

StatMod2 - Hierarchical Models and Shrinkage - Polls

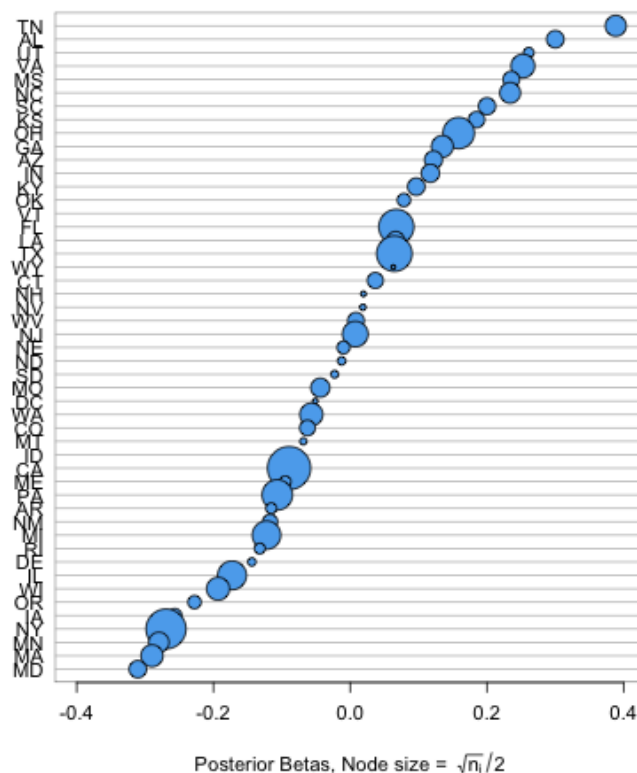
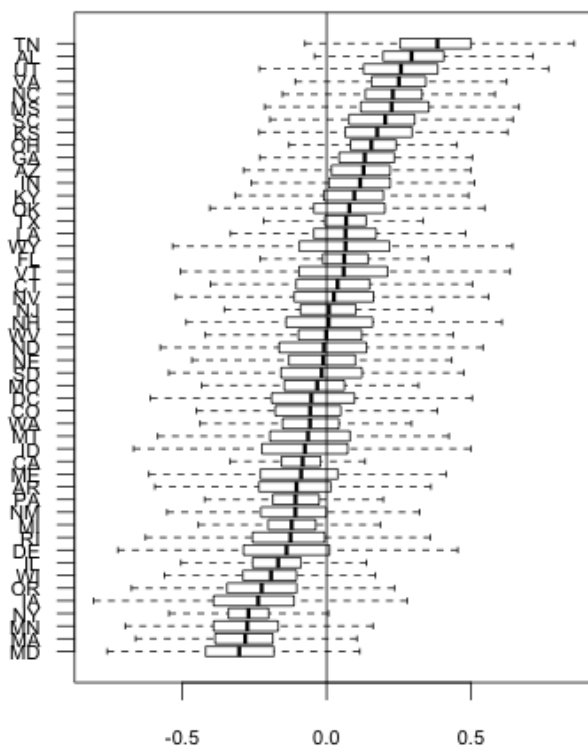
Maurice Diesendruck

April 22, 2015

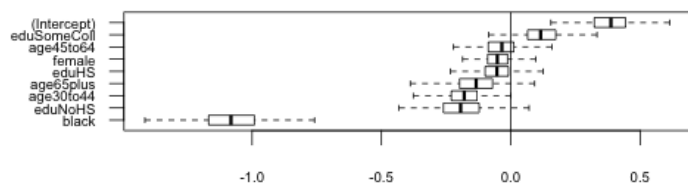
1 Summary

Albert and Chib (1993) present exact Bayesian methods for posterior sampling of binary response data. Where a probit regression would normally be used, they augment the model by introducing a left- or right-truncated normal variable Z , that corresponds to 1- or 0-valued y response, respectively. In doing so, the model enables standard linear model results for normal-normal posteriors, which can be used to find full conditionals for Gibbs sampling.

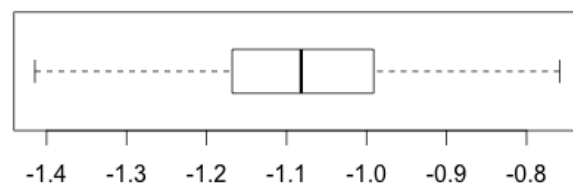
Mu Values by State



Beta Values by Covariate:
Baseline=Bacc/18to29!/female!/black



Boxplot of τ^2 Chain



2 Full R Code

```
# StatMod2 - Polls

library(MASS)
library(truncnorm)
library(Hmisc)

#####
# LOAD DATA.
# Load and investigate data.
data <- read.csv("polls.csv")
attach(data)
#aggregate(data, by=list(state=state), FUN=mean)
#subset(data, is.na(bush) & state=="CA", select=c(bush, state))

#####
# CLEAN DATA.
# Remove NA's.
d <- subset(data, !is.na(bush))
d <- subset(d, select=c(bush, state, edu, age, female, black))

#####
# BUILD DESIGN MATRIX.
y <- d$bush
# Add indicators: all 49 states (fixed effects); then with an intercept, with
# baselines for edu and age, plus female and black (random effects). The
# baseline is Bacc/18to29/!female/!black. The fixed effects are used to select
# values/sum over values based on state. The random effects are (1, HS, NoHS,
# SomeColl, age30to44, age45to60, age65plus, female, black), which is a vector
# for each respondent. Thus, the Beta vector has a first term associated with
# the intercept for the baseline person.
X <- model.matrix(~-1+state, d)
X <- cbind(X, model.matrix(~edu+age+female+black, d))

# Initialize operational variables.
n.states <- length(unique(d$state))
fixed.eff <- 1:n.states
random.eff <- (n.states+1):(dim(X)[2])
state.counts <- colSums(X[,fixed.eff])

#####
# DO SOME EXPLORATORY OLS.
f1 <- lm(y~X)
```

```

f2 <- lm(data$bush ~ data$state+data$edu+data$age+data$female+data$black)

#####
# DO GIBBS SAMPLER.
Gibbs <- function(y, X, n.iter=100) {

  # Initialize mu, beta, tausq, n.i, for first iteration of Gibbs.
  mu <- rep(0, length(fixed.eff))
  beta <- rep(0, length(random.eff))
  tausq <- 1
  CHAINS <- matrix(NA, nrow=n.iter, ncol=n.states+length(beta)+length(tausq))

  # Do Gibbs many times.
  for (iter in 1:n.iter) {
    z <- sample.z(mu, beta, y, X)
    mu <- sample.mu(state.counts, tausq, z, X, beta)
    tausq <- sample.tausq(n.states, mu)
    beta <- sample.beta(tausq, X, y, mu, z)
    CHAINS[iter,] <- c(mu, beta, tausq)
  }

  colnames(CHAINS) <- c(substring(colnames(X[,fixed.eff]),6,7),
                        colnames(X[,random.eff]), "tausq")

  return (list(CHAINS=CHAINS, n.iter=n.iter))
}

sample.z <- function(mu, beta, y, X) {
  n <- length(y)
  z <- rep(NA, n)
  for (i in 1:n) {
    if (y[i]==1) {
      z[i] <- rtruncnorm(1, a=0,
                        mean=mu%*%X[i,fixed.eff]+X[i,random.eff]%*%beta, sd=1)
    }
    if (y[i]==0) {
      z[i] <- rtruncnorm(1, b=0,
                        mean=mu%*%X[i,fixed.eff]+X[i,random.eff]%*%beta, sd=1)
    }
  }
  return (z)
}

sample.mu <- function(state.counts, tausq, z, X, beta) {
  n <- length(state.counts)
  C <- ginv(diag(x=(state.counts+1/tausq), n))
  mean <- C%*(t(X[,fixed.eff])%*(z - X[,random.eff]%*%beta))
  mu <- mvrnorm(1, mu=mean, Sigma=C)
}

```

```

    return (mu)
  }
sample.tausq <- function(n.states, mu) {
  shape <- n.states/2 + 1
  rate <- 1/2*(t(mu)%*%mu)
  tausq.inv <- rgamma(1, shape=shape, rate=rate)
  return (1/tausq.inv)
}
sample.beta <- function(tausq, X, y, mu, z) {
  XtX <- t(X[,random.eff])%*%X[,random.eff]
  XtOut <- t(X[,random.eff])%*%(z - X[,fixed.eff])%*%mu)
  C <- ginv(XtX)
  mean <- C%*%(XtOut)
  beta <- mvrnorm(1, mu=mean, Sigma=C)
  return (beta)
}

#####
# EVALUATE RESULTS.
results <- Gibbs(y, X, n.iter=500)
chain <- results$CHAINS
chain.mu <- chain[,fixed.eff]
chain.beta <- chain[,random.eff]
chain.tausq <- chain[,ncol(X)]
n.iter <- results$n.iter

# Show boxplots for each state's chain of Mu's.
par(las=1)
o <- order(apply(chain.mu, 2, median)) # Order chains by median.
boxplot(chain.mu[,o], horizontal=T, outline=F, cex.axis=0.7,
        main="Mu Values by State"); abline(v=0)
o <- order(apply(chain.beta, 2, median))
boxplot(chain.beta[,o], horizontal=T, outline=F, cex.axis=.6,
        main="")
title(bquote(atop("Beta Values by Covariate:",
                  "Baseline=Bacc/18to29/!female/!black"))); abline(v=0)
boxplot(chain.tausq, horizontal=T, outline=F,
        main=bquote("Boxplot of"~tau^2~"Chain"))

dfx = data.frame(ev1=colMeans(chain.mu), ev2=colnames(chain.mu),
                 ev3=state.counts)
dfx <- dfx[order(-dfx$ev1),]
dotchart2(dfx$ev1, labels=dfx$ev2, cex=.7,
          fg=NULL, pch=21, dotsize=sqrt(dfx$ev3)/2, add=F, ann=F,
          width.factor=2.5, bg="steelblue2", lty=1, lcolor="gray",
          xlab=bquote("Posterior Betas, Node size = " ~ sqrt(n[i])/2),
          xlim=c(-0.4,0.4))

```