5
 n.sims = 1000 iterations saved
          mean  sd  2.5%   50%   97.5% Rhat n.e
lambda[1] 536.7 0.9 535.0  536.7  538.6    1  10
lambda[2] 548.9 1.2 546.3  548.9  551.3    1   7
theta      12.1 1.4   9.2   12.3   14.6    1  10
pi[1]       0.6 0.1   0.4    0.6    0.8    1  10
pi[2]       0.4 0.1   0.2    0.4    0.6    1  10
sigma       3.8 0.6   3.0    3.6    5.3    1  10
```