Lab4: Fitting Bayesian Models in JAGS

Me & Merlise

Duke University

October 11, 2019

Block g-prior (Normal-Jeffreys)

Recall our standard centered parametrization of the likelihood:

$$\mathbf{Y} \;\; = \;\; \mathbf{1}\beta_0 + (\mathbf{I}_n - \mathbf{P}_1)\mathbf{X}_1\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

Now that we're 'being Bayesian', we want to set priors on the random quantities present in the likelihood; namely, $\beta_0, \boldsymbol{\beta},$ and ϕ.

Using the decomposition we've seen several times now – $\mathbf{Y} = (\mathbf{I} - \mathbf{P_1})\mathbf{Y} + \mathbf{P_1}\mathbf{Y}$ – we can write the log likelihood:

$$\mathcal{L}(\beta_0, \beta_1, \phi) \propto \frac{n}{2}\log(\phi) - \frac{\phi}{2}\left(n(\beta_0 - \bar{y})^2 + (\mathbf{Y}_c - \mathbf{X}_c\boldsymbol{\beta})^T(\mathbf{Y}_c - \mathbf{X}_c)\right)$$

Bayesian Model Specification Using the g-Prior

We can specify priors as follows:

$$\begin{aligned}
\mathbf{Y} &= \mathbf{1}\beta_0 + (\mathbf{I}_n - \mathbf{P}_1)\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \\
p(\phi) &\propto 1/\phi \\
p(\beta_0) &\propto 1 \\
\boldsymbol{\beta} \mid \phi &\sim \mathrm{N}(\mathbf{0}, \frac{g}{\phi}(\mathbf{X}^T(\mathbf{I}_n - \mathbf{P_1})\mathbf{X})^{-1}) \\
\tau := 1/g &\sim G(1/2, n/2)
\end{aligned}$$

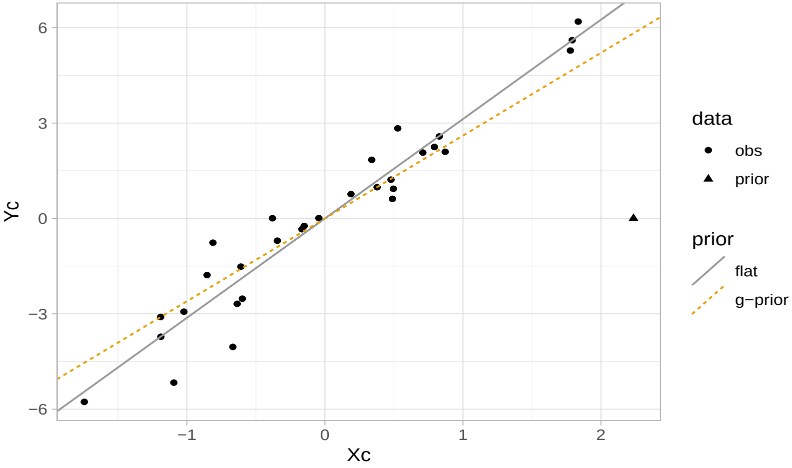
Here, we've specified independent priors for β_0 and $\boldsymbol{\beta}$ and also g.

Note that this is a specific case of the more general block g-prior (see Slide 16 in *G-Priors and Mixture Distributions*). In this formulation, we're being uninformative about our prior for β_0.

Note also that Merlise will concisely combine the priors for β_0 and ϕ and write $p(\phi, \beta_0) \propto 1/\phi$.

Example from class: g=5, n=30

In SLR g prior contribution is like adding an extra $Y_0 = 0$ at $\mathbf{X}_o = \sqrt{\frac{SS_x}{g}}$:



(Gibbs) Sampling from the Posterior

Now that we've specified our model (likelihood + priors), we would like to get samples from the joint posterior of $\beta_0, \boldsymbol{\beta}, \phi, g$.

We can do this by finding the full conditionals 1. $\beta_0|\boldsymbol{\beta}, \phi, g, \mathbf{Y}$, 2. $\boldsymbol{\beta}|\beta_0, \phi, g, \mathbf{Y}$, 3. $\phi|\beta_0, \boldsymbol{\beta}, g, \mathbf{Y}$, and 4. $g|\beta_0, \boldsymbol{\beta}, \phi, \mathbf{Y}$.

Given starting values, we can sequentially draw from these full conditionals and the resulting samples will represent samples from the joint posterior (Gibbs sampling).

Implement in JAGS or STAN (this way you don't need to find the full conditionals).

- If you're doing this on your local machine, must download both R2jags **R** package *and* OpenBUGS / WinBUGS.

JAGS Code: library(R2jags)

```
model = function(){
  for (i in 1:n) {
      Y[i] ~ dnorm(beta0+ (X[i] -Xbar)*beta, phi)
  }
  beta0 ~ dnorm(0, .000001*phi) # Precision is 2nd arg
  beta ~ dnorm(0, phi*tau*SSX)  # Precision is 2nd arg
  phi ~ dgamma(.001, .001) # Approx. to 1/phi prior
  tau ~ dgamma(.5, .5*n)
  g <- 1/tau
  sigma <- pow(phi, -.5)
}
data = list(Y=Y, X=X, n =length(Y), SSX=sum(Xc^2),
            Xbar=mean(X))
ZSout = jags(data, inits=NULL,
             parameters.to.save=c("beta0","beta", "g",
                                  "sigma"),
             model=model, n.iter=10000)
```

HPD intervals

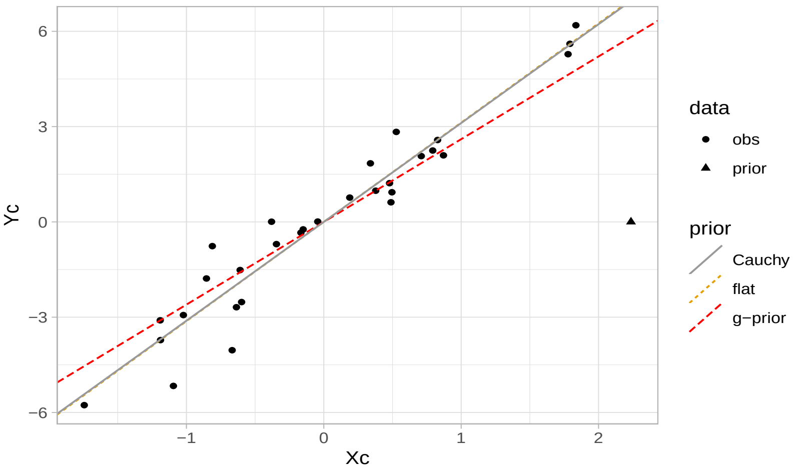
```
confint(lm(Y ~ Xc))

##                  2.5 %    97.5 %
## (Intercept) -0.3985359 0.2048303
## Xc           2.7945824 3.4555162

HPDinterval(as.mcmc(ZSout$BUGSoutput$sims.matrix))

##               lower        upper
## beta      2.7823047    3.4453690
## beta0    -0.3764027    0.2095465
## deviance 70.2043917   78.4813041
## g        19.4503373 3782.7134974
## sigma     0.6171029    1.0504892
## attr(,"Probability")
## [1] 0.95
```

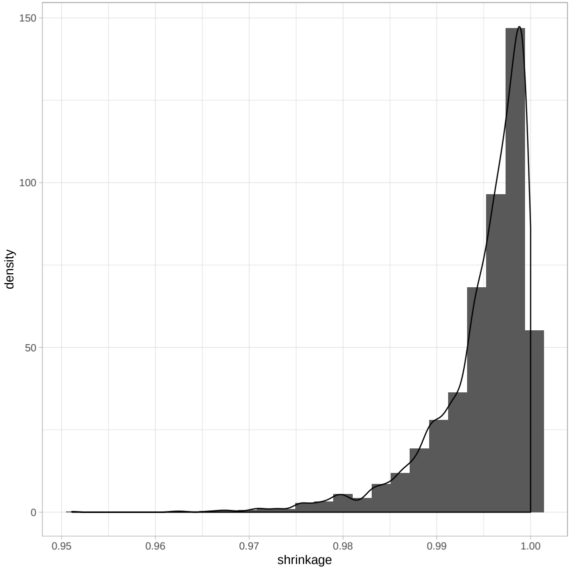
Compare



```
ZSout
```

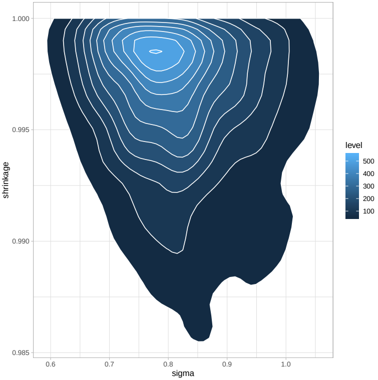
```
## Inference for Bugs model at "/var/folders/92/xd3h81wx6hxfv8rr_20mc3mh0000gn/T//RtmpuWpNbm/modelba7f69c3
##  3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
##  n.sims = 3000 iterations saved
##           mu.vect   sd.vect   2.5%     25%     50%     75%    97.5%  Rhat
## beta        3.112     0.170  2.782   2.997   3.115   3.225    3.445 1.001
## beta0      -0.099     0.152 -0.384  -0.204  -0.099   0.001    0.204 1.002
## g        2263.147 38967.029 48.273 146.129 282.298 697.063 9018.709 1.001
## sigma       0.827     0.114  0.636   0.747   0.816   0.896    1.079 1.001
## deviance   73.347     2.563 70.390  71.458  72.680  74.500   79.882 1.002
##          n.eff
## beta      3000
## beta0     1200
## g         3000
## sigma     3000
## deviance  1600
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 3.3 and DIC = 76.6
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Posterior Distribution of shrinkage



Joint Distribution of σ and $g/(1+g)$

```
ggplot(postdf, aes(x=sigma, y=shrinkage) ) +
 stat_density_2d(aes(fill = ..level..),
                 geom = "polygon", colour="white") +
  theme_light()
```



Summary

- The above prior specification is known as the Zellner-Siow (Cauchy) prior, because the marginal prior induced on $\boldsymbol{\beta}$ is Cauchy.
- This model rejects prior information not supported by the data
- The above robustness stems from a concept known as 'bounded influence' (more later)
- Sampling from the posterior requires numerical integration or MCMC sampling