ST 540 Exam 2

Parker Trostle

April 9, 2018

1 Statistical model

I proceeded in two steps to estimate $\boldsymbol{\theta}$. The first step is a Bayesian SLR, and the second step is the expected MCMC.

During my initial data review, I saw $\boldsymbol{Y_{1,1}}$ (that is, Y_1's estimate of the first pixel) and $\boldsymbol{Y_2}$ were highly correlated when they were both non-missing. Their correlation is nearly perfect: it's 99.6%. In an effort to improve runtime of the upcoming MCMC code, my first step is to use the non-missing $\boldsymbol{Y_{1,1}}$ and $\boldsymbol{Y_2}$ pairs to impute missing values in both samples. I performed this imputation using Bayesian SLR. I then used this imputed data in the larger, more complicated MCMC.[1] To save space, I omit the model here.[2]

My main MCMC model attempts to simulate Dr. Reich's own simulation using the satellite data as observed data. My philosophy was that $\boldsymbol{Y_1}$ should be trusted highly and that I should account for noise and a small amount of bias in $\boldsymbol{Y_2}$ and $\boldsymbol{Y_3}$. At a high level, and suppressing the priors and some of the notation, my model is:

$$P(\boldsymbol{\theta}|\boldsymbol{Y_1}, \boldsymbol{Y_2}, \boldsymbol{Y_3}) \propto P(\boldsymbol{Y_1}|\boldsymbol{\theta})P(\boldsymbol{Y_2}|\boldsymbol{\theta_1})P(\boldsymbol{Y_3}|\sum_{j=1}^{6}\boldsymbol{\theta_{t,j}}/6)$$

Additional details are in the JAGS code, but the main highlights are:

$$\begin{aligned}
&Y_{1\{i,j\}} \sim N(\theta_{\{i,j\}}, \sigma^2_{Y_1}) && \text{Where } \sigma^2_{Y_1} \sim Gamma(3, 0.5)\\
&Y_{2\{i\}} \sim N(\theta_{1\{i\}} + bias[Y_2], \sigma^2_{Y_2}) && \text{Where } \sigma^2_{Y_2} \sim Gamma(0.1, 0.1)\\
&Y_{3\{i\}} \sim N(\sum_{j=1}^{6}\theta_{3\{i,j\}}/6 + bias[Y_3], \sigma^2_{Y_3}) && \text{Where } \sigma^2_{Y_3} \sim Gamma(0.1, 0.1)
\end{aligned}$$

I simulated both biases as uniformly distributed on (-2,2). $\boldsymbol{\theta}$ is simulated as described in Dr. Reich's notes using uninformed priors on ρ, $\boldsymbol{\mu_1}$, $\boldsymbol{\mu_2}$, $\boldsymbol{\Sigma_1}$, and $\boldsymbol{\Sigma_2}$.

[1] In an ideal world, I wouldn't use this step. I realize that as a result of using imputed data, I'm not quantifying some of the uncertainty in the model for my eventual estimates of $\boldsymbol{\theta}$. As a counterpoint, the relationship is almost perfectly linear, and I know it's based on simulated data. I ultimately decided under the circumstances the trade-off in faster MCMC code for debugging and testing was worth it.

[2] I used uninformed priors identical to those used in HW7. Note that a frequentist SLR yields nearly exactly the same point estimates.

2 JAGS code

```
##### Likelihoods
for(i in 1:n) {
  for(j in 1:6) {
    Y1[i,j] ~ dnorm(theta[i,j], Y1_variance)
  }
  Y2[i] ~ dnorm(theta[i,1] + Y2_bias, Y2_variance)
  Y3[i] ~ dnorm(1/6*(theta[i,1] + theta[i,2] + theta[i,3] +
       theta[i,4] + theta[i,5] + theta[i,6]) + Y3_bias, Y3_variance)
}

##### Theta matrix
theta[1,1:6] ~ dmnorm(mu1[1:6], Sigma1[1:6, 1:6])
for(i in 2:n) {
  theta[i,1:6] ~ dmnorm(mu2[1:6] + rho * theta[i-1,1:6], Sigma2[1:6, 1:6])
}

##### Priors
rho ~ dunif(0, 1)
for(i in 1:6) {
  mu1[i] ~ dnorm(0, 0.01)
  mu2[i] ~ dnorm(0, 0.01)
}

# initialize covariance matrices
# note: I sent R[,] as an argument. It's a 6x6 diag(1/6.1) matrix
Sigma1[1:6,1:6] ~ dwish(R[,], 6.1)
Sigma2[1:6,1:6] ~ dwish(R[,], 6.1)

# satellite characteristics
Y1_variance ~ dgamma(3, 0.5)
Y2_variance ~ dgamma(0.1, 0.1)
Y3_variance ~ dgamma(0.1, 0.1)
Y2_bias ~ dunif(-2, 2)
Y3_bias ~ dunif(-2, 2)
```

For reference, one of the Bayesian SLR models referenced as step 1 is:

```
# Likelihood
for(i in 1:n) {
  Y1[i] ~ dnorm(mu[i], inv.var)
  mu[i] <- beta[1] + beta[2]*Y2[i]
}

# Priors
beta[1] ~ dnorm(0, 0.01)
beta[2] ~ dnorm(0, 0.0001)
inv.var ~ dgamma(0.01, 0.01)
```

3 Convergence diagnostics

Overall, convergence across the 2,190 simulated θ values was good. I used three chains of 40,000 burn-in samples with 500,000 actual samples (with significant thinning). There were autocorrelation problems with some $\boldsymbol{\theta}$ posterior distributions, but they were mostly eradicated by increasing the sample sizes. If I were to continue to improve this analysis, I would try to improve the Geweke diagnostics at the edge cases.

Table 1: Effective sample sizes (percentiles)

1%	10%	25%	50%	75%	90%	99%
19,457	33,497	44,366	54,366	58,819	60,000	61,157

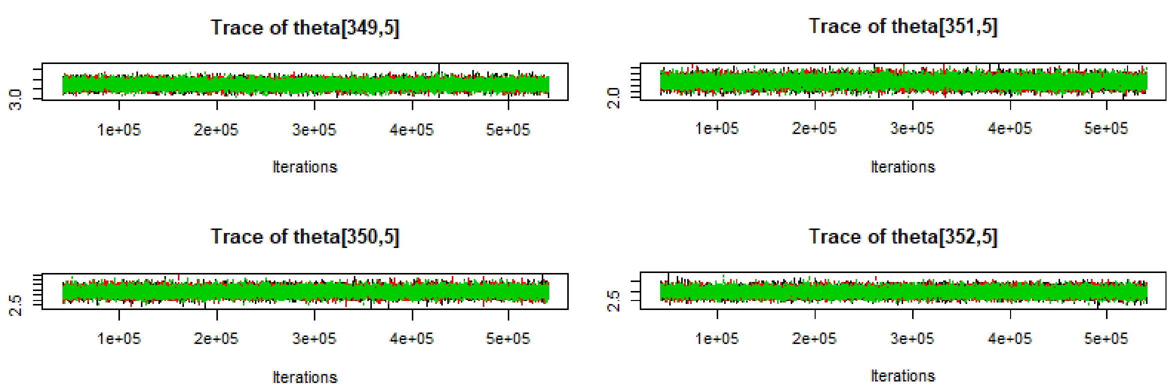
Table 2: Gelman diagnostics (percentiles)

1%	10%	25%	50%	75%	90%	99%
1.0000	1.0000	1.0000	1.0000	1.0001	1.0002	1.0004

Table 3: Geweke diagnostics (percentiles)

1%	10%	25%	50%	75%	90%	99%
-2.14	-1.21	-0.66	0.02	0.71	1.39	2.31

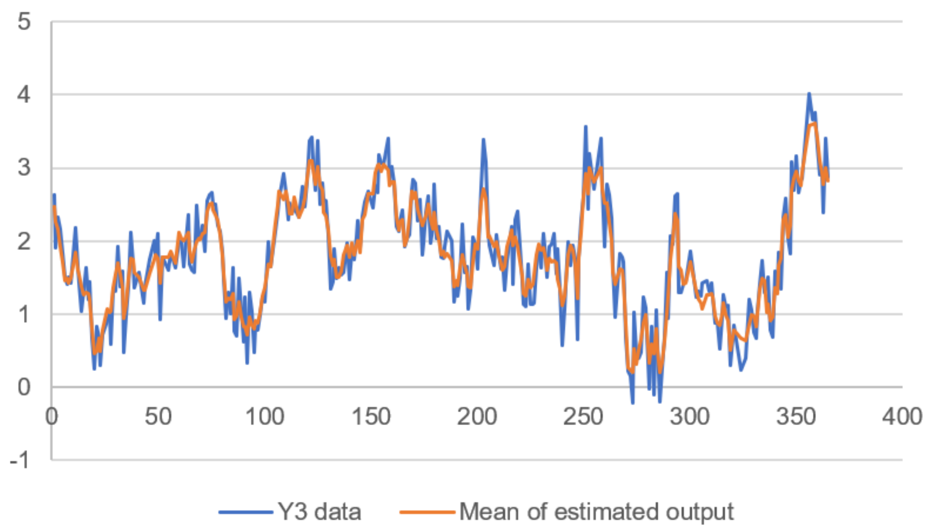
All the trace plots appeared fine. They all looked like:



4 Final results

I performed three checks to confirm the point estimates produced by my analysis were reasonable. First, I wanted to make sure the point estimates for $\boldsymbol{\theta}$ when $\boldsymbol{Y_1}$ wasn't missing were close to $\boldsymbol{Y_1}$. They are: the MSE between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{Y_1}$ when $\boldsymbol{Y_1}$ is not missing is only 0.005.

Second, I wanted to make sure the pattern of $\sum_{j=1}^{6} \hat{\boldsymbol{\theta}}_{\boldsymbol{t,j}}/6$ wasn't entirely different from that of $\boldsymbol{Y_3}$. I assumed $\boldsymbol{Y_3}$ was noisy and biased, so I wouldn't expect the results to perfectly match up, but they should be similar. In fact, the results match well.



Finally, I made my own $\boldsymbol{\theta}$ randomly using the same procedure as Dr. Reich, generated my own Y_1, Y_2, and Y_3 data (missing in the same spots and with noise and biases thrown in), and then attempted to run my analysis against my own simulated data. To summarize briefly, I generated data with a lot more variance than Dr. Reich's, but most of my individual squared errors were still less than 1.